
Los Alamos
NATIONAL LABORATORY
EST. 1943

Virtualizing the Network for Testing & Development

By: Robby Rollins & Conner Whitfield

Mentors:
Jesse Martinez
Brett Holman

Overview

- Problem & Solution
- Configuring the KVM server
- Establishing the virtual network
- Monitoring the virtual network
- Limitations discovered
- Conclusions

The Problem & Solution

The Problem:

Some LANL HPC environments currently don't have constant real-time network monitoring. This is a common place problem since at SC19 last year SCInet used sFlow, Prometheus, and Grafana to show their network flows. There is also a need for testbed environments at LANL to test network configuration and topology changes before implementing them in the production environment.

The Solution:

Implement a KVM server for virtualizing a test network environment, enabling testing of certain network configurations and services such as sFlow and Prometheus.

Project resources

KVM programs:

- qemu-kvm
- libvirt
- libvirt-python
- libguestfs-tools
- virt-install
- virt-manager
- virt-tools

Images:

- Arista
- Cumulus
- Juniper
- CentOS 8

Other:

- sFlow-rt
- Firefox
- Prometheus
- Grafana

Issues with deploying KVM images

- Installing images
  - Kernel update issue
  - Getting images onto the server
- CentOS 8
  - No console port to connect to while installing ISO file
- Arista
  - Must define the ISO image plus qcow2 file when creating
- Cumulus Linux
  - No issues

ARISTA

HIGH PERFORMANCE COMPUTING

Configuring the KVM server

- Installing images
  - Install CentOS 8 & Arista ISO files
  - Install Cumulus and Arista .vmdk files (WinSCP)
  - Virt-install
- CentOS 8
  - Edit grub file to create a console port
  - No graphics
  - Extra args used to define the console port on the VM
- Arista
  - Convert .vmdk to .qcow2 format
  - Install with .qcow2 file being main drive
  - ISO image is added as a live cd
- Cumulus Linux
  - Install .qcow2 file from website

Configuring the KVM server commands

Cent OS:

```
virt-install  --name centos8test --memory 2048 --vcpus 2 --location
CentOS-8.1.1911-x86_64-dvd1.iso --disk size=20 --os-type=linux
--os-variant=centos7.0 --nographics --extra-args="console=tty0
console=ttyS0,115200n8"
```

Arista:

```
virt-install --name=Aristatest --description "Aristatest" --os-type=linux
--ram=2048 --vcpus=2  --boot cdrom,hd,menu=on --cdrom
/tmp/Aboot-veos-8.0.0.iso --livecd --disk vEOS-4.21.7.1M.qcow2 --import
--nographics
```

Cumulus:

```
virt-install --name CumulusLinuxtest --memory 8192 --vcpus 2 --disk
cumulus-linux-4.1.1-vx-amd64-qemu.qcow2 --import --os-variant debian9
```

Virtual Network Setup Basics

- Bridges are utilized for communications between VMs and host.
- Many types of virtual networks can be made, but we opted for a private isolated network that only includes the VMs and host.
  - Intended to isolate the network from other production networks
- KVM offers all of the command line tools necessary for establishing a network as root:
  - `virsh net-define <VM_NAME>`
  - `virsh net-undefine <VM_NAME>`
  - `virsh net-start <VM_NAME>`
  - `virsh net-autostart <VM_NAME>`
  - `virsh net-destroy <VM_NAME>`
  - `virsh net-edit <VM_NAME>`
- KVM also provides commands that any user can use to connect to existing `virsh` networks:
  - `virsh attach-interface <VM_NAME>`
  - `virsh detach-interface <VM_NAME>`
  - `virsh domiflist <VM_NAME>`

Network Setup Basics - The XML File

Example XML file (`priv.xml`):

```
<network>
  <name>private</name>
  <bridge name="privbr0"/>
  <ip address="192.168.152.1" netmask="255.255.255.0">
    <dhcp>
      <range start="192.168.152.2" end="192.168.152.254"/>
    </dhcp>
  </ip>
  <ip family="ipv6" address="2001:db8:ca2:3::1" prefix="64"/>
</network>
```

