AFSTUDEERPROJECT TOEGEPASTE INFORMATICA
Schooljaar 2019-2020

Virtualisatie en Simulatie van het UCLL core netwerk d.m.v. Cisco VIRL

Auteurs:

Arne Bauters (r0703015)

Dieter Maes (r0653749)

Wietse Vandeput (r0632785)

Promotor:

Pieter Geens

Contents

1 Intro

1.1 Assignment

1.1.1 Original assignment

Our assignment was to virtualise/simulate the UCLL core network using Cisco VIRL. With this virtualized setup, changes can be tested before being pushed to production. We were asked whether VIRL could be a viable solution to get a test environment, or not. The goal is to get a working network - using technologies like VRF Lite, BGP, OSPF, QoS - in VIRL.

1.1.2 Concrete guidelines

Like always the assignment diverts slightly from expectations. After a couple of sessions with Pieter Geens we defined our scope as following:

We are to research VIRL, it’s possible limitations as to virtualise the network and the pro’s and con’s of the free online Cisco DevNet versus the payed license for running it on own hardware. The research has to be documented so those willing to use VIRL at the UCLL can easily start doing work instead of figuring out how to use VIRL. The best way to research this, is to actually implement (a part of) the core network. The network diagrams were delivered, together with a lot of information, so we have everything to deliver an attempt at virtualizing of this network. This document will thus mainly research how to do certain things like VRF Lite, Etherchannel, multi-area OSPF, etc. in Cisco VIRL.

2 Pre-setup Research

2.1 VIRL via DevNet

The Cisco DevNet sandbox is a free sandbox environment where you can test multiple things, including VIRL. A closed environment with readily available images for routers, switches, firewalls and servers, can be reserved for a limited time and with limited resources. We noticed reserving resources for a couple of hours (enough for a full workday) was fairly simple and without much waiting and after a couple of minutes you could connect to your environment trough VPN and have access to multiple simulations. For reservations longer than one day, with a maximum of one week, your reservation would be in the queue for a few hours.

We started by testing the main technologies we knew the core network would need: Etherchannel, OSPF, VRF Lite, vPC, AVA clustering, etc. We did have some trouble with vPC not working on the nx switches but resolved it by reserving a different environment that did already have the NX-OSv9K switches included. Another solution would have been download the image of the NX-OSv9K switch online and import it in the Web interface. (More over adding images later in section 3.2.8)

2.2 Alternatives to VIRL

Aside from the comparison between the free, online, Cisco DevNet sanbox and the paid license for the less-limited self-hosted variant of VIRL, we also made the comparison with a few other platforms.

2.2.1 dCloud

dCloud is an alternative to Cisco DevNet and the paid license, that is also online and free, like Cisco DevNet. It does not require you to reserve your environment, but this comes at the cost of a more limited environment. We were only able to use up to 2 machines, which made it pretty useless in our case.

2.2.2 GNS3

GNS3 is an open-source solution to the whole VIRL platform. It is also not vendor-specific. However, we were not able to quickly find free images for Cisco networking gear. (they need to be imported, since GNS3 being open-source and vendor-unspecific means they can not provide free images from vendors) Combined with statements that GNS3 had some troubles with Layer2 functionality in the past, and the uncertainty if they had already been fixed, we decided to not further explore this option.

If we had images available this could have been a great alternative, provided that Layer2 functionality is better than in the past.

3 VIRL Web interface user guide

3.1 Cisco DevNet specific instructions

3.1.1 Cisco DevNet Webservice

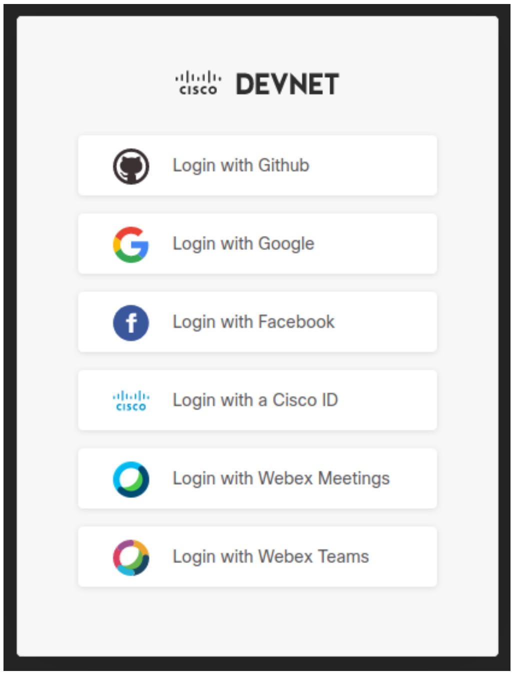
The webservice provides free resources and a free VIRL test environment, but it comes with some limitations:

- You will be limited in the amount of machines you can use (only 12).
- You will have to reserve the resources, with your reservation being limited in time. (They do however state you can contact support to extend your reservation, if needed)

`https://devnetsandbox.cisco.com`

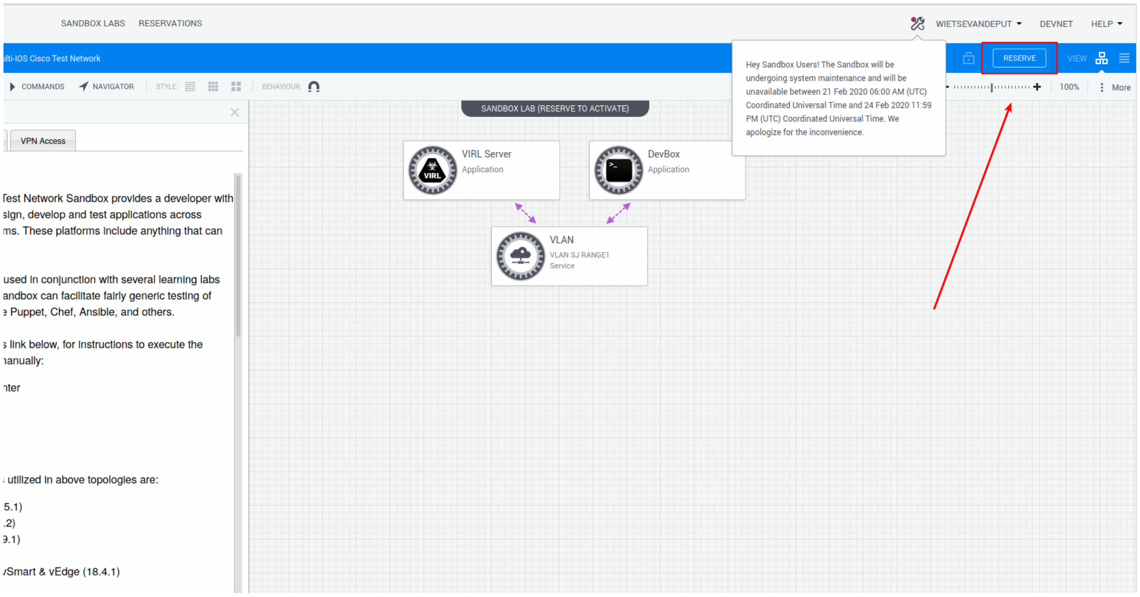
3.1.2 Login to Devnet

Obviously, a login is required. Luckily some options for sign-on with external providers as well as other Cisco services is available.



3.1.3 The Devnet sandbox labs

The environment offers a lot of services, but the scope of this project limits us to the VIRL sandbox. To get this specific sandbox running you have to search the sandbox labs page for either "VIRL" or "Multi-IOS Cisco Test Network". This page already provides a lot of information on how to use the sandbox, even more is given by mail once you reserve the resources. To do this you will have to click *reserve* in the upper-left corner.



Hey Sandbox Users! The Sandbox will be undergoing system maintenance and will be unavailable between 21 Feb 2020 06:00 AM (UTC) Coordinated Universal Time and 24 Feb 2020 11:59 PM (UTC) Coordinated Universal Time. We apologize for the inconvenience.

Test Network Sandbox provides a developer with sign, develop and test applications across ms. These platforms include anything that can

used in conjunction with several learning labs andbox can facilitate fairly generic testing of e Puppet, Chef, Ansible, and others.

s link below, for instructions to execute the nanually:

nter

utilized in above topologies are:

5.1)
.2)
9.1)

vSmart & vEdge (18.4.1)

3.1.4 Reserving resources

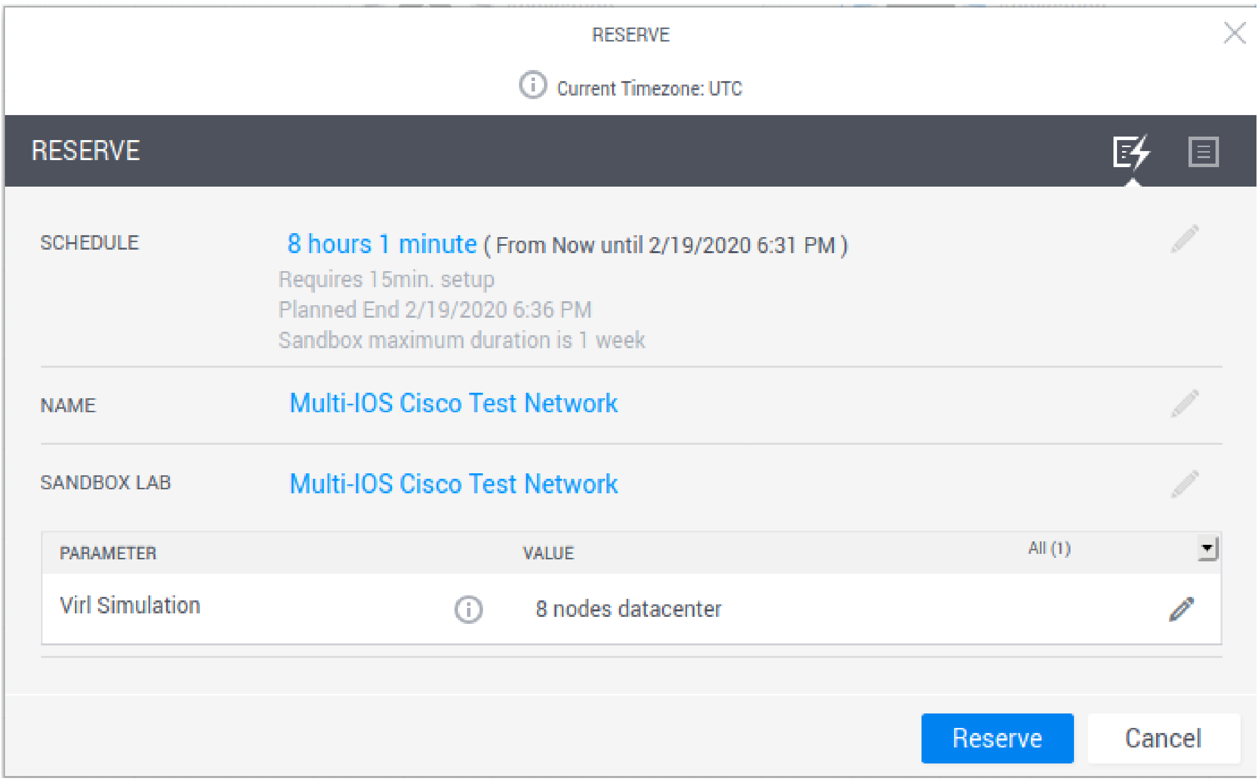
In the reservation screen you can define a couple of things:

Schedule Define for how long you would like to reserve your test environment. The longer you would want to keep your reservation, the longer it will take before there is a slot available and the resources are reserved.

Name This very self-explanatory, you can give your sandbox a custom name.

Sandbox Lab You can still change what sandbox you would like to use.

