

K8s w/Calico for LoxiLB Lab

2023년 7월

안종석

james@jslab.kr



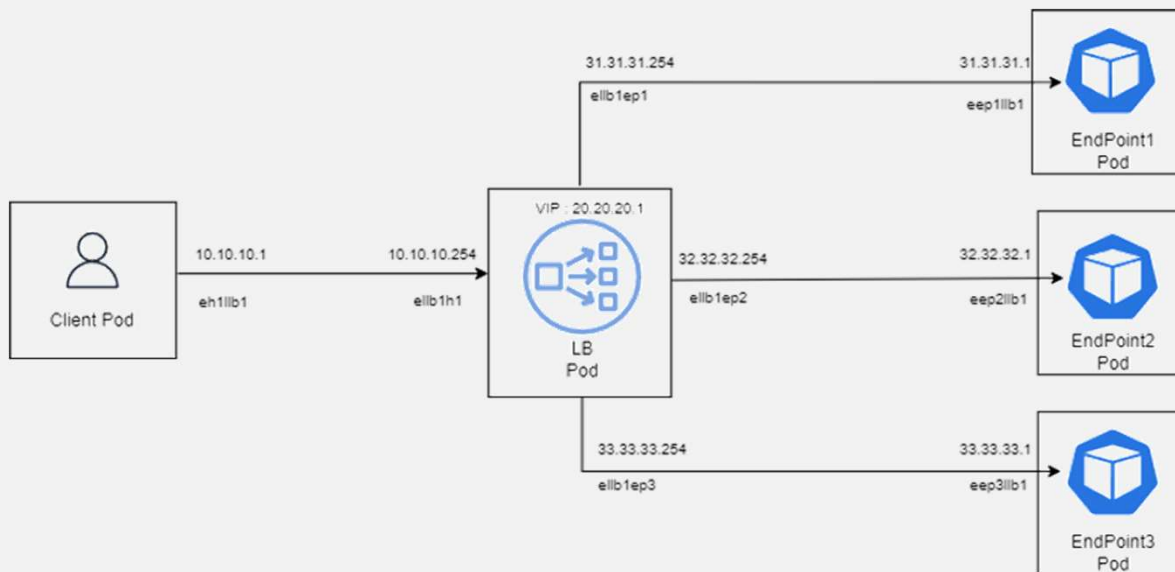
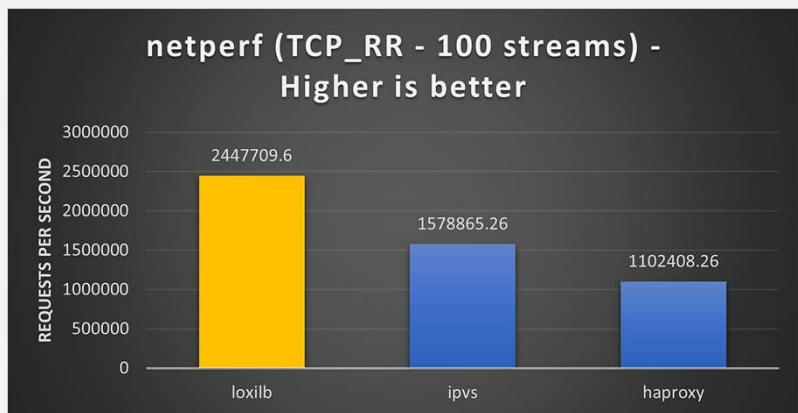
james@jslab.kr

JS Lab

OVERVIEW

❖ Reference: Throughput for LoxiLB

- Comparing LoxiLB with ipvs and HAProxy on AWS Graviton2-based EC2 instance



Source: <https://www.loxilb.io/post/running-loxilb-on-aws-graviton2-based-ec2-instance>