Color key:
Network name
Bridge name
Bridge IPv4 info
DHCP range
Bridge IPv6 info

Use `virsh` as root to complete setup:

```
virsh net-define priv.xml
virsh net-start private
virsh net-autostart private
```

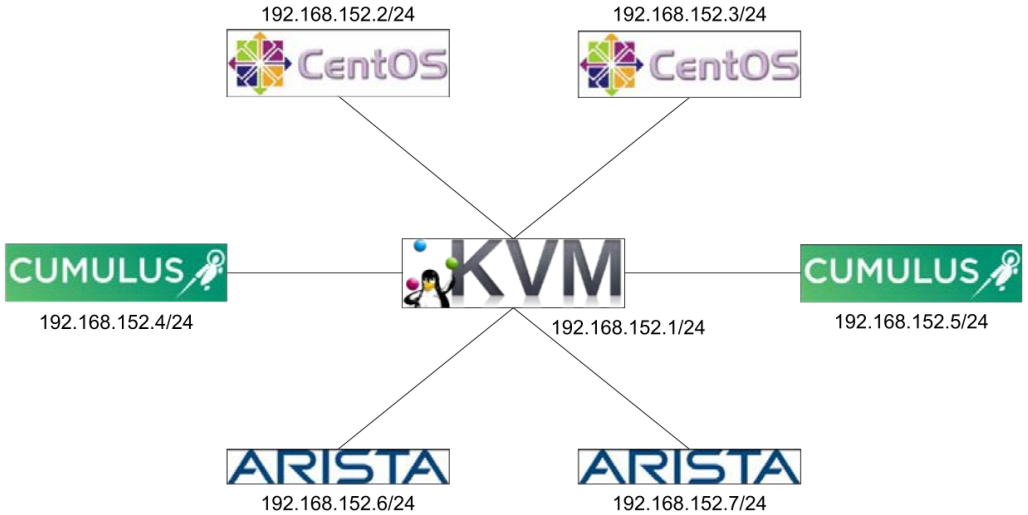
```
privbr0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 192.168.152.1  netmask 255.255.255.0  broadcast 192.168.152.255
        inet6 2001:db8:ca2:3::1  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::5054:ff:fe38:e85c  prefixlen 64  scopeid 0x20<link>
        ether 52:54:00:38:e8:5c  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 14  bytes 1447 (1.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Network Setup Basics - VM Connection

- Once the network “private” is running, each VM attaches an interface that connects to bridge privbr0.