VIRL Simulation Here you can choose to already start a predefined simulation, or just start with a clean slate ("none")



RESERVE

Current Timezone: UTC

RESERVE

SCHEDULE: 8 hours 1 minute (From Now until 2/19/2020 6:31 PM)
Requires 15min. setup
Planned End 2/19/2020 6:36 PM
Sandbox maximum duration is 1 week

NAME: Multi-IOS Cisco Test Network

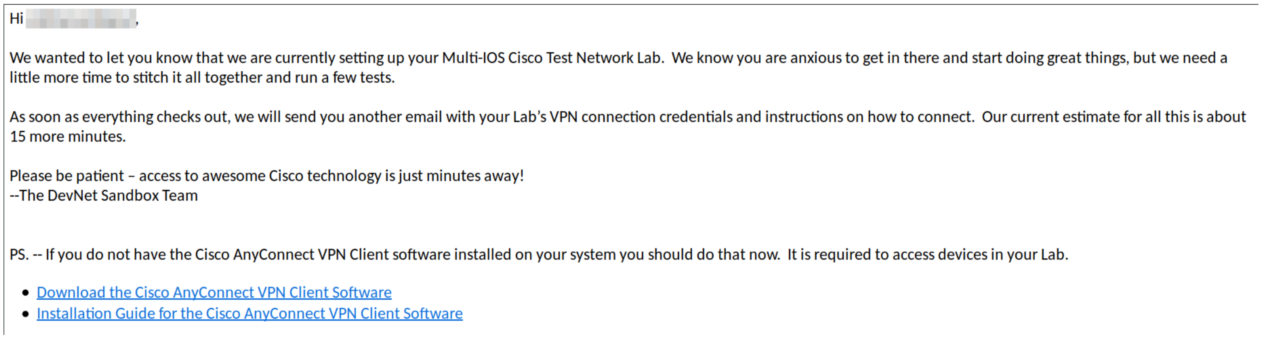
SANDBOX LAB: Multi-IOS Cisco Test Network

PARAMETER	VALUE
Virl Simulation	8 nodes datacenter

There are also a couple of advanced settings like "owner" and "description" for optimal customization.

3.1.5 Mails after reservation

After reserving the resources and the sandbox environment you will receive 2 mails. One telling you they are currently setting up te sandbox and the time they expect it to take. Also informing you that you need to connect to it with a VPN. For Windows they provide a link to their free AnyConnect VPN Client, on Linux you will need to use OpenConnect.



Hi [redacted],

We wanted to let you know that we are currently setting up your Multi-IOS Cisco Test Network Lab. We know you are anxious to get in there and start doing great things, but we need a little more time to stitch it all together and run a few tests.

As soon as everything checks out, we will send you another email with your Lab's VPN connection credentials and instructions on how to connect. Our current estimate for all this is about 15 more minutes.

Please be patient – access to awesome Cisco technology is just minutes away!
--The DevNet Sandbox Team

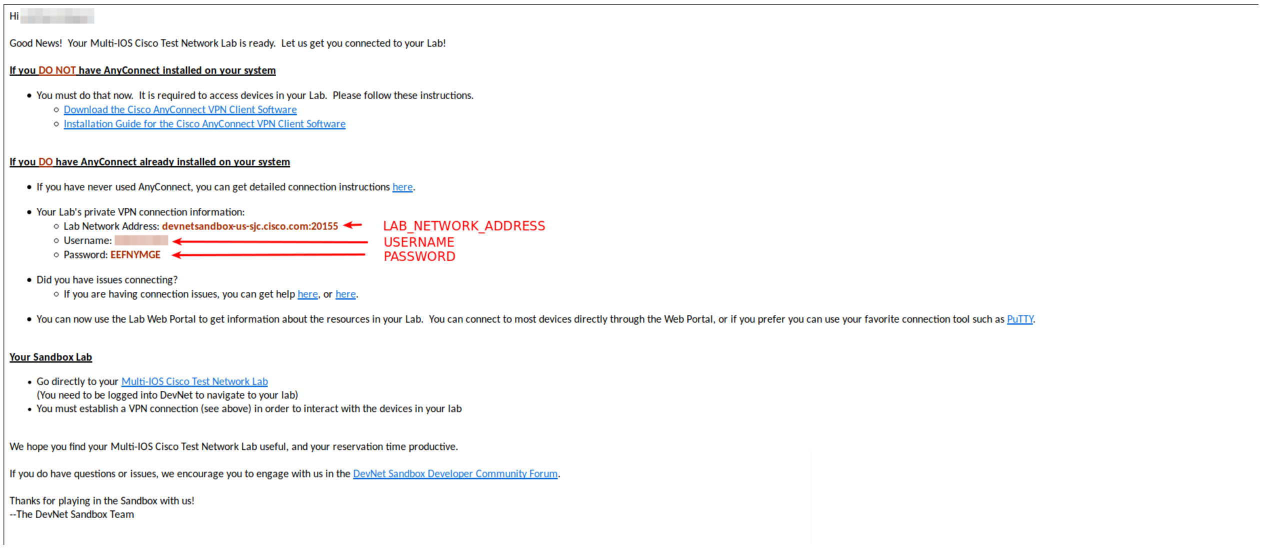
PS. -- If you do not have the Cisco AnyConnect VPN Client software installed on your system you should do that now. It is required to access devices in your Lab.

- Download the Cisco AnyConnect VPN Client Software
- Installation Guide for the Cisco AnyConnect VPN Client Software

The full Linux command is

```
sudo openconnect --script /usr/share/vpnc-scripts/vpnc-script LAB_ADDRESS
```

The *LAB_ADDRESS* will be provided in the next mail, together with the username and the password you will be asked for.



Hi [redacted]

Good News! Your Multi-IOS Cisco Test Network Lab is ready. Let us get you connected to your Lab!

If you DO NOT have AnyConnect installed on your system

- You must do that now. It is required to access devices in your Lab. Please follow these instructions.
  - Download the Cisco AnyConnect VPN Client Software
  - Installation Guide for the Cisco AnyConnect VPN Client Software

If you DO have AnyConnect already installed on your system

- If you have never used AnyConnect, you can get detailed connection instructions here.
- Your Lab's private VPN connection information:
  - Lab Network Address: **devnetsandbox-us-sjc.cisco.com:20155**
  - Username: [redacted]
  - Password: **EEFNYMGE**
- Did you have issues connecting?
  - If you are having connection issues, you can get help here, or here.
- You can now use the Lab Web Portal to get information about the resources in your Lab. You can connect to most devices directly through the Web Portal, or if you prefer you can use your favorite connection tool such as PuTTY.

Your Sandbox Lab

- Go directly to your Multi-IOS Cisco Test Network Lab
  (You need to be logged into DevNet to navigate to your lab)
- You must establish a VPN connection (see above) in order to interact with the devices in your lab

We hope you find your Multi-IOS Cisco Test Network Lab useful, and your reservation time productive.

If you do have questions or issues, we encourage you to engage with us in the DevNet Sandbox Developer Community Forum.

Thanks for playing in the Sandbox with us!
--The DevNet Sandbox Team

With these you can connect to the VPN and through that to `http://10.10.20.160` in your browser of choice. You will arrive in the sandbox overview and from here on, there is little to no difference between the VIRL environment provided by DevNet and the VIRL PE.

3.2 The Web Interface

As explained earlier from now on the use of the Web interface will be the same for both the self-hosted VIRL PE and the free DevNet Solution.

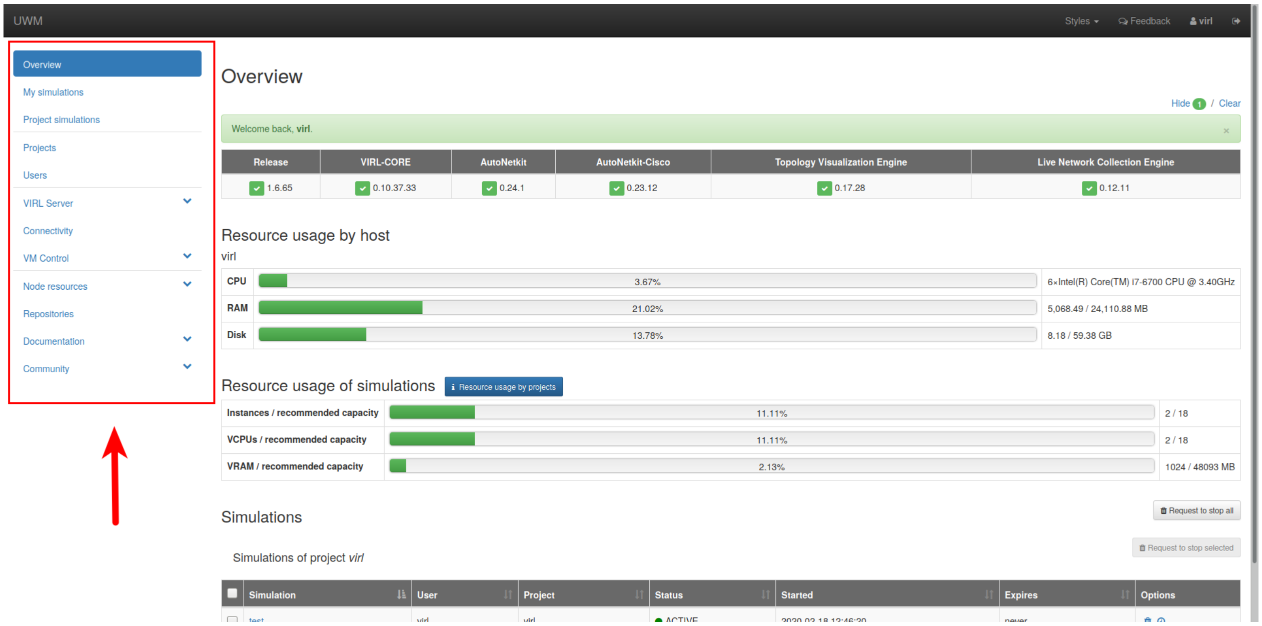
3.2.1 Overview

When connected to the VIRL Server you can navigate to the UWM (User Workspace Management) page. This will be your Web interface where you will be able to do all your virtualization and monitoring.

On the left side you find a couple of tabs:

- The "Overview" page will give you a representation of the resources available, the resources being used and a quick overview of your active simulations.
- The "My Simulations" page will be fully dedicated to your simulations. You get an overview of all the active simulations and resource usage of specific users.
- The "Project Simulations" and "Projects" display the projects and their simulations (obviously). Projects get created and named after users automatically. Each User has a password and a role within the project.
- The "Users" page is quite self-explanatory and thus displays all the users on the server.
- The "VIRL Server" page is where you can update your license, find some tools and downloads.
- The "Connectivity" tab gives you information about the network and how to connect to certain projects.
- The "VM Control" page gives you all the information about every node, network, connection, host and port currently being virtualized on the server.
- The "Node Resource" page will give you all information about the different nodes you can use in the virtualization. Information about the images, etc.

- The "Repository" page gives you the option to connect to a GIT repository.
- The "Documentation" and the "Community" pages are for additional information either provided by the developers or the community.



Overview

Welcome back, **virl**.