PREREQUISITES

❖ Vagrant and VirtualBox

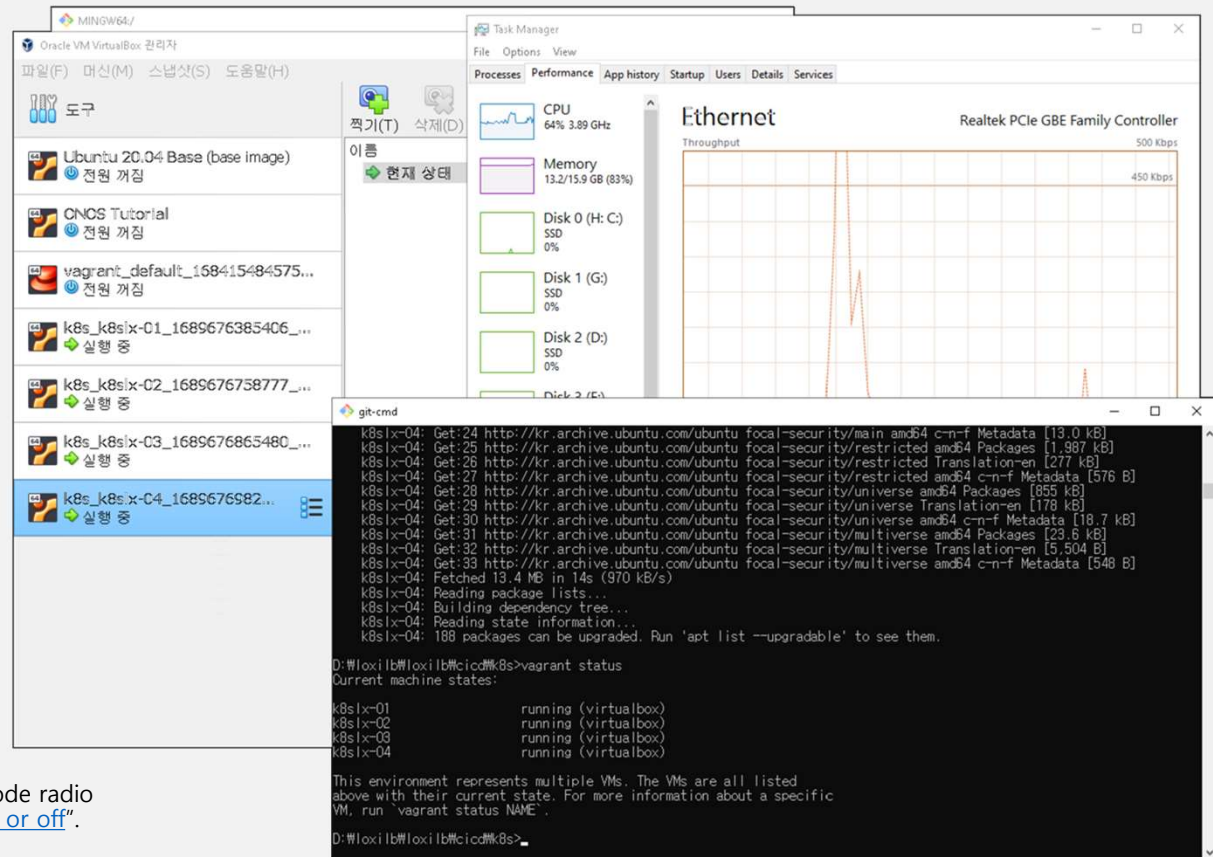
• Vagrant Installation w/VirtualBox

• vagrant up --provider=

- 1) hyperv
- 2) virtualbox
- 3) vmware_desktop

• vagrant status

• vagrant halt



Go to Settings > Update & Security > For Developers. Check the Developer Mode radio button. And search for "Windows Features", choose "Turn Windows features on or off".


Source: <https://itslinuxfoss.com/install-rustdesk-ubuntu-22-04/>

Source: <https://gist.github.com/1eedaegon/6a5a2ab64f09b080eadf3f84dad96f8>

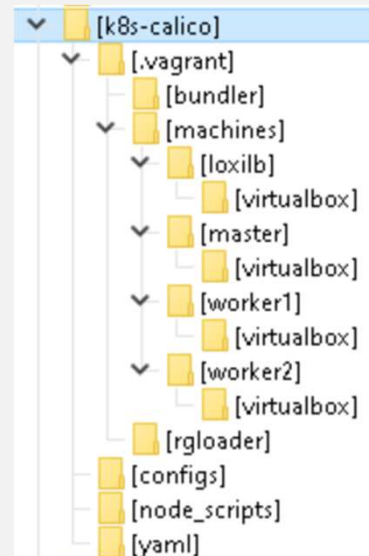


PREREQUISITES

❖ **K8s-calico** ** cimd is the best for Ubuntu Desktop, this guide is for some limited Windows and for Mac

-  **Git Installation** (for Windows)
 - **Cloning:** `sudo git clone https://github.com/loxilb-io/loxilb.git`
 - **cimd Location for Vagrant:** `loxilb/cimd/k8s-calico`
 - **directory**

- **configs**
- **node_scripts**
- **yaml**
- **config.sh**
- **loxilb-ip**
- **rmconfig.sh**
- **Vagrantfile**
- **validation.sh**



```
PS D:\loxilb\loxilb\cimd\k8s-calico> dir
Directory: D:\loxilb\loxilb\cimd\k8s-calico

Mode                LastWriteTime         Length Name
----                -
d-----            2023-07-18 오후 8:44             .vagrant
d-----            2023-07-18 오후 7:21             configs
d-----            2023-07-18 오후 7:21             node_scripts
d-----            2023-07-18 오후 7:21             yaml
-a-----            2023-07-18 오후 7:21             1599 config.sh
-a-----            2023-07-18 오후 8:49             13 loxilb-ip
-a-----            2023-07-18 오후 7:21             192 rmconfig.sh
-a-----            2023-07-18 오후 7:21             2945 Vagrantfile
-a-----            2023-07-18 오후 7:21             3674 validation.sh

PS D:\loxilb\loxilb\cimd\k8s-calico>
```