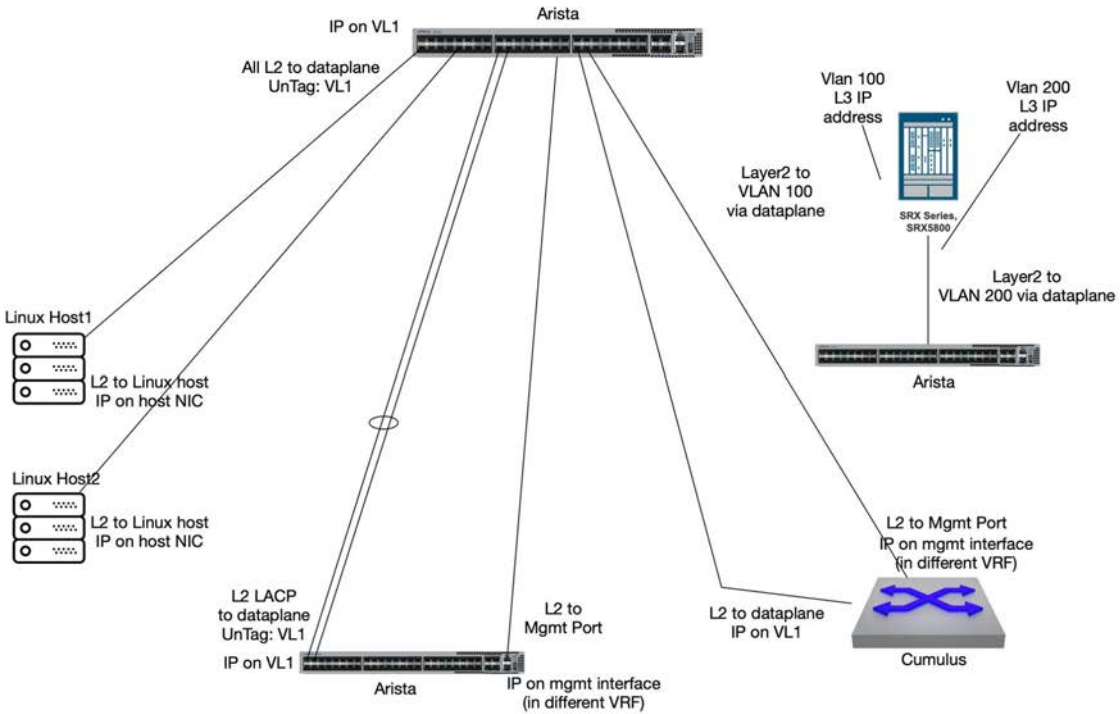
```
1. virsh attach-interface --domain <VM_NAME> --type bridge --source privbr0
   --model virtio --config --live
2. virsh shutdown <VM_NAME>
3. virsh destroy --graceful <VM_NAME>
4. virsh start <VM_NAME>
```

- After this setup process, the following network is made:



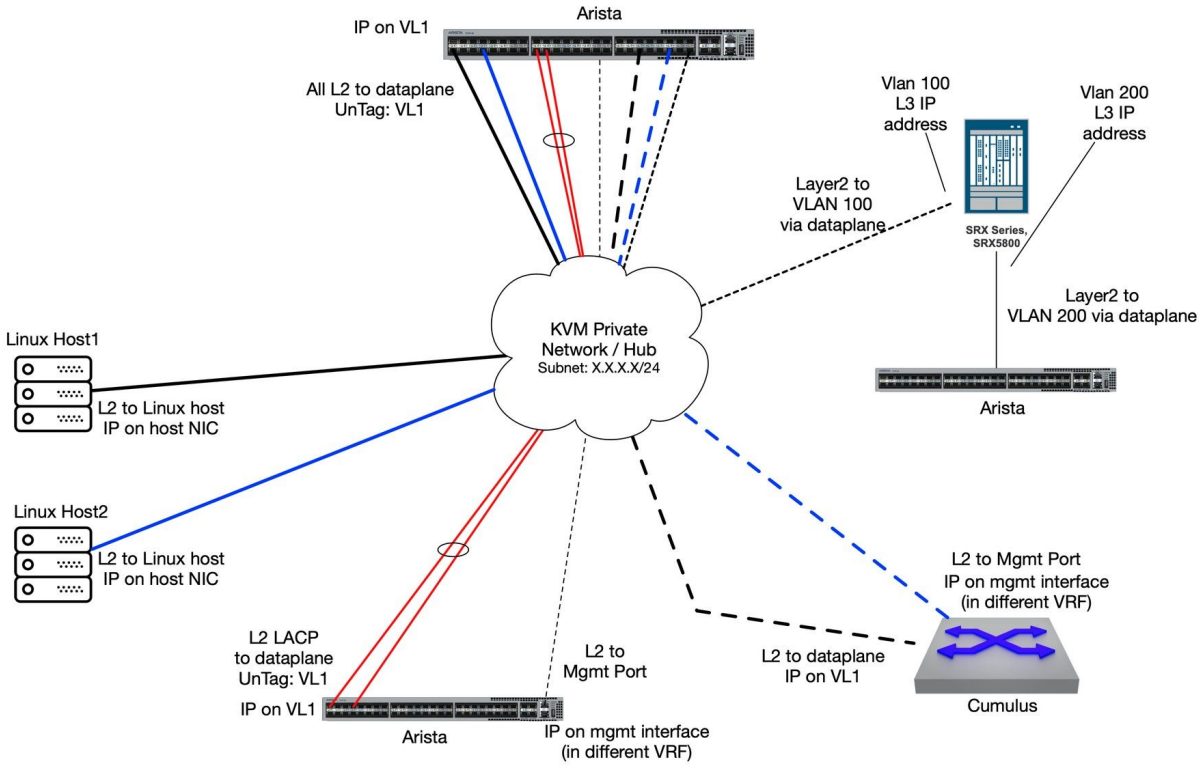
A More Advanced Network

After the basics of KVM network interaction were understood, we had to design a network that resembles a production network.



A More Advanced Network

The same network, but reformatted to indicate how KVM is integrated.



A More Advanced Network - Network Setup Steps

- For each connection between two devices in the network diagram, a new bridge needs to be made.
- New minimal XML template for defining each bridge:

```
<network>
    <name>foo</name>
    <bridge name="foobr0"/>
</network>
```

Color key:
Network name
Bridge name

We can modify this XML template then use `virsh net-define` and `virsh net-start` to initialize each network. `virsh attach-interface` is again used to connect VMs to the correct bridges.

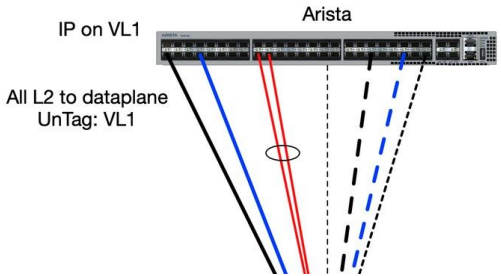
Network name	Bridge name
Host1-AristaTop	H1ATbr0
Host2-AristaTop	H2ATbr0
AristaBottom-AristaTop	ABATbr0
AristaBottom-AristaTop2	ABAT2br0
mgmt-AristaBottom-AristaTop	mgmtABATbr0
Cumulus-AristaTop	CATbr0
mgmt-Cumulus-AristaTop	mgmtCATbr0

A More Advanced Network - Network Setup Steps

Arista vEOS setup with IP address 192.168.153.5/24 on VLAN 2. Every connection is a VLAN 2 access switchport.

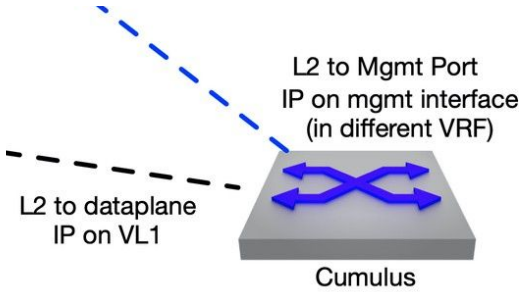
Four connections are missing due to limitations:

- Cumulus Linux mgmt port
- Arista mgmt port
- Channel-port bond to the bottom Arista switch
- Juniper firewall



Use `net add interface <name> <IP>` to establish a working interface. IP address set to 192.168.153.7/24.

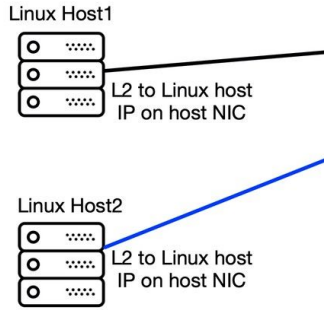
Missing mgmt port connection seen in the diagram.



`/etc/network-scripts/ifcfg-enp7s0` was edited on each CentOS VM to designate the IP addresses.