Release	VIRL-CORE	AutoNetkit	AutoNetkit-Cisco	Topology Visualization Engine	Live Network Collection Engine
✓ 1.6.65	✓ 0.10.37.33	✓ 0.24.1	✓ 0.23.12	✓ 0.17.28	✓ 0.12.11

Resource usage by host

virl

CPU	3.67%	6×Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz
RAM	21.02%	5,068.49 / 24,110.88 MB
Disk	13.78%	8.18 / 59.38 GB

Resource usage of simulations

Instances / recommended capacity	11.11%	2 / 18
VCPUs / recommended capacity	11.11%	2 / 18
VRAM / recommended capacity	2.13%	1024 / 48093 MB

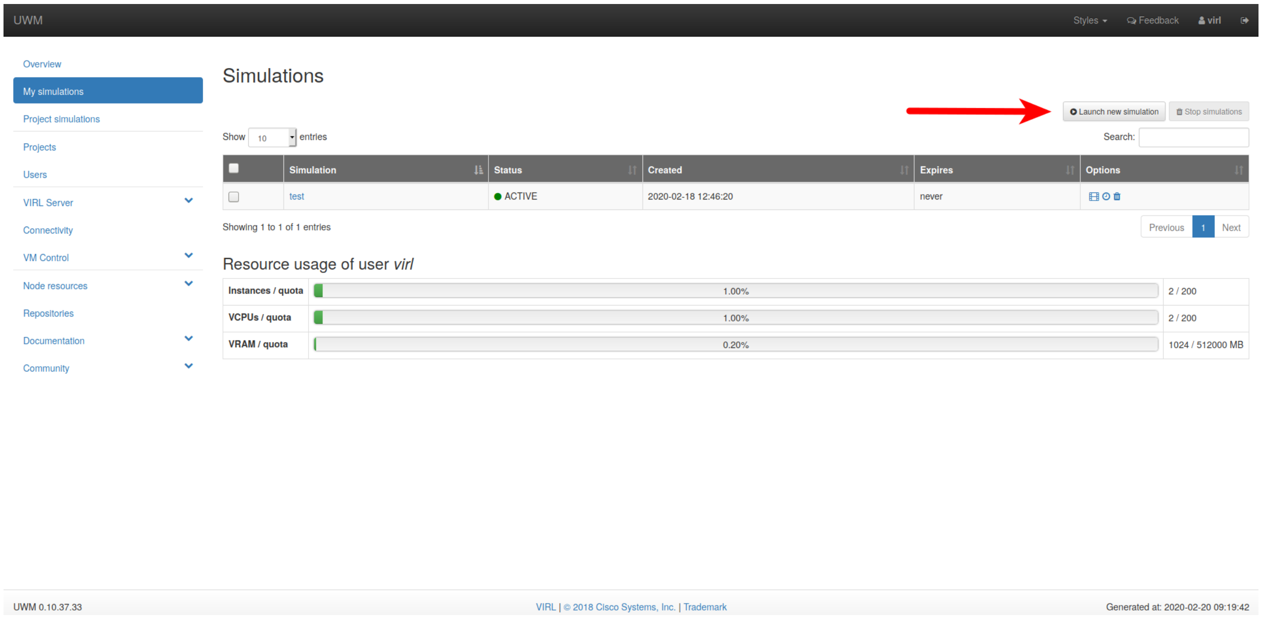
Simulations

Simulations of project *virl*

□	Simulation	User	Project	Status	Started	Expires	Options
□	test	virl	virl	● ACTIVE	2020-02-18 12:46:20	never	

3.2.2 Make a simulation

To launch a new simulation is actually very easy and rather fast. Start by clicking the "Launch new simulation" button on the "Simulations" page.



Simulations

Show 10 entries

Search:

□	Simulation	Status	Created	Expires	Options
□	test	● ACTIVE	2020-02-18 12:46:20	never	

Showing 1 to 1 of 1 entries

Resource usage of user *virl*

Instances / quota	1.00%	2 / 200
VCPUs / quota	1.00%	2 / 200
VRAM / quota	0.20%	1024 / 512000 MB

UWM 0.10.37.33

VIRL | © 2018 Cisco Systems, Inc. | Trademark

Generated at: 2020-02-20 09:19:42

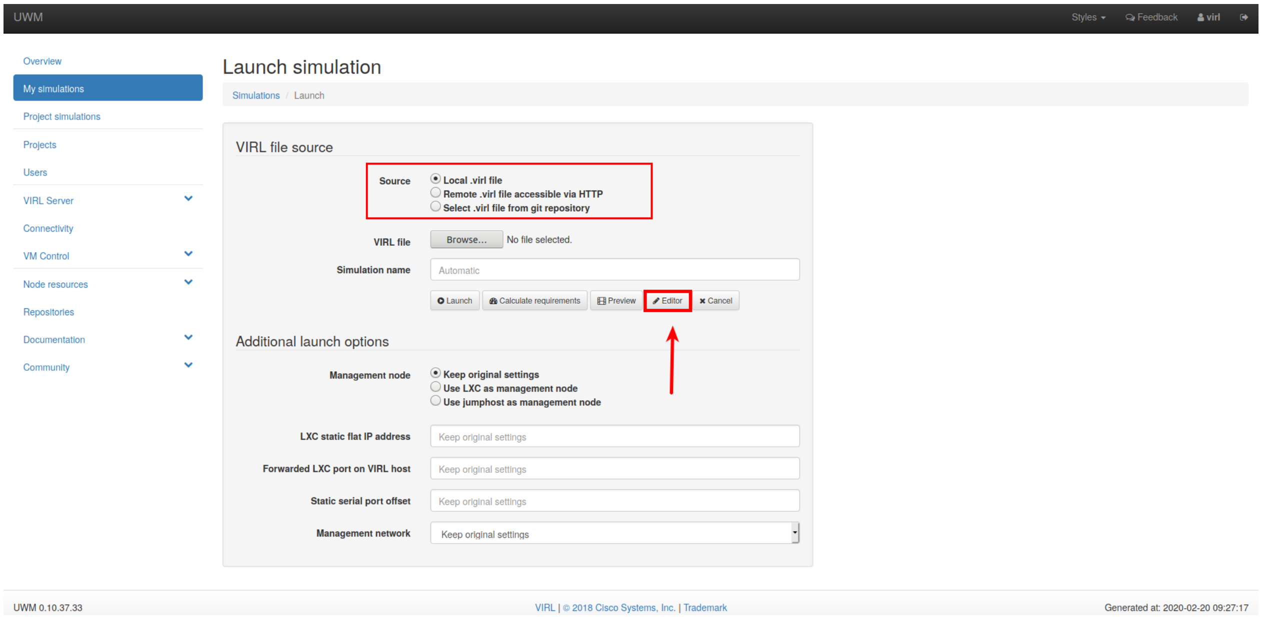
Each simulation will take up resources as long as they are active. As far as we know there is no option to disable them other than to stop them completely. If you are done with one simulation we thus advise to get the full configuration file as well as the original .virl file and save them either locally or on git. You will be able to re-simulate already existing .virl files very easily.

Next you will have to choose the source of the simulation:

Local .virl file: You can use this if you already have a .virl file that you want to launch.

Remote .virl file: You can also simulate .virl files that are accessible either via HTTP or from a git repository.

Editor: This one is not so clearly stated but you can also make a new .virl file in the editor. To do this you have to click the "editor" button, this will bring you to the VIRL editor (more of this in the next section).



Launch simulation

VIRL file source

Source
● Local .virl file
○ Remote .virl file accessible via HTTP
○ Select .virl file from git repository

VIRL file: No file selected.

Simulation name: Automatic

Additional launch options

Management node
● Keep original settings
○ Use LXC as management node
○ Use jumphost as management node

LXC static flat IP address: Keep original settings

Forwarded LXC port on VIRL host: Keep original settings

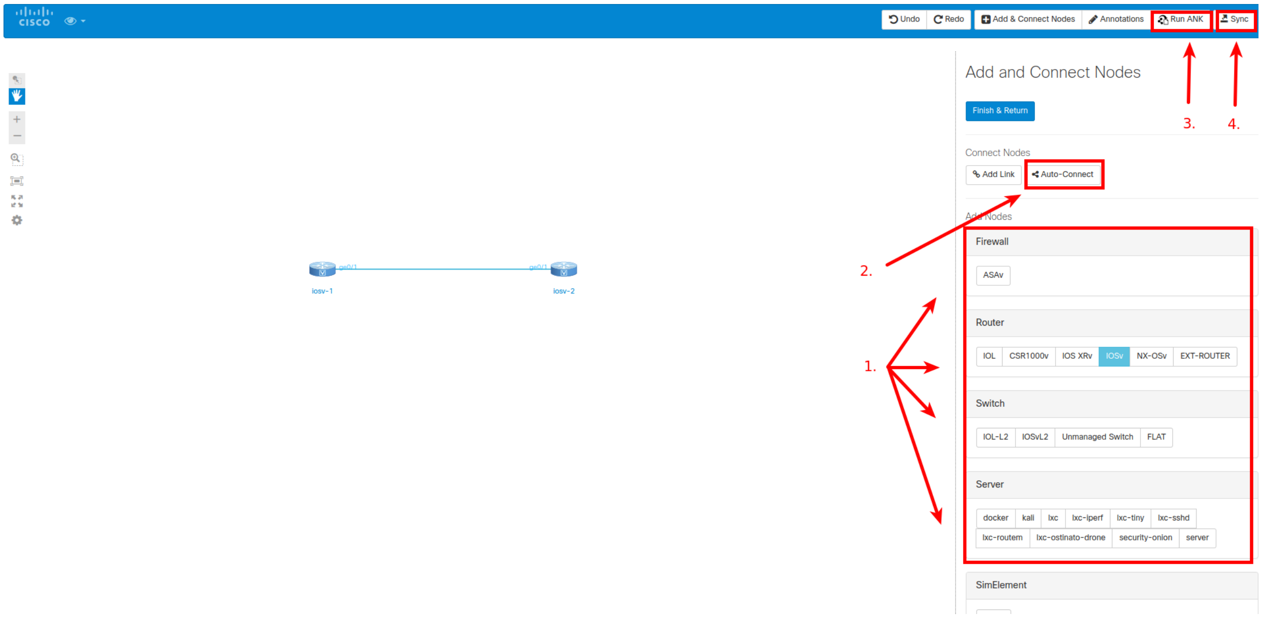
Static serial port offset: Keep original settings

Management network: Keep original settings

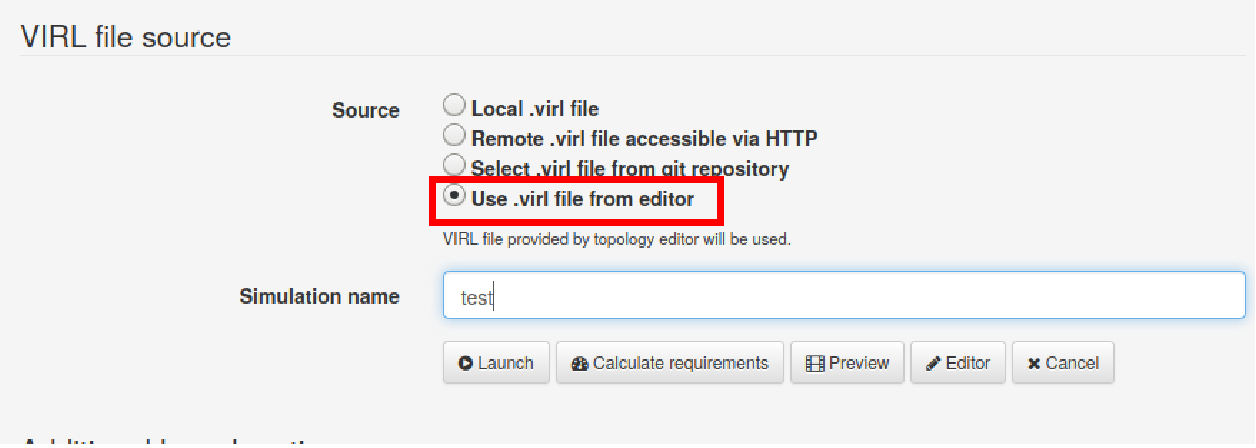
UWM 0.10.37.33 VIRL | © 2018 Cisco Systems, Inc. | Trademark Generated at: 2020-02-20 09:27:17

Now that you have chosen a source for your simulation all you have to do is to give it a name and launch it.
There are some additional launch options you can edit but we have little information about this as we didn't need any alterations for these.