• **vagrant up** ### Check Vagrantfile source

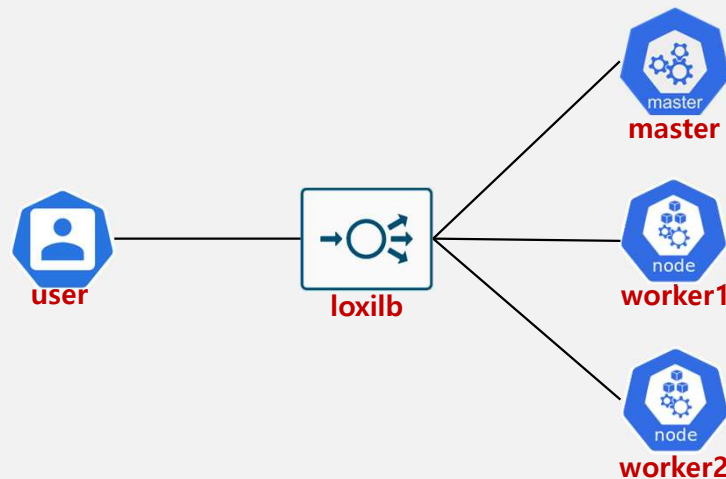
Source: <https://github.com/loxilb-io/loxilb/tree/main/cimd/k8s-calico>



VAGRANT UP

❖ Vagrant up

- vagrant up
- vagrant status
- vagrant halt



Source: <https://github.com/loxilb-io/loxilb/blob/main/cicd/k8s-calico/Vagrantfile>

```
# -*- mode: ruby -*-
# vi: set ft=ruby :
require "yaml"
settings = YAML.load_file "yaml/settings.yaml"
workers = settings["nodes"]["workers"]["count"]

Vagrant.configure("2") do |config|
  config.vm.define "loxilb" do |loxilb|
    loxilb.vm.box = settings["software"]["loxilb"]["box"]["name"]
    loxilb.vm.box_version = settings["software"]["loxilb"]["box"]["version"]
    loxilb.vm.hostname = "llb1"
    #loxilb.vm.network "forwarded_port", guest: 55002, host: 5502, protocol: "tcp"
    loxilb.vm.network :private_network, ip: settings["network"]["loxilb_ip"], :netmask => "255.255.255.0"
    loxilb.vm.provision :shell, :path => "node_scripts/loxilb.sh"
    loxilb.vm.provider :virtualbox do |vbox|
      vbox.customize ["modifyvm", :id, "--memory", 6000]
      vbox.customize ["modifyvm", :id, "--opus", 4]
    end
  end

  config.vm.define "cluster" do |cluster|
    config.vm.define "master" do |master|
      master.vm.hostname = "master"
      master.vm.network :private_network, ip: settings["network"]["control_ip"], :netmask => "255.255.255.0"
      master.vm.provision "shell",
        env: {
          "DNS_SERVERS" => settings["network"]["dns_servers"].join(" "),
          "ENVIRONMENT" => settings["environment"],
          "KUBERNETES_VERSION" => settings["software"]["kubernetes"],
          "OS" => settings["software"]["os"]
        },
        path: "node_scripts/common.sh"
      master.vm.provision "shell",
        env: {
          "CALICO_VERSION" => settings["software"]["calico"],
          "CONTROL_IP" => settings["network"]["control_ip"],
          "POD_CIDR" => settings["network"]["pod_cidr"],
          "SERVICE_CIDR" => settings["network"]["service_cidr"]
        },
        path: "node_scripts/master.sh"
      master.vm.provider :virtualbox do |vbox|
        vbox.customize ["modifyvm", :id, "--memory", 4096]
        vbox.customize ["modifyvm", :id, "--opus", 2]
      end
    end

    (1..workers).each do |node_number|
      config.vm.define "worker#{node_number}" do |worker|
        worker.vm.hostname = "worker#{node_number}"
        ip = node_number + 100
        worker.vm.network :private_network, ip: "192.168.80.#{ip}", :netmask => "255.255.255.0"
        worker.vm.provision "shell",
          env: {
            "DNS_SERVERS" => settings["network"]["dns_servers"].join(" "),
            "ENVIRONMENT" => settings["environment"],
            "KUBERNETES_VERSION" => settings["software"]["kubernetes"],
            "OS" => settings["software"]["os"]
          },
          path: "node_scripts/common.sh"
        worker.vm.provision "shell", path: "node_scripts/worker.sh"
      end

      worker.vm.provider :virtualbox do |vbox|
        vbox.customize ["modifyvm", :id, "--memory", 4096]
        vbox.customize ["modifyvm", :id, "--opus", 2]
      end
    end
  end
end
```