Host 1 configured with IP address 192.168.153.1/24.

Host 2 configured with IP address 192.168.153.28/24.

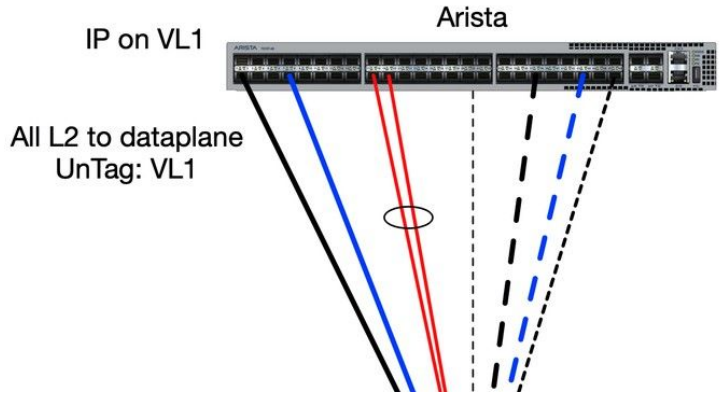


Network Metrics

Network metrics were then collected from the network configuration.

The main Arista switch was configured to send sFlow sample data to the KVM host via KVMATbr0, a new bridge connection on network KVM-AristaTop. IP address 192.168.153.99/24 was assigned to the KVM Host.

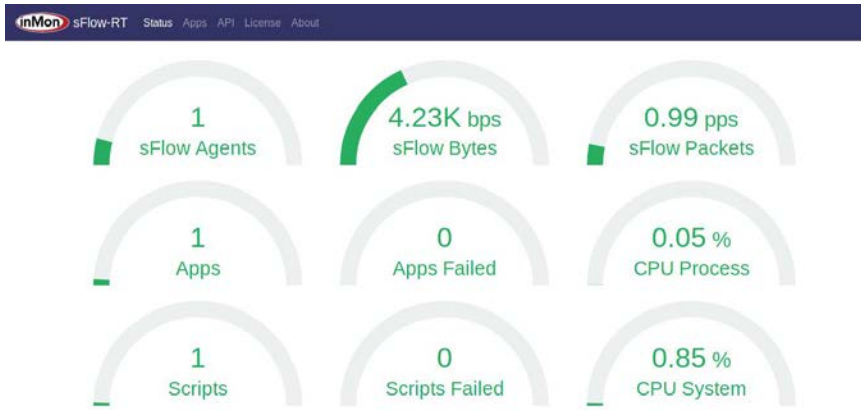
The KVM host ran sFlow-RT to receive sFlow data and used the Prometheus add-on to output data that is usable in Prometheus and Grafana.



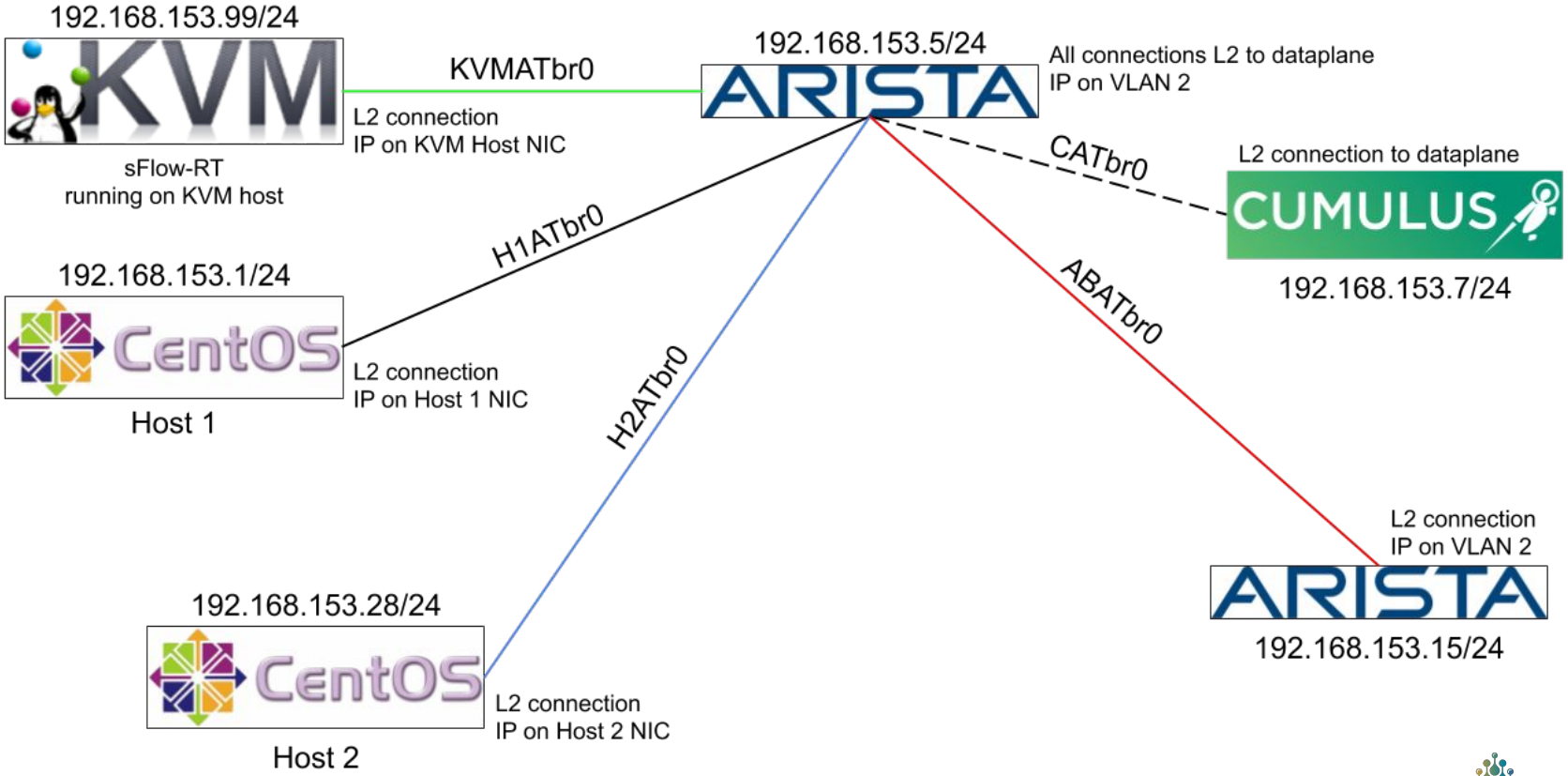
Current State of Network Metrics

sFlow sample packets were received on the KVM host. Initially, sFlow-RT could not detect the packets. After a variety of troubleshooting methods were attempted, it was discovered that firewalld was blocking the traffic.

SFlow integration with Prometheus allowed for network monitoring with analytic tools such as Grafana.



The Final Network



Conclusions & Future Work

- Conclusions:
  - Virtualization is an effective method of testing most network configurations.