3.2.3 The VIRL editor



1. Here you can find all different nodes to add to you topology. Just select the node you want to add and click anywhere in your editor to place.
2. You can add links manually but you can also auto-connect all devices.
3. You can run ANK after everything seems to be alright. This will auto configure basic configuration on all devices. This is very useful for tests as you don't have to do the basic configuration anymore. If you like to test everything or don't like basic configuration you do not have to click this for most nodes. If you use a server node anywhere in your topology you are required to push this ANK button otherwise you will not have access to these server nodes.(You can always delete the configuration on the other nodes.) In the next chapter we will cover ANK.
4. When you are all set you need to push "Sync" in the upper right corner. This will sync the .virl file to the UWM Simulations> Launch new simulation page. there will be a new option by source: "Use .virl file from editor"



VIRL file source

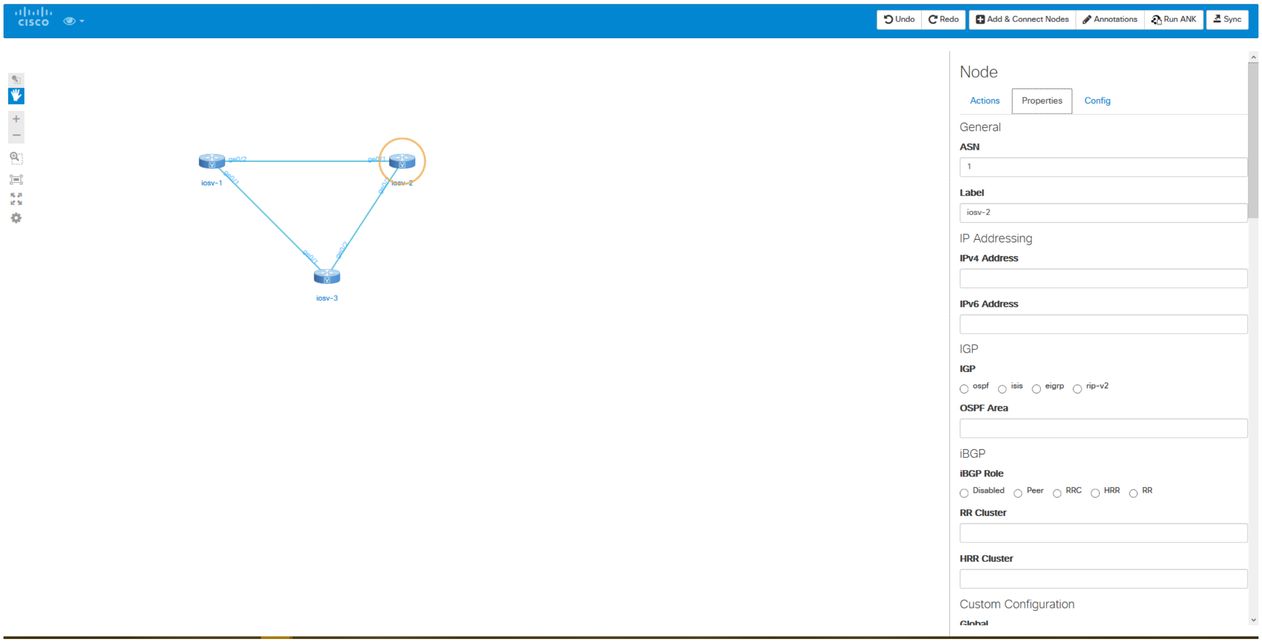
Source
○ Local .virl file
○ Remote .virl file accessible via HTTP
○ Select .virl file from git repository
● Use .virl file from editor

VIRL file provided by topology editor will be used.

Simulation name: test

3.2.4 Autonetkit ANK

AutoNetKit is a tool that can configure the network automatically. This is very handy to create a simple configuration for your topology. It automatically adds protocols, allocates IP addresses and generates configuration files with technologies like BGP, OSPF and many more.



Node

General

ASN: 1

Label: iosv-2

IP Addressing

IPv4 Address:

IPv6 Address:

IGP

IGP: ○ ospf ○ isis ○ eigrp ○ rip-v2

OSPF Area:

iBGP

iBGP Role: ○ Disabled ○ Peer ○ RRC ○ HRR ○ RR

RR Cluster:

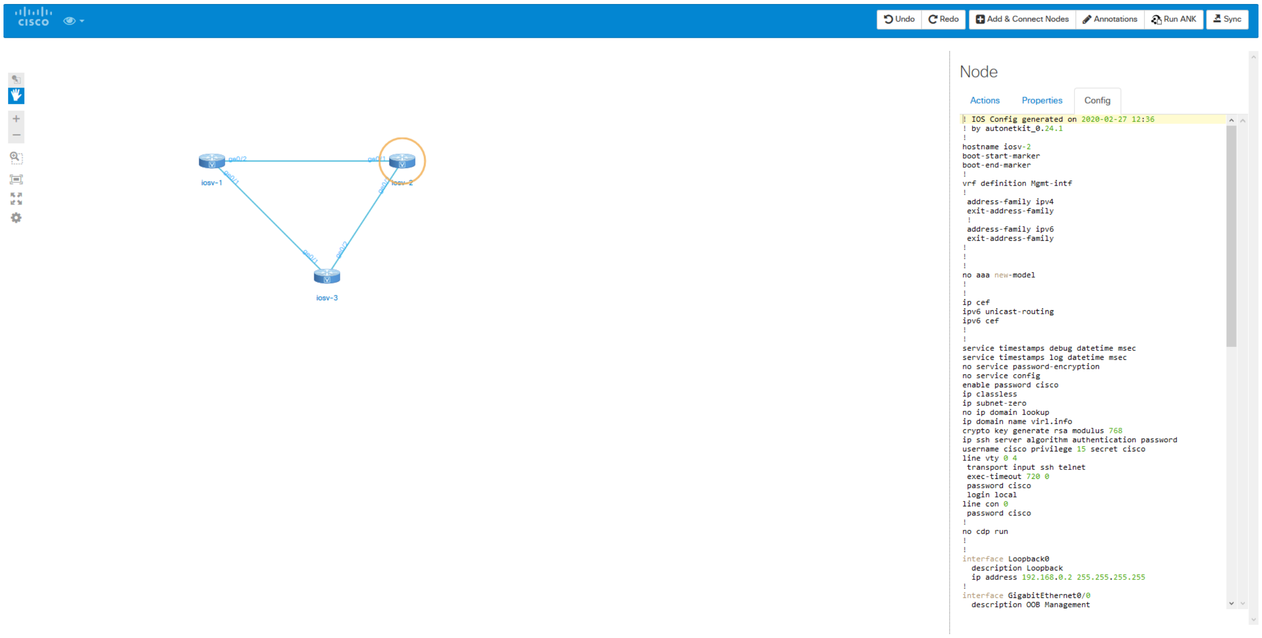
HRR Cluster:

Custom Configuration

Global

When you click on a device like an OSv router and select the properties pane, you see multiple fields where you can fill in a bunch of properties like the ipv4-address, bgp configuration, OSPF area's . . . This can also be done for multiple devices at the same time by holding down the SHIFT button while clicking the devices.

When you're done filling in your preferences you can click on ANK in the upper right corner. Now you can see the config of the device in the config pane. Here you can also remove parts or edit the configuration to your liking. However sometimes when you fill in your preferences like ipv4-addresses ANK will override this and not use your configuration. This often happened in our project. We ran ANK to get a basic configuration and removed the parts of the config we didn't need or want.



```
! IOS Config generated on 2020-02-27 12:36
! by autonetkit_0.24.1
!
hostname iosv-2
boot-start-marker
boot-end-marker
!
vrf definition Mgmt-intf
!
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
!
!
no aaa new-model
!
!
ip cef
ipv6 unicast-routing
ipv6 cef
!
!
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
no service config
enable password cisco
ip classless
ip subnet-zero
no ip domain lookup
ip domain name virl.info
crypto key generate rsa modulus 768
ip ssh server algorithm authentication password
username cisco privilege 15 secret cisco
line vty 0 4
 transport input ssh telnet
 exec-timeout 720 0
 password cisco
 login local
line con 0
 password cisco
!
no cdp run
!
!
interface Loopback0
  description Loopback
  ip address 192.168.0.2 255.255.255.255
!
interface GigabitEthernet0/0
  description OOB Management
```

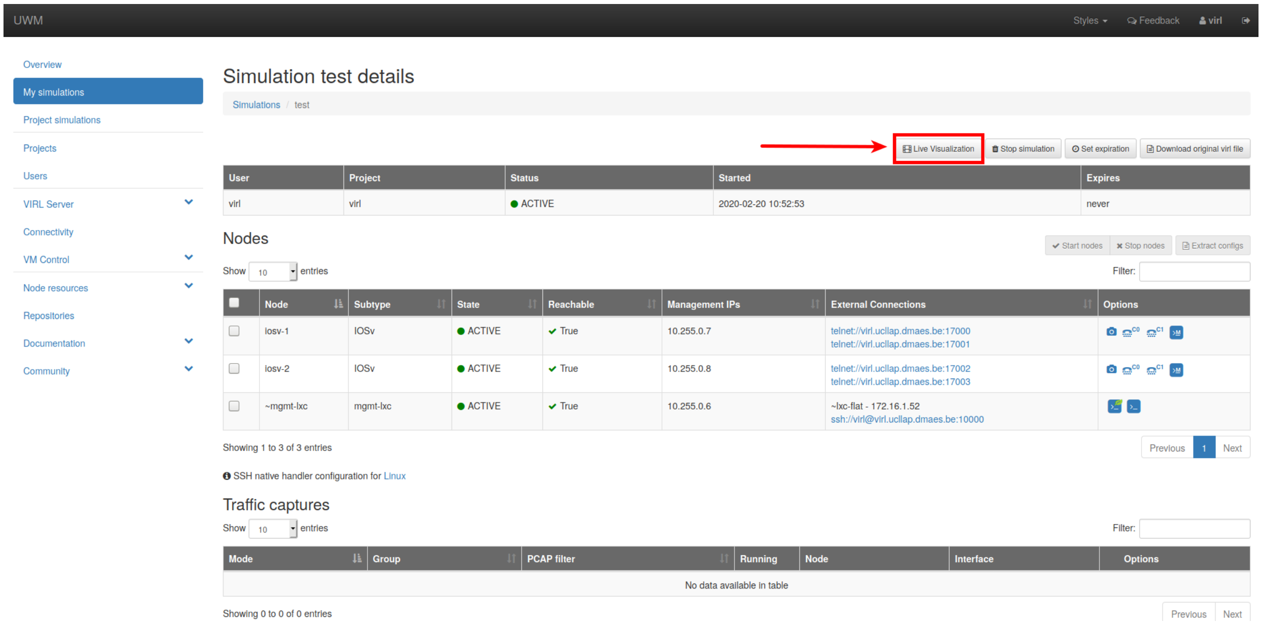
When you use servers you are forced to run ANK on these devices, otherwise there are no usernames and passwords configured and you will be unable to gain access to the devices. This can't be done manually. We used ANK for creating the basic configuration so we didn't need to manually type the same things over and over again. Thus saving time for the more complex configurations.

3.2.5 The Simulation

Congratulations, you now have made a simulation in the VIRL Web interface. The nodes will take a moment to initialize but once everything is active and reachable you are able to start playing around with your network.

You will be send to the Simulation "test" details page. You can get back to this page by going to "My Simulations" and clicking on the simulation you want to view the details of.

On this page you see a number of things, ranging from the nodes overview (with the links and information how to connect to these), traffic captures, interfaces, links and simulation messages. If you want to start playing around with your network you can connect to any node you want using the telnet commands or you can click "Live Visualization".