VAGRANT STATUS

❖ Vagrant status

- vagrant up
- vagrant status
- vagrant halt

The screenshot shows the Oracle VM VirtualBox Manager interface. On the left, a list of VMs is displayed, including 'Ubuntu 20.04 Base (base image)', 'ONOS Tutorial', and several 'k8s-calico' VMs. A Windows PowerShell terminal window is open in the foreground, showing the output of the 'vagrant status' command. The terminal output indicates that the 'k8s-calico' environment consists of multiple VMs, all of which are currently in a 'running (virtualbox)' state.

```

worker2: SSH username: vagrant
worker2: SSH auth method: private key
worker2: Warning: Connection reset. Retrying...
worker2: Warning: Connection aborted. Retrying...
==> worker2: Machine booted and ready!
==> worker2: Checking for guest additions in VM...
==> worker2: Setting hostname...
==> worker2: Configuring and enabling network interfaces...
==> worker2: Mounting shared folders...
worker2: /vagrant => D:\loxilb\loxilb\cicd\k8s-calico
==> worker2: Machine already provisioned. Run 'vagrant provision' or use the '--provision'
==> worker2: flag to force provisioning. Provisioners marked to run always will still run.
PS D:\loxilb\loxilb\cicd\k8s-calico> vagrant status
Current machine states:

loxilb                running (virtualbox)
master                running (virtualbox)
worker1               running (virtualbox)
worker2               running (virtualbox)

This environment represents multiple VMs. The VMs are all listed
above with their current state. For more information about a specific
VM, run 'vagrant status NAME'.
PS D:\loxilb\loxilb\cicd\k8s-calico>
  
```

The screenshot shows the Windows Task Manager Performance tab. It displays system specifications and resource utilization for the CPU, Memory, Disk, and Ethernet. The CPU is an Intel(R) Core(TM) i5-8500 CPU @ 3.00GHz, running at 3.98 GHz with 9% utilization. Memory is at 10.8/15.9 GB (68%) utilization. The system has 6 cores and 6 logical processors. Ethernet adapters are shown with 50.45 Mbps and 0.45 Mbps speeds.

Component	Value
CPU	9% 3.98 GHz
Memory	10.8/15.9 GB (68%)
Disk 0 (H: C:)	SSD 0%
Disk 1 (G:)	SSD 0%
Disk 2 (D:)	SSD 0%
Disk 3 (E:)	SSD 1%
Disk 4 (F:)	HDD 0%
Ethernet	50.45 S: 80.0 R: 8.0 Kbps
Ethernet	0.45 1.45 S: 0 R: 0 Kbps



VAGRANT BOX LIST

❖ Operations for vagrant

• vagrant box list


```
PS D:/loxilb/loxilb/cicd/k8s-calico> vagrant box list
bento/ubuntu-22.04      (virtualbox, 202303.13.0)
centos/7                (virtualbox, 2004.01)
manageiq/najdorf       (virtualbox, 14.1.3)
netlox/loxilight       (virtualbox, 1.0.0)
nix_inho/loxilbtest    (virtualbox, 1.0.0)
sysnet4admin/Ubuntu-k8s (virtualbox, 0.7.1)
PS D:/loxilb/loxilb/cicd/k8s-calico>
```

Source: <https://github.com/loxilb-io/loxilb/actions/runs/5312300896/jobs/9616620226>



VAGRANT SSH FOR K8S MASTER

❖ Vagrant ssh for K8s master

-  `vagrant global-status`
- `vagrant ssh f17541e` ### k8s master

```
PS D:/loxilb/loxilb/cicd/k8s-calico> vagrant global-status
id      name      provider  state  directory
-----
5a6c28c loxilb    virtualbox running D:/loxilb/loxilb/cicd/k8s-calico
f17541e master    virtualbox running D:/loxilb/loxilb/cicd/k8s-calico
7763de1 worker1   virtualbox running D:/loxilb/loxilb/cicd/k8s-calico
0cca613 worker2   virtualbox running D:/loxilb/loxilb/cicd/k8s-calico
```

The above shows information about all known Vagrant environments on this machine. This data is cached and may not be completely up-to-date (use "vagrant global-status --prune" to prune invalid entries). To interact with any of the machines, you can go to that directory and run Vagrant, or you can use the ID directly with Vagrant commands from any directory. For example: "vagrant destroy 1a2b3c4d"

```
PS D:/loxilb/loxilb/cicd/k8s-calico> vagrant ssh f17541e
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-67-generic x86_64)
```

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

System information as of Tue Jul 18 01:48:55 PM UTC 2023

```
System load: 1.2900390625   Users logged in: 0
Usage of /: 20.2% of 30.34GB IPv4 address for eth0: 10.0.2.15
Memory usage: 21%         IPv4 address for eth1: 192.168.80.10
Swap usage: 0%           IPv4 address for tun10: 172.16.219.64
Processes: 177
```

```
* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.
```