  - Multiple users can connect to the same bridges, making collaboration smoother.
- Limitations:
  - Kernel restrictions on network bridges prohibit implementation of port channels and LACP bonded connections.
  - The `virsh console` command prevents connecting VM mgmt ports to the network.
- Future work:
  - Fully integrate Grafana with Prometheus/sFlow-RT to observe metrics.
  - Include Juniper firewall in network design.

Acknowledgements

- Mentors:
  - Jesse Martinez
  - Brett Holman
- Special Thanks:
  - Thomas Areba
  - Chuck Wilder
  - Marc Santoro
  - Julie Wiens
- Entire HPC Division

Questions?

https://dilbert.com/strip/2013-04-07

References:

https://blog.sflow.com/2019/11/sc19-scinet-grafana-network-traffic.html

Over 70 years at the forefront of supercomputing

What is KVM?

- Developed in 2006
- Became apart of Linux in 2007 making it a type-1 hypervisor
  - Type-1: A hypervisor that runs in the OS and can be installed on a bare-metal server (e.g. Hyper-V, Linux, and VSphere)
  - Type-2: A hypervisor software that is ran on top of the OS (e.g. VMware workstation, Microsoft virtual PC, and VirtualBox)
- Open-source virtualization technology
- Uses a command line interface or graphical user interface