Simulation test details

User	Project	Status	Started	Expires
virl	virl	● ACTIVE	2020-02-20 10:52:53	never

Nodes

Show 10 entries

Filter:

□	Node	Subtype	State	Reachable	Management IPs	External Connections	Options
□	iosv-1	IOSv	● ACTIVE	✓ True	10.255.0.7	telnet://virl.ucllap.dmaes.be:17000 telnet://virl.ucllap.dmaes.be:17001	
□	iosv-2	IOSv	● ACTIVE	✓ True	10.255.0.8	telnet://virl.ucllap.dmaes.be:17002 telnet://virl.ucllap.dmaes.be:17003	
□	~mgmt-lxc	mgmt-lxc	● ACTIVE	✓ True	10.255.0.6	~lxc-flat - 172.16.1.52 ssh://virl@virl.ucllap.dmaes.be:10000	

Showing 1 to 3 of 3 entries

SSH native handler configuration for Linux

Traffic captures

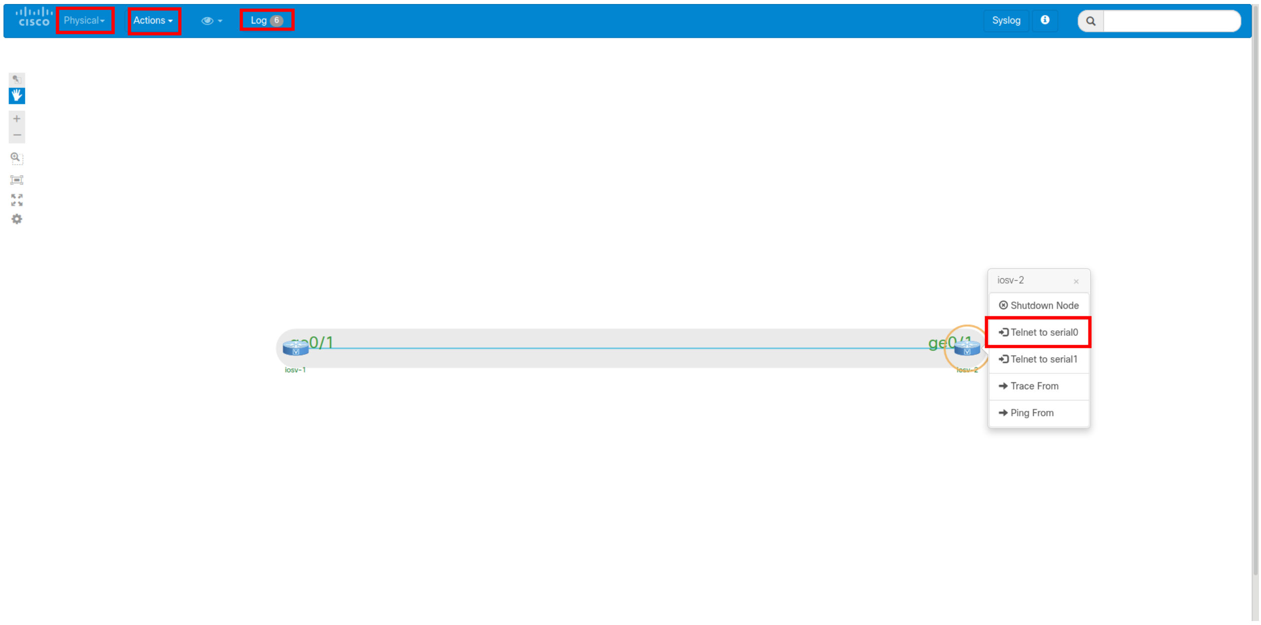
Show 10 entries

Filter:

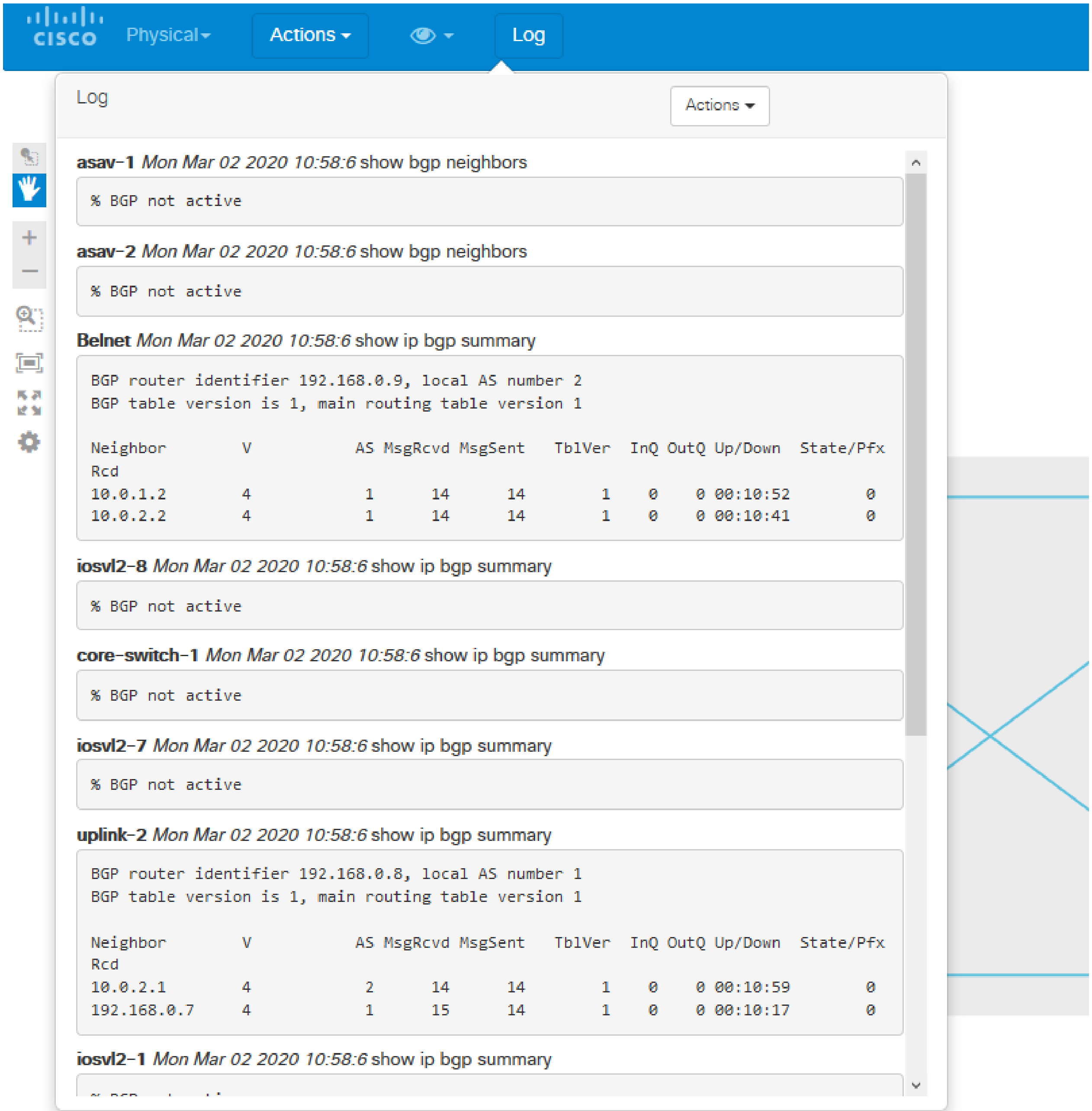
Mode	Group	PCAP filter	Running	Node	Interface	Options
No data available in table						

Showing 0 to 0 of 0 entries

The AutoNetkit visualization screen allows you to see how all the nodes are connected to each other, you can see the different routing protocols, autonomous system numbers . . . To use the visualization screen you must have created the topology and generated the config files using the ANK tool. In this screen you can click any node and connect through telnet to the devices. From there you can configure them. You can also connect through the terminal. When hovering over the nodes a pop-up menu is available from where you can Telnet to the devices, ping from or to the devices . . .



There are a couple of view options for the topology. The default one showed on the screen is the pysical model of the topology. This shows the nodes and conncetions between them. Under "Physical", you can select other options like a BGP, OSPF, Layer one, Layer two or Layer three Views. The Live visualization also allows you to run a couple of "Actions" to for example collect data or parts of the running-config. In "Log" you get the results of actions or pings.



asav-1 *Mon Mar 02 2020 10:58:6* show bgp neighbors

```
% BGP not active
```

asav-2 *Mon Mar 02 2020 10:58:6* show bgp neighbors

```
% BGP not active
```

Belnet *Mon Mar 02 2020 10:58:6* show ip bgp summary

```
BGP router identifier 192.168.0.9, local AS number 2
BGP table version is 1, main routing table version 1

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/Pfx
Rcd
10.0.1.2        4            1      14      14        1    0    0 00:10:52        0
10.0.2.2        4            1      14      14        1    0    0 00:10:41        0
```

iosvl2-8 *Mon Mar 02 2020 10:58:6* show ip bgp summary

```
% BGP not active
```

core-switch-1 *Mon Mar 02 2020 10:58:6* show ip bgp summary

```
% BGP not active
```

iosvl2-7 *Mon Mar 02 2020 10:58:6* show ip bgp summary

```
% BGP not active
```

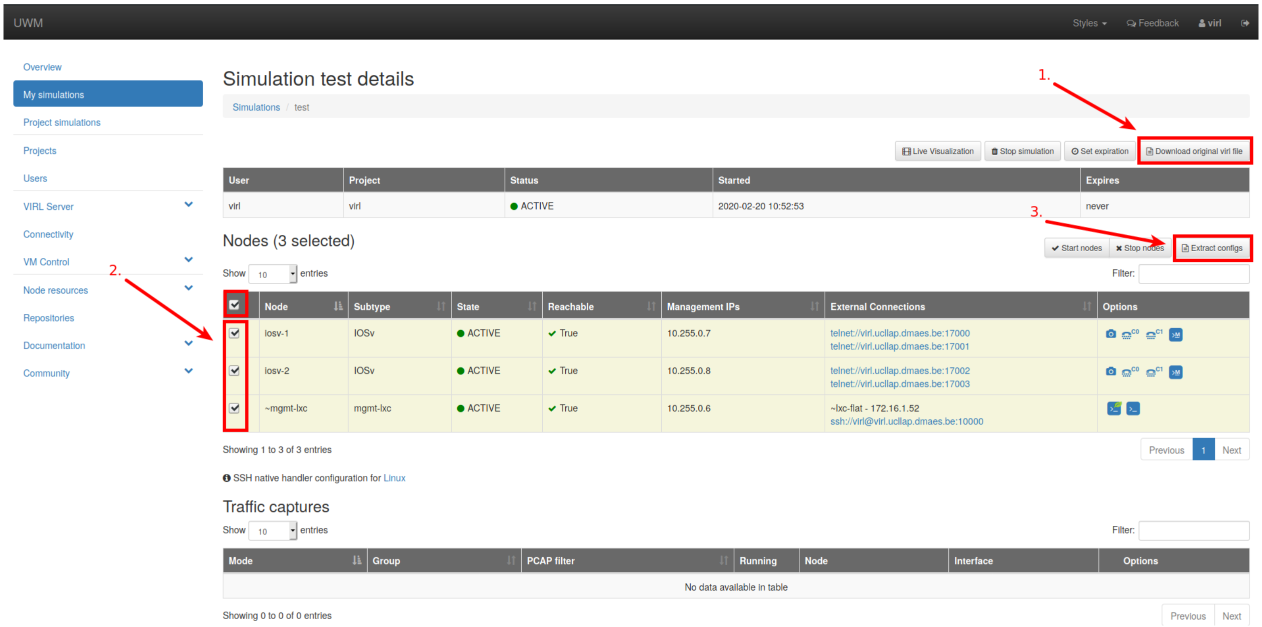
uplink-2 *Mon Mar 02 2020 10:58:6* show ip bgp summary

```
BGP router identifier 192.168.0.8, local AS number 1
BGP table version is 1, main routing table version 1

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/Pfx
Rcd
10.0.2.1        4            2      14      14        1    0    0 00:10:59        0
192.168.0.7     4            1      15      14        1    0    0 00:10:17        0
```

iosvl2-1 *Mon Mar 02 2020 10:58:6* show ip bgp summary

Here you see all the BGP information from every device of the topology under the "log" tab.