<https://ubuntu.com/pro>

```
This system is built by the Bento project by Chef Software
More information can be found at https://github.com/chef/bento
vagrant@master:~$
```



K8S MASTER OPERATIONS

❖ ip operations @ k8s master node


 • ip a

```
vagrant@master:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:10:0b:45 brd ff:ff:ff:ff:ff:ff
    altname enp0s3
    inet 10.0.2.15/24 metric 100 brd 10.0.2.255 scope global dynamic eth0
        valid_lft 82968sec preferred_lft 82968sec
    inet6 fe80::a00:27ff:fe10:b45/64 scope link
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:20:95:06 brd ff:ff:ff:ff:ff:ff
    altname enp0s8
    inet 192.168.80.10/24 brd 192.168.80.255 scope global eth1
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe20:9506/64 scope link
        valid_lft forever preferred_lft forever
4: tunl0@NONE: <NOARP,UP,LOWER_UP> mtu 1480 qdisc noqueue state UNKNOWN group default qlen 1000
    link/ipip 0.0.0.0 brd 0.0.0.0
    inet 172.16.219.64/32 scope global tunl0
        valid_lft forever preferred_lft forever
7: calie7334d930d1@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1480 qdisc noqueue state UP group default qlen 1000
    link/ether ee:ee:ee:ee:ee:ee brd ff:ff:ff:ff:ff:ff link-netns 43f245a6-d463-4c64-8578-cdb54436f140
    inet6 fe80::ecee:eeff:feee:eeee/64 scope link
        valid_lft forever preferred_lft forever
8: calibaf4d069c15@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1480 qdisc noqueue state UP group default qlen 1000
    link/ether ee:ee:ee:ee:ee:ee brd ff:ff:ff:ff:ff:ff link-netns 0558ca38-c141-4c83-a0f9-048b1b2afbfa
    inet6 fe80::ecee:eeff:feee:eeee/64 scope link
        valid_lft forever preferred_lft forever
9: cali4fc5e35f484@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1480 qdisc noqueue state UP group default qlen 1000
    link/ether ee:ee:ee:ee:ee:ee brd ff:ff:ff:ff:ff:ff link-netns 4c3109e9-6025-49f1-b383-964b763f5fba
    inet6 fe80::ecee:eeff:feee:eeee/64 scope link
        valid_lft forever preferred_lft forever
vagrant@master:~$
```



K8S MASTER OPERATIONS

❖ operations @ k8s master node

-  kubectl get node
- kubectl get services
- kubectl get pods
- kubectl get pods --all-namespaces
- kubectl get services --all-namespaces


```
vagrant@master:~$ kubectl get node
NAME      STATUS    ROLES    AGE   VERSION
master    Ready    control-plane   120m   v1.27.1
worker1   Ready    worker         116m   v1.27.1
worker2   Ready    worker         113m   v1.27.1
vagrant@master:~$ kubectl get services
NAME      TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
kubernetes ClusterIP     172.17.0.1    <none>        443/TCP    121m
vagrant@master:~$ kubectl get pods
No resources found in default namespace.
vagrant@master:~$ kubectl get pods --all-namespaces
NAMESPACE   NAME                                                    READY   STATUS    RESTARTS   AGE
kube-system  calico-kube-controllers-786b679988-vrx57              1/1     Running   1           128m
kube-system  calico-node-fm97n                                       1/1     Running   1           128m
kube-system  calico-node-jnvdc                                       1/1     Running   1           124m
kube-system  calico-node-rdc8h                                       1/1     Running   1           121m
kube-system  coredns-5d78c9869d-cm75d                               1/1     Running   1           128m
kube-system  coredns-5d78c9869d-tpwtd                               1/1     Running   1           128m
kube-system  etcd-master                                             1/1     Running   1           128m
kube-system  kube-apiserver-master                                   1/1     Running   1           128m
kube-system  kube-controller-manager-master                         1/1     Running   2           128m
kube-system  kube-loxilb-796ff78df4-hcrnb                           1/1     Running   1           128m
kube-system  kube-proxy-8w5d9                                        1/1     Running   1           124m
kube-system  kube-proxy-cm2f7                                        1/1     Running   1           128m
kube-system  kube-proxy-mx8rd                                       1/1     Running   1           121m
kube-system  kube-scheduler-master                                  1/1     Running   2           128m
kube-system  metrics-server-754586b847-ks67n                       1/1     Running   3 (43m ago)  128m
vagrant@master:~$ kubectl get services --all-namespaces
NAMESPACE   NAME           TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
default     kubernetes     ClusterIP     172.17.0.1    <none>        443/TCP          130m
kube-system  kube-dns       ClusterIP     172.17.0.10   <none>        53/UDP, 53/TCP, 9153/TCP  130m
kube-system  metrics-server ClusterIP     172.17.21.245 <none>        443/TCP          130m
vagrant@master:~$
```