3.2.6 Save the original .virl file as well as the configured one



Simulation test details

Simulations / test

User	Project	Status	Started	Expires
virl	virl	● ACTIVE	2020-02-20 10:52:53	never

Nodes (3 selected)

☑	Node	Subtype	State	Reachable	Management IPs	External Connections	Options
☑	iosv-1	IOSv	● ACTIVE	✓ True	10.255.0.7	telnet://virl.ucllap.dmaes.be:17000 telnet://virl.ucllap.dmaes.be:17001	
☑	iosv-2	IOSv	● ACTIVE	✓ True	10.255.0.8	telnet://virl.ucllap.dmaes.be:17002 telnet://virl.ucllap.dmaes.be:17003	
☑	~mgmt-lxc	mgmt-lxc	● ACTIVE	✓ True	10.255.0.6	~lxc-flat - 172.16.1.52 ssh://virl@virl.ucllap.dmaes.be:10000	

Showing 1 to 3 of 3 entries

SSH native handler configuration for Linux

Traffic captures

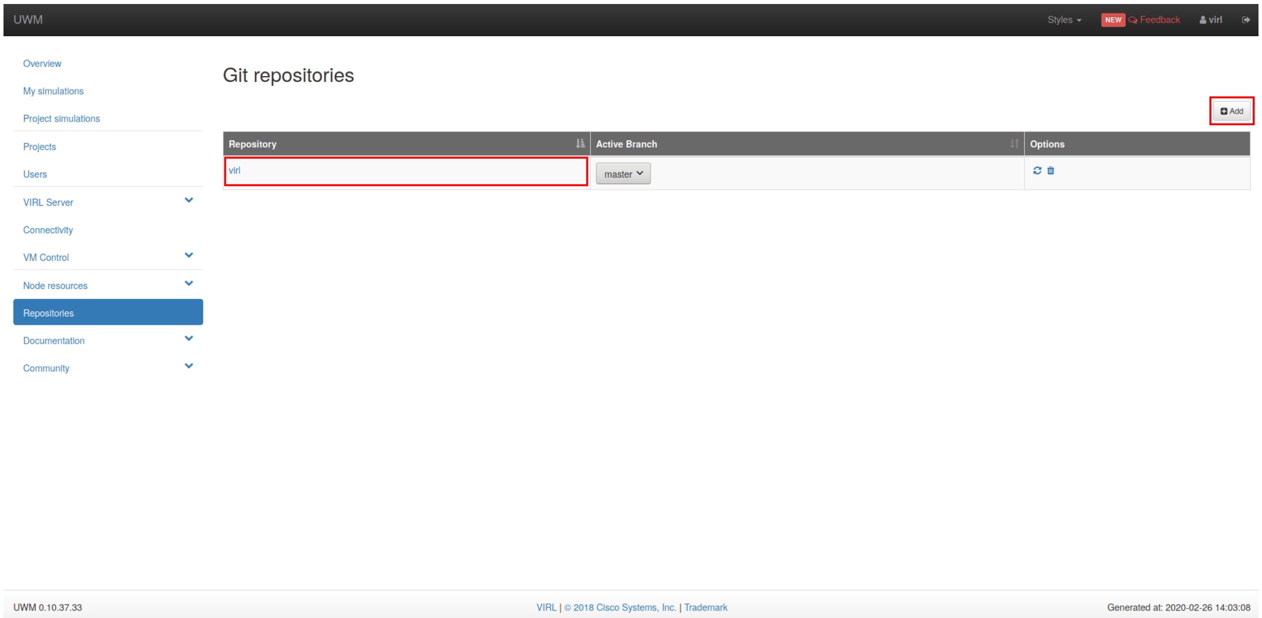
Mode	Group	PCAP filter	Running	Node	Interface	Options
No data available in table						

Showing 0 to 0 of 0 entries

1. If you want to save the original .virl file you can click the button on the top right, this will have only the default configuration if you pressed ANK in the editor.
2. If you also want to save the configuration you have done in the simulation you can select the nodes from witch you want to save it.
3. And press "Extract configs".

3.2.7 Adding a git repository

There is an option for adding git repositories to the VIRL UWM:

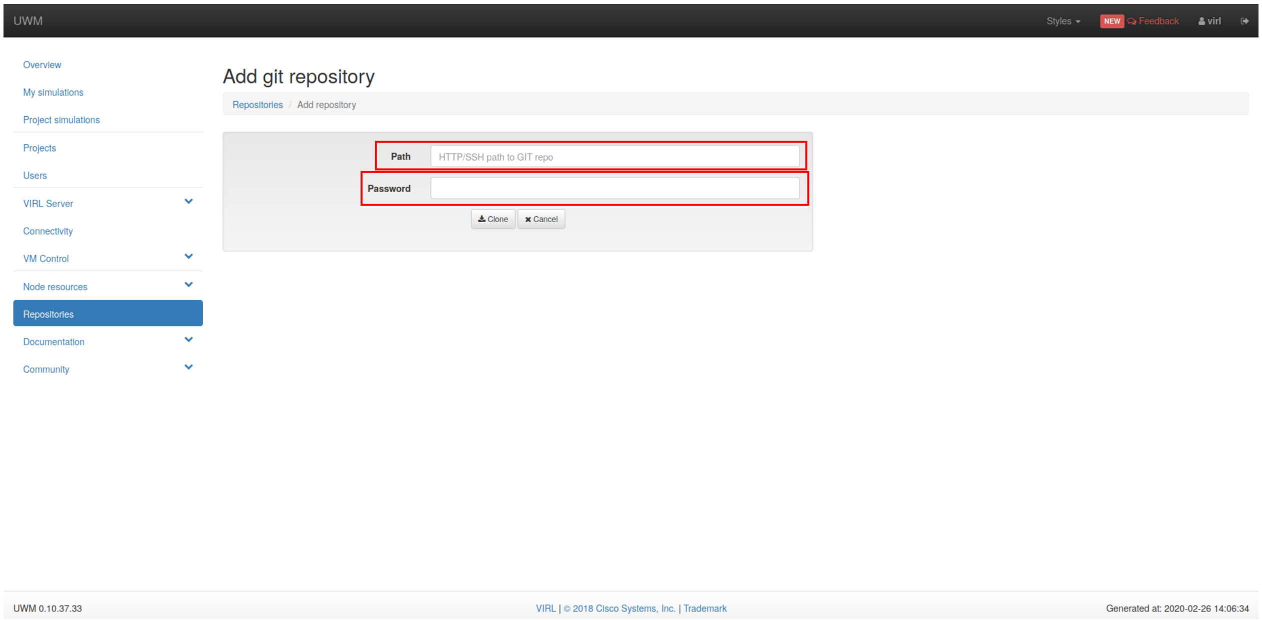


Git repositories

Repository	Active Branch	Options
virl	master	

UWM 0.10.37.33 VIRL | © 2018 Cisco Systems, Inc. | Trademark Generated at: 2020-02-26 14:03:08

You can either add a new repository or enter one you already have connected. We will first take a look at adding one.



Add git repository

Repositories / Add repository

Path: HTTP/SSH path to GIT repo

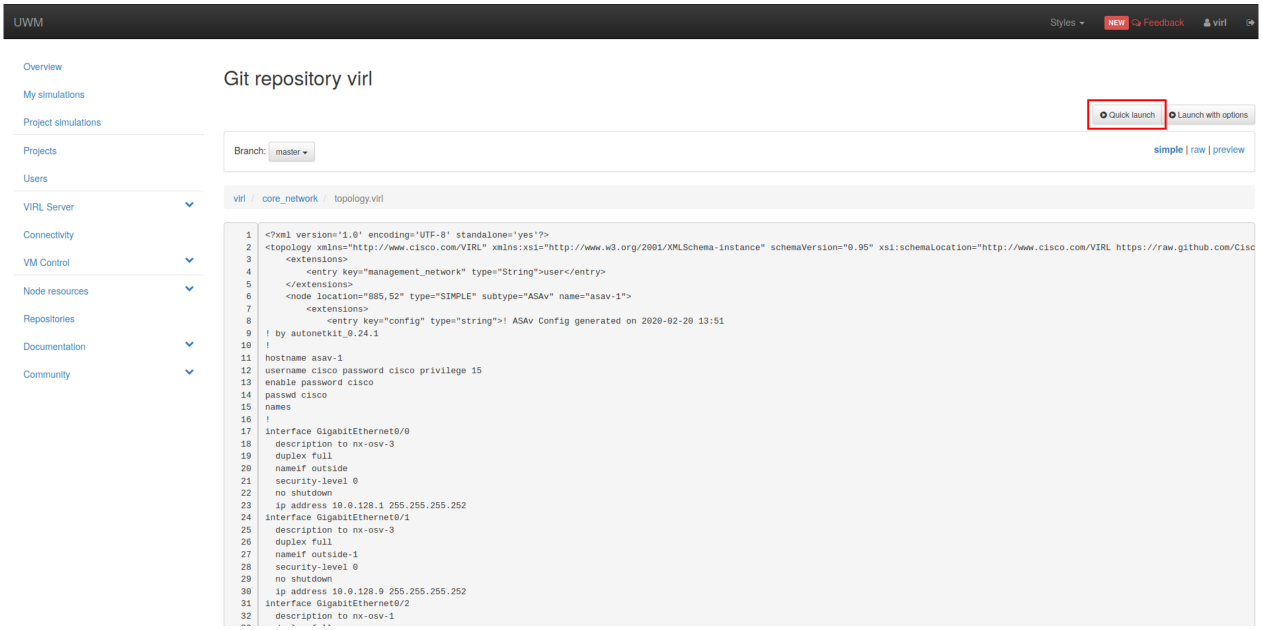
Password:

UWM 0.10.37.33 VIRL | © 2018 Cisco Systems, Inc. | Trademark Generated at: 2020-02-26 14:06:34

Here you will have to specify the clone path and the password of the user. For this to work make a user on your gitlab with the same name as your VIRL user or the other way around make your VIRL user match your gitlab username.

When you have added a repository you can enter it and even launch .virl files straight from the repository.

Sadly we have not found an easy way to push to the repository from UWM. You can manually extract the .virl files and push them yourself as seen in the previous section.