Source: <https://github.com/loxilb-io/loxilb/actions/runs/5312300896/jobs/9616620226>



LOXILB OPERATIONS

❖ ip operations @ loxilb


-  ip a
- sudo docker ps

```
vagrant@l1b1:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
    altname enp0s3
    inet 10.0.2.15/24 metric 100 brd 10.0.2.255 scope global dynamic eth0
        valid_lft 82327sec preferred_lft 82327sec
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:8c:98:2c brd ff:ff:ff:ff:ff:ff
    altname enp0s8
    inet 192.168.80.9/24 brd 192.168.80.255 scope global eth1
        valid_lft forever preferred_lft forever
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:27:f8:1a brd ff:ff:ff:ff:ff:ff
    altname enp0s9
    inet 192.168.90.9/24 brd 192.168.90.255 scope global eth2
        valid_lft forever preferred_lft forever
5: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:64:93:f3:46 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever
6: llb0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 xdpgeneric/id:24 qdisc fq_codel state UNKNOWN group default qlen 1000
    link/ether ea:5d:05:69:f2:59 brd ff:ff:ff:ff:ff:ff
vagrant@l1b1:~$ sudo docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS   NAMES
66c18a4f7a72  ghcr.io/loxilb-io/loxilb:latest    "/root/loxilb-io/lox...  3 hours ago   Up About an hour   loxilb
vagrant@l1b1:~$
```



VAGRANT SSH FOR LOXILB

❖ Vagrant ssh for loxilb

-  `vagrant global-status`
- `vagrant ssh 5a6c28c` `### loxilb`
- `sudo docker exec -it loxilb bash`
- `ip a`
- `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`

```
PS D:\loxilb\loxilb\ci\k8s-calico> vagrant global-status
id      name      provider  state  directory
-----
5a6c28c loxilb    virtualbox running D:/loxilb/loxilb/ci/k8s-calico
f17541e master    virtualbox running D:/loxilb/loxilb/ci/k8s-calico
7763de1 worker1    virtualbox running D:/loxilb/loxilb/ci/k8s-calico
0cca613 worker2    virtualbox running D:/loxilb/loxilb/ci/k8s-calico
```

The above shows information about all known Vagrant environments on this machine. This data is cached and may not be completely up-to-date (use "vagrant global-status --prune" to prune invalid entries). To interact with any of the machines, you can go to that directory and run Vagrant, or you can use the ID directly with Vagrant commands from any directory. For example: "vagrant destroy 1a2b3c4d"

```
PS D:\loxilb\loxilb\ci\k8s-calico> vagrant ssh 5a6c28c
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-52-generic x86_64)
```

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

```
* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s just raised the bar for easy, resilient and secure K8s cluster deployment.
```

```
https://ubuntu.com/engage/secure-kubernetes-at-the-edge
```


```
Last login: Sat Mar 20 18:04:46 2021 from 10.0.2.2
vagrant@l1b1:~$
```

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



LOXILB OPERATIONS

❖ Operations for loxilb

-  vagrant global-status
- vagrant ssh 5a6c28c ### loxilb
- **sudo docker exec -it loxilb bash**
- **ip a**
- **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**

```
vagrant@l1b1:~$ sudo docker exec -it loxilb bash
root@l1b1:/# loxicmd get lb -o wide
root@l1b1:/# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
    altname enp0s3
    inet 10.0.2.15/24 metric 100 brd 10.0.2.255 scope global dynamic eth0
        valid_lft 84022sec preferred_lft 84022sec
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:8c:98:2c brd ff:ff:ff:ff:ff:ff
    altname enp0s8
    inet 192.168.80.9/24 brd 192.168.80.255 scope global eth1
        valid_lft forever preferred_lft forever
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:27:f8:1a brd ff:ff:ff:ff:ff:ff
    altname enp0s9
    inet 192.168.90.9/24 brd 192.168.90.255 scope global eth2
        valid_lft forever preferred_lft forever
5: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:4c:33:52:13 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever
6: l1b0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 xdpgeneric/id:24 qdisc fq_codel state UNKNOWN group default
    qlen 1000
    link/ether ea:5d:05:69:f2:59 brd ff:ff:ff:ff:ff:ff
root@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
```

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



CICD OPERATIONS @ UBUNTU DESKTOP

❖ Ubuntu Desktop Operations

- `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)
- `chmod +x file_name`
- `common.sh @ upper directory`
- `./config.sh`
- `./validation.sh`
- `./rmconfig.sh`