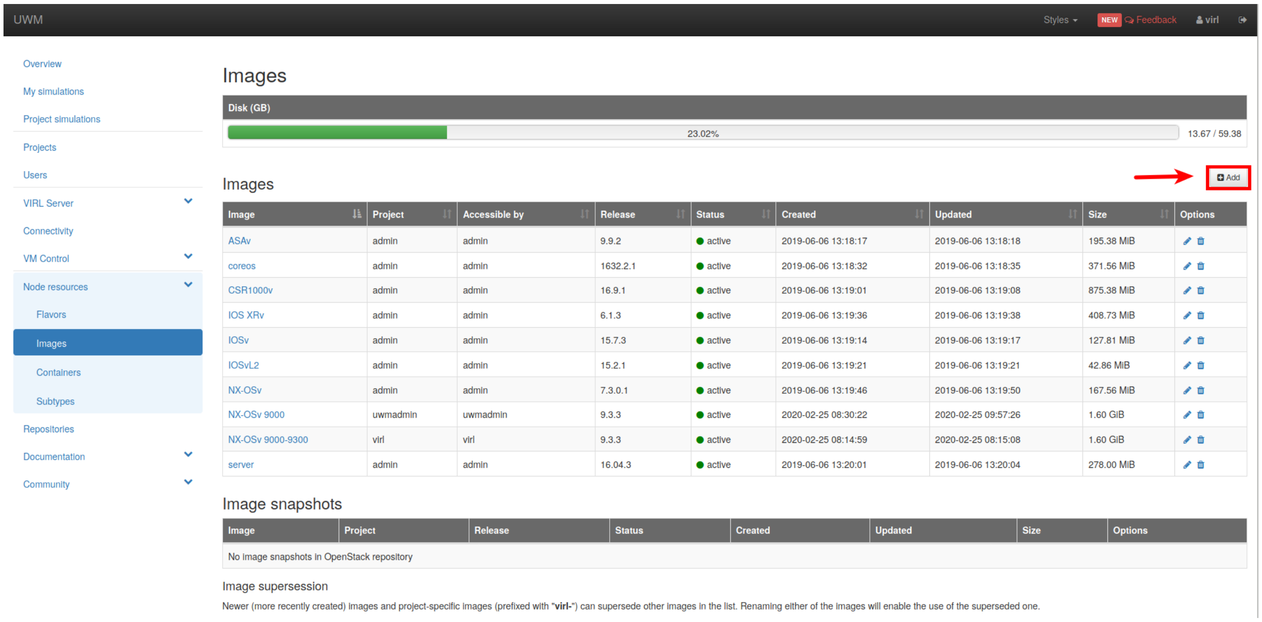


```
<?xml version='1.0' encoding='UTF-8' standalone='yes'?>
<topology xmlns="http://www.cisco.com/VIRL" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" schemaVersion="0.95" xsi:schemaLocation="http://www.cisco.com/VIRL https://raw.github.com/Cisc
    <extensions>
        <entry key="management_network" type="String">user</entry>
    </extensions>
    <node location="885,52" type="SIMPLE" subtype="ASAv" name="asav-1">
        <extensions>
            <entry key="config" type="string">! ASAv Config generated on 2020-02-20 13:51
! by autonetkit_0.24.1
!
hostname asav-1
username cisco password cisco privilege 15
enable password cisco
passwd cisco
names
!
interface GigabitEthernet0/0
  description to nx-osv-3
  duplex full
  nameif outside
  security-level 0
  no shutdown
  ip address 10.0.128.1 255.255.255.252
interface GigabitEthernet0/1
  description to nx-osv-3
  duplex full
  nameif outside-1
  security-level 0
  no shutdown
  ip address 10.0.128.9 255.255.255.252
interface GigabitEthernet0/2
  description to nx-osv-1
```

3.2.8 Adding images

Earlier we already spoke of having troubles with certain nodes not supporting certain techniques. As a solution you can look for a different sandbox environment in the Devnet solution, or if you run your own VIRL server you can also add images or update an existing device with a newer image in the UWM.

Go to the main page and click *Node resources rightarrow Images rightarrow Add*.



Images

Disk (GB)

23.02% 13.67 / 59.38

Images

Image	Project	Accessible by	Release	Status	Created	Updated	Size	Options
ASAv	admin	admin	9.9.2	● active	2019-06-06 13:18:17	2019-06-06 13:18:18	195.38 MiB	✎ 🗑
coreos	admin	admin	1632.2.1	● active	2019-06-06 13:18:32	2019-06-06 13:18:35	371.56 MiB	✎ 🗑
CSR1000v	admin	admin	16.9.1	● active	2019-06-06 13:19:01	2019-06-06 13:19:08	875.38 MiB	✎ 🗑
IOS XRv	admin	admin	6.1.3	● active	2019-06-06 13:19:36	2019-06-06 13:19:38	408.73 MiB	✎ 🗑
IOSv	admin	admin	15.7.3	● active	2019-06-06 13:19:14	2019-06-06 13:19:17	127.81 MiB	✎ 🗑
IOSvL2	admin	admin	15.2.1	● active	2019-06-06 13:19:21	2019-06-06 13:19:21	42.86 MiB	✎ 🗑
NX-OSv	admin	admin	7.3.0.1	● active	2019-06-06 13:19:46	2019-06-06 13:19:50	167.56 MiB	✎ 🗑
NX-OSv 9000	uwmadmin	uwmadmin	9.3.3	● active	2020-02-25 08:30:22	2020-02-25 09:57:26	1.60 GiB	✎ 🗑
NX-OSv 9000-9300	virl	virl	9.3.3	● active	2020-02-25 08:14:59	2020-02-25 08:15:08	1.60 GiB	✎ 🗑
server	admin	admin	16.04.3	● active	2019-06-06 13:20:01	2019-06-06 13:20:04	278.00 MiB	✎ 🗑

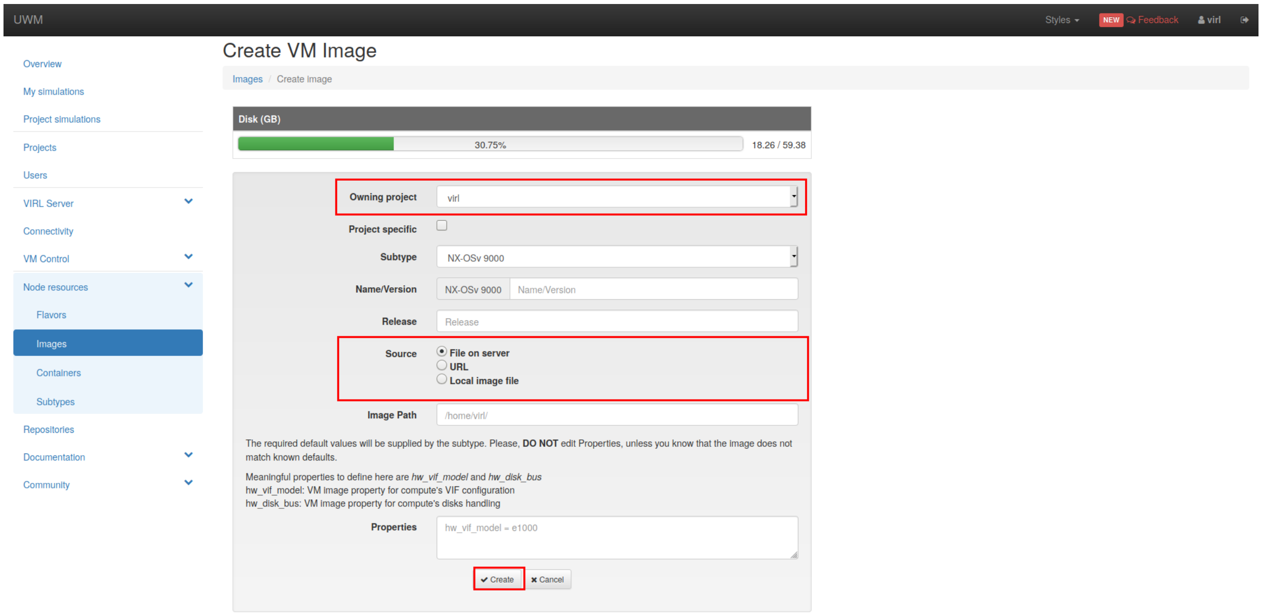
Image snapshots

Image	Project	Release	Status	Created	Updated	Size	Options
No image snapshots in OpenStack repository							

Image supersession

Newer (more recently created) images and project-specific images (prefixed with "**virl-**") can supersede other images in the list. Renaming either of the images will enable the use of the superseded one.

We will use the example of adding the NX-OSv9K since we needed these for our simulation.



Create VM Image

Disk (GB) 30.75% 18.26 / 59.38

Owning project: virl

Project specific: □

Subtype: NX-OSv 9000

Name/Version: NX-OSv 9000 Name/Version

Release: Release

Source: ● File on server ○ URL ○ Local image file

Image Path: /home/virl/

The required default values will be supplied by the subtype. Please, **DO NOT** edit Properties, unless you know that the image does not match known defaults.

Meaningful properties to define here are *hw_vif_model* and *hw_disk_bus*
hw_vif_model: VM image property for compute's VIF configuration
hw_disk_bus: VM image property for compute's disks handling

Properties: hw_vif_model = e1000

✔ Create ✖ Cancel

- On top you can see how many space you still have on your disk.
- You can choose for what project and users you want to enable it.
- You have to select the correct Subtype and Source.
- And lastly you have to provide the image and create the VM image.

Now, to be able to use our new image, we have to make it available in the VIRL editor. Ideally, this would happen automatically, but since the VIRL editor is clearly still under heavy development, we have to do it manually. More specifically, you need to add the following code snippet to

`/usr/local/lib/python2.7/dist-packages/autonetkit_cisco_webui/web_content/sampleData/subtypes.json`

```
{
    "intfManagement": "mgmt0",
    "icon": "nx_osv",
    "intfNameFormat": "Ethernet1/{0}",
    "minIntfIndices": [
        1
    ],
    "name": "NX-OSv 9000",
    "intfSegmentSizes": [
        0
    ],
    "maxIntfIndices": [
        27
    ],
    "visible": true,
    "serialConsoles": 2
}
```

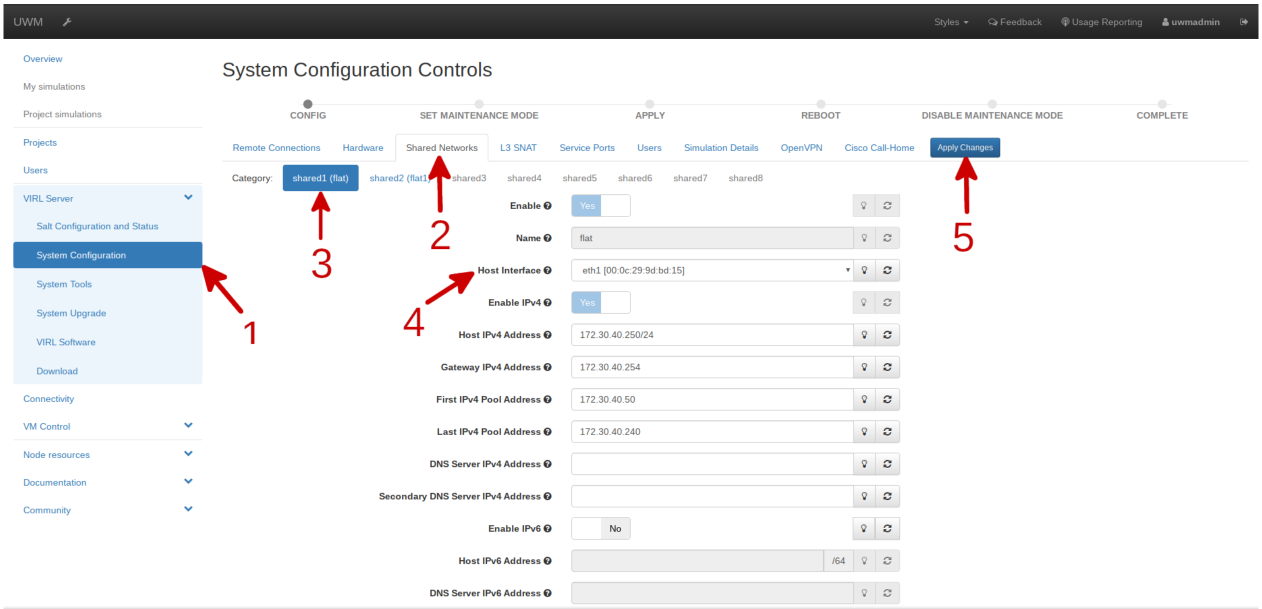
Listing 1: JSON code for NX-OSv 9000

3.2.9 Interconnecting VIRL simulations and connecting to external networks

In VIRL, it is possible to interconnect VIRL simulations and to connect VIRL simulations to other external networks. Interconnecting simulations is possible on both the DevNet sandbox platform and the VIRL PE, connecting simulations to external networks is, for security reasons, only possible on the VIRL PE.

Interconnecting simulations is as easy as connecting adding a **FLAT** node (found under the *switches* group in the VIRL editor) to each topology and ensuring they are connected both to the same link.