Go to Settings > Update & Security > For Developers. Check the Developer Mode radio button. And search for "Windows Features", choose "Turn Windows features on or off".

```
sudo su - root
(return with ctrl-d)
```

```
jslab@jslab:~/cicd/scenario1$ sudo su - root
root@jslab:~# cd /home/jslab/cicd/scenario1
root@jslab:/home/jslab/cicd/scenario1#
```

```
jslab@jslab:~/cicd$ sudo su - root
root@jslab:~# cd /home/jslab/cicd
root@jslab:/home/jslab/cicd/$ dir
advconnect      hal                k3s-cilium-cluster  k8s-nat64  README.md          tcpepscale      tcplbmark          wrtcp1b1
cluster1        http2ep            k3s_common.sh       lb6timeout sconnect           tcplb           tcplbmon          wrtcp1b2
cluster2        httpsep            k3s-flannel          lbtimeout  sctpfullnat        tcplbcps        tcplbmon6
cluster3        ipsec1             k3s-flannel-cluster nat64tcp    sctpfullnat12     tcplbdsr1       topctpperf
cluster-intKA   ipsec2             k3s-sctpmh           nat66sctp  sctplb             tcplbdsr2       tcptunlb
cluster-intKA-vip k0s-weave          k8s                   nat66tcp   sctplbdsr          tcplbepmod      udplb
common          k3s-calico         k8s-calico           nat66udp   sctplbmon          tcplbhash       udplbmon
common.sh       k3s-calico-dual-stack k8s-calico-ubuntu22 onearm12   sctponearm         tcplb13dsr      ulclctplb
data-store      k3s-cilium         k8slbsim             perf       sctptunlb          tcplb13dsrha    ulclctplb
```

```
root@jslab:/home/jslab/cicd~# chmod +x common.sh
root@jslab:/home/jslab/loxilb/cicd# cd k8s-calico
root@jslab:/home/jslab/cicd/k8s-calico# dir
configs config.sh node_scripts rmconfig.sh Vagrantfile validation.sh yml
root@jslab:/home/jslab/cicd/k8s-calico# chmod +x rmconfig.sh
root@jslab:/home/jslab/cicd/k8s-calico# chmod +x validation.sh
```



CICD OPERATIONS @ UBUNTU DESKTOP

15

❖ Ubuntu Desktop Operations



- Vagrant Installation with VirtualBox
- Cloning: `sudo git clone https://github.com/loxilb-io/loxilb.git`
- `chmod +x file_name`
- `sudo su - root ###` ('control' + d for exit)
- `common.sh @ upper directory`
- `./config.sh`
- `./validation.sh`
- `./rmconfig.sh`

Source: <https://github.com/loxilb-io/loxilb/blob/main/cicd/k8s-calico/config.sh>

```
#!/bin/bash
VMs=$(vagrant global-status | grep -i virtualbox)
while IFS= read -a VMs; do
  read -a vm <<< "$VMs"
  cd ${vm[4]} 2>&1>/dev/null
  echo "Destroying ${vm[1]}"
  vagrant destroy -f ${vm[1]}
  cd - 2>&1>/dev/null
done <<< "$VMs"

vagrant up
sudo ip route add 123.123.123.1 via 192.168.90.9

for((i=1; i<=60; i++))
do
  fin=1
  pods=$(vagrant ssh master -c 'kubectl get pods -A' 2>/dev/null | grep -v "NAMESPACE")

  while IFS= read -a pods; do
    read -a pod <<< "$pods"
    if [[ ${pod[3]} != *"Running"* ]]; then
      echo "${pod[1]} is not UP yet"
      fin=0
    fi
  done <<< "$pods"
  if [ $fin == 1 ];
  then
    break;
  fi
  echo "Will try after 10s"
  sleep 10
done

#Create default Service
vagrant ssh master -c 'kubectl apply -f /vagrant/yaml/tcp.yml' 2>/dev/null
vagrant ssh master -c 'kubectl apply -f /vagrant/yaml/udp.yml' 2>/dev/null
vagrant ssh master -c 'kubectl apply -f /vagrant/yaml/sctp.yml' 2>/dev/null

#Create onearm Service
vagrant ssh master -c 'kubectl apply -f /vagrant/yaml/top_onearm.yml' 2>/dev/null
vagrant ssh master -c 'kubectl apply -f /vagrant/yaml/udp_onearm.yml' 2>/dev/null
vagrant ssh master -c 'kubectl apply -f /vagrant/yaml/sctp_onearm.yml' 2>/dev/null

#Create fullnat Service
vagrant ssh master -c 'kubectl apply -f /vagrant/yaml/top_fullnat.yml' 2>/dev/null
vagrant ssh master -c 'kubectl apply -f /vagrant/yaml/udp_fullnat.yml' 2>/dev/null
vagrant ssh master -c 'kubectl apply -f /vagrant/yaml/sctp_fullnat.yml' 2>/dev/null
```



CICD OPERATIONS @ UBUNTU DESKTOP

❖ Ubuntu Desktop Operations

- Vagrant Installation with VirtualBox
- Cloning: `sudo git clone https://github.com/loxilb-io/loxilb.git`
- `chmod +x file_name`
- `sudo su - root` ### ('control' + d for exit)
- `common.sh` @ upper directory
- `./config.sh`
- `./validation.sh`
- `./rmconfig.sh`

```
#!/bin/bash
source ./common.sh
echo k8s-calico

if [ "$1" ]; then
  KUBECNF="$1"
fi

# Set space as the delimiter
IFS=
for ((i=0; i<120; i++))
do
  extLB=$(vagrant ssh master -c 'kubectl get svc' 2> /dev/null | grep "top-lb-default")
  read -a strarr <<< "$extLB"
  len=${#strarr[*]}
  if [[ $(len) -lt 6 ]]; then
    echo "Can't find top-lb service"
    sleep 1
    continue
  fi
  if [[ ${strarr[3]} != *"none"* ]]; then
    extIP=${strarr[3]}
    break
  fi
  echo "No external LB allocated"
  sleep 1
done

## Any routing updates ??
sleep 30

echo Service IP : $extIP
echo -e "\nEnd Points List"
echo "*****"
vagrant ssh master -c 'kubectl get endpoints -A' 2> /dev/null
echo "*****"
echo -e "\nSvc List"
echo "*****"
vagrant ssh master -c 'kubectl get svc' 2> /dev/null
echo "*****"
echo -e "\nPod List"
echo "*****"
vagrant ssh master -c 'kubectl get pods -A' 2> /dev/null
echo "*****"
echo -e "\nLB List"
echo "*****"
vagrant ssh loxilb -c 'sudo docker exec -it loxilb loxicmd get lb -o wide' 2> /dev/null
echo "*****"
echo -e "\nEP List"
echo "*****"
vagrant ssh loxilb -c 'sudo docker exec -it loxilb loxicmd get ep -o wide' 2> /dev/null
echo "*****"
```

```
echo -e "\nTEST RESULTS"
echo "*****"
mode=( "default" "onarm" "fullnat" )
top_port=( 55002 56002 57002 )
udp_port=( 55003 56003 57003 )
sctp_port=( 55004 56004 57004 )
code=0
for ((i=0; i<2; i++)); do
  out=$(curl -s --connect-timeout 10 http://$extIP:$top_port[$i])
  if [[ $out == *"Welcome to nginx"* ]]; then
    echo -e "K8s-calico TCP\t($mode[$i])\t[OK]"
  else
    echo -e "K8s-calico TCP\t($mode[$i])\t[FAILED]"
  fi
  ## Dump some debug info
  echo "lb1 lb-info"
  vagrant ssh loxilb -c 'sudo docker exec -it lb1 loxicmd get lb -o wide' 2> /dev/null
  echo "lb1 route-info"
  vagrant ssh loxilb -c 'sudo docker exec -it lb1 ip route' 2> /dev/null
  code=1
fi

out=$(timeout 5 ./common/udp_client $extIP $udp_port[$i])
if [[ $out == *"Client"* ]]; then
  echo -e "K8s-calico UDP\t($mode[$i])\t[OK]"
else
  echo -e "K8s-calico UDP\t($mode[$i])\t[FAILED]"
fi
## Dump some debug info
echo "lb1 lb-info"
vagrant ssh loxilb -c 'sudo docker exec -it lb1 loxicmd get lb -o wide' 2> /dev/null
echo "lb1 route-info"
vagrant ssh loxilb -c 'sudo docker exec -it lb1 ip route' 2> /dev/null
code=1
fi

out=$(timeout 5 ./common/sctp_client 192.168.90.1 34951 $extIP $sctp_port[$i])
if [[ $out == *"server"* ]]; then
  echo -e "K8s-calico SCTP\t($mode[$i])\t[OK]"
else
  echo -e "K8s-calico SCTP\t($mode[$i])\t[FAILED]"
fi
## Dump some debug info
echo "lb1 lb-info"
vagrant ssh loxilb -c 'sudo docker exec -it lb1 loxicmd get lb -o wide' 2> /dev/null
echo "lb1 route-info"
vagrant ssh loxilb -c 'sudo docker exec -it lb1 ip route' 2> /dev/null
code=1
fi
done
exit $code
```


Source: <https://github.com/loxilb-io/loxilb/blob/main/cicd/k8s-calico/validation.sh>



CICD OPERATIONS @ UBUNTU DESKTOP

17

❖ Ubuntu Desktop Operations

-  Vagrant Installation with VirtualBox
- Cloning: `sudo git clone https://github.com/loxilb-io/loxilb.git`
- `chmod +x file_name`
- `sudo su - root ###` ('control' + d for exit)
- `common.sh @ upper directory`
- `./config.sh`
- `./validation.sh`
- `./rmconfig.sh`

Source: <https://github.com/loxilb-io/loxilb/blob/main/cicd/k8s-calico/rmconfig.sh>

```
#!/bin/bash
sudo ip route del 123.123.123.1 via 192.168.90.9
vagrant destroy -f worker2
vagrant destroy -f worker1
vagrant destroy -f master
vagrant destroy -f loxilb
sudo rm loxilb-ip
```