The **FLAT** node essentially connects the interface on your router with a specifically defined bridge on your VIRL server. So, connecting to an external network would seem as simple as adding a physical interface to the bridge. The easiest way to do this correctly, is in *System Configuration* under the **VIRL Server** (1) tab in UWM (login as **uwmadmin**). Once there, navigate to *Shared Networks* (2) → *shared1 (flat)* (3) and set the physical interface under *Host Interface* (4). You can change other settings too here, like your host ip-address, the ip-range for your virtual routers, . . . When done, click *Apply Changes* (5) and follow the on-screen instructions.

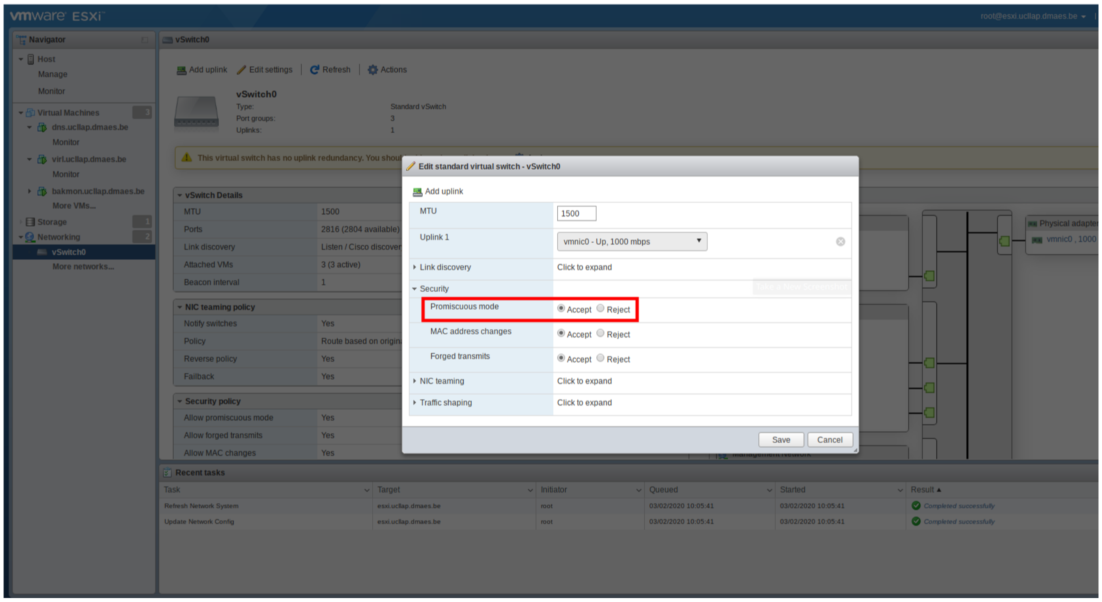


System Configuration Controls

Enable	Yes
Name	flat
Host Interface	eth1 [00:0c:29:9d:bd:15]
Enable IPv4	Yes
Host IPv4 Address	172.30.40.250/24
Gateway IPv4 Address	172.30.40.254
First IPv4 Pool Address	172.30.40.50
Last IPv4 Pool Address	172.30.40.240
DNS Server IPv4 Address	
Secondary DNS Server IPv4 Address	
Enable IPv6	No
Host IPv6 Address	/64
DNS Server IPv6 Address	

Figure 1: Configuring the *flat* network

When running VIRL PE in an ESXi vm, you may need to enable *promiscuous mode* for the used vSwitch.



vSwitch0
Type: Standard vSwitch
Port groups: 3
Uplinks: 1

This virtual switch has no uplink redundancy. You shoul

vSwitch Details

MTU	1500
Ports	2816 (2804 available)
Link discovery	Listen / Cisco discover
Attached VMs	3 (3 active)
Beacon interval	1

NIC teaming policy

Notify switches	Yes
Policy	Route based on origin
Reverse policy	Yes
Failback	Yes

Security policy

Allow promiscuous mode	Yes
Allow forged transmits	Yes
Allow MAC changes	Yes

Edit standard virtual switch - vSwitch0

Add uplink

MTU	1500
Uplink 1	vmnic0 - Up, 1000 mbps
Link discovery	Click to expand
Security	
Promiscuous mode	● Accept ○ Reject
MAC address changes	● Accept ○ Reject
Forged transmits	● Accept ○ Reject
NIC teaming	Click to expand
Traffic shaping	Click to expand

Save Cancel

Recent tasks

Task	Target	Initiator	Queued	Started	Result ▲
Refresh Network System	esxi.ucllap.dmaes.be	root	03/02/2020 10:05:41	03/02/2020 10:05:41	Completed successfully
Update Network Config	esxi.ucllap.dmaes.be	root	03/02/2020 10:05:41	03/02/2020 10:05:41	Completed successfully

Figure 2: Enabling promiscuous mode in ESXi

4 Our test setup

Using VIRL on your own hardware can be done quiet easily. Downloading the ISO meant for bare-metal, installing it and configuring users and interfaces, is enough to get started.

For practical reasons we made our setup a bit more complicated. Firstly, this made work easier. Secondly, we learned new stuff while building the setup, and it could be used for another task given to us.

In detail, our setup exists of a gateway (we used VyOS) to manage networks (the OpenVPN-network, used for external access, the private network, to which we connect when on-site, the services network, which contains the VIRL server and DNS/backup server, and the management network, used to manage gateway, switch and hypervisor), a Cisco catalyst switch to connect devices to the those networks, a ESXi-box to host our VMs, a DNS-VM, a VM for backup & monitoring, and the VIRL-VM.

4.1 Installation Guide

I'll explain installing VIRL, LibreNMS and RANCID. The other parts in our setup our just there for logistical reasons, and will be different in any other setup.

4.1.1 VIRL

Installing VIRL is pretty simple and straightforward. There are images provided for installation either on bare-metal or in ESXi. We installed VIRL in ESXi, this allowed us to take snapshots of the machine on the one hand, and use the same hardware for other services on the other hand.

Installation of VIRL in ESXi is done by downloading the provided .ova-file, importing it in ESXi, and following the VIRL configuration prompt. There isn't much more to it than setting users and passwords, and assigning an interface for access to VIRL.

When all done, you can login to the VIRL-Web interface and import your key as explained by the big yellow banner greeting you.

4.1.2 LibreNMS

Another question we got, aside from testing VIRL, was to setup router/switch config-backups with RANCID in a change-based manner instead of a time-based manner.

We did this by enabling SNMP-traps for configuration changes on our switch. Then, we needed something that could catch those SNMP-traps and trigger a rancid-run. Because LibreNMS is pretty easy to set up and comes with good support for SNMP-traps out-of-the-box, we chose this solution.

We suggest you follow the official documentation for setting up LibreNMS, it is very straightforward: `https://docs.librenms.org/Installation/`

Sadly, LibreNMS does not have handlers for the right Cisco SNMP-traps and only logs them. However, writing your own handler is pretty easy and explained here: `https://docs.librenms.org/Developing/SNMP-Traps/`

We wrote the following handler:

```
<?php
/**
    * CiscoConfigChanged.php
    *
    * Handles CISCO-CONFIG-MAN-MIB::ciscoConfigManEvent
*/

namespace LibreNMS\Snmptrap\Handlers;

use App\Models\Device;
use LibreNMS\Interfaces\SnmptrapHandler;
use LibreNMS\Snmptrap\Trap;
use Log;

class CiscoConfigChanged implements SnmptrapHandler {
    /**
        * Handle snmptrap.
        * Data is pre-parsed and delivered as a Trap.
        *
        * @param Device $device
        * @param Trap $trap
        * @return void
        */
    public function handle(Device $device, Trap $trap) {
        shell_exec(’{{ librenms_ciscoConfigChanged }}’);
    }
}
```

Listing 2: LibreNMS SnmptrapHandler for CISCO-CONFIG-MAN-MIB::ciscoConfigManEvent

Naturally, any other monitoring platform that can trigger arbitrary code on SNMP-traps can be used instead of LibreNMS. The principle stays the same.

4.1.3 RANCID

RANCID is a platform for easy backups of networking devices. Our setup is as default as it gets, and I suggest you follow this fine guide: `http://www.linuxhomenetworking.com/wiki/index.php/Quick_HOWTO_:_Ch1_:_Network_Backups_With_Rancid` We did, however, use Git instead of the default CVS for version-management. Using git is as simple as setting RCSSYS=`git;` in `rancid.conf`, before you run `rancid-cvs`.

5 The UCLL core network

5.1 Current specific part of the core network that we virtualized

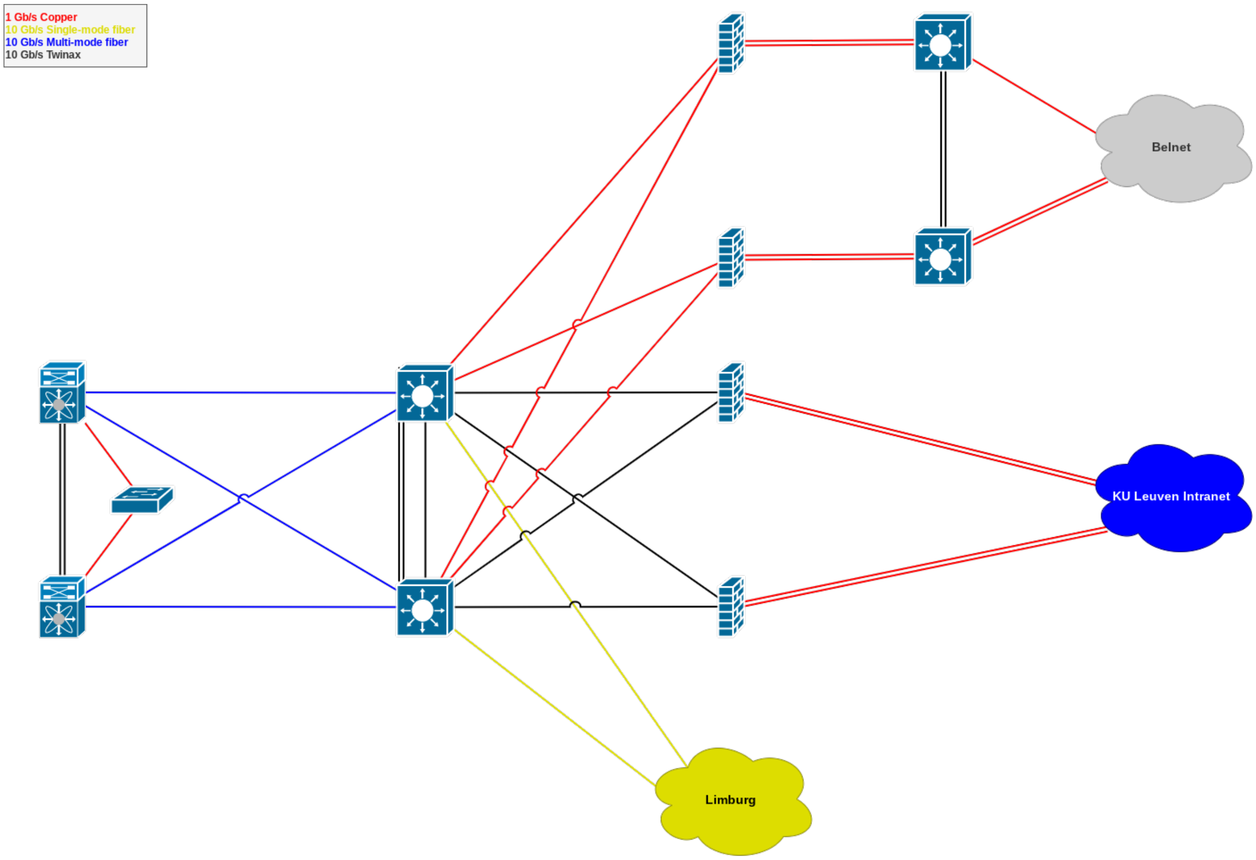


Figure 3: Layer one topology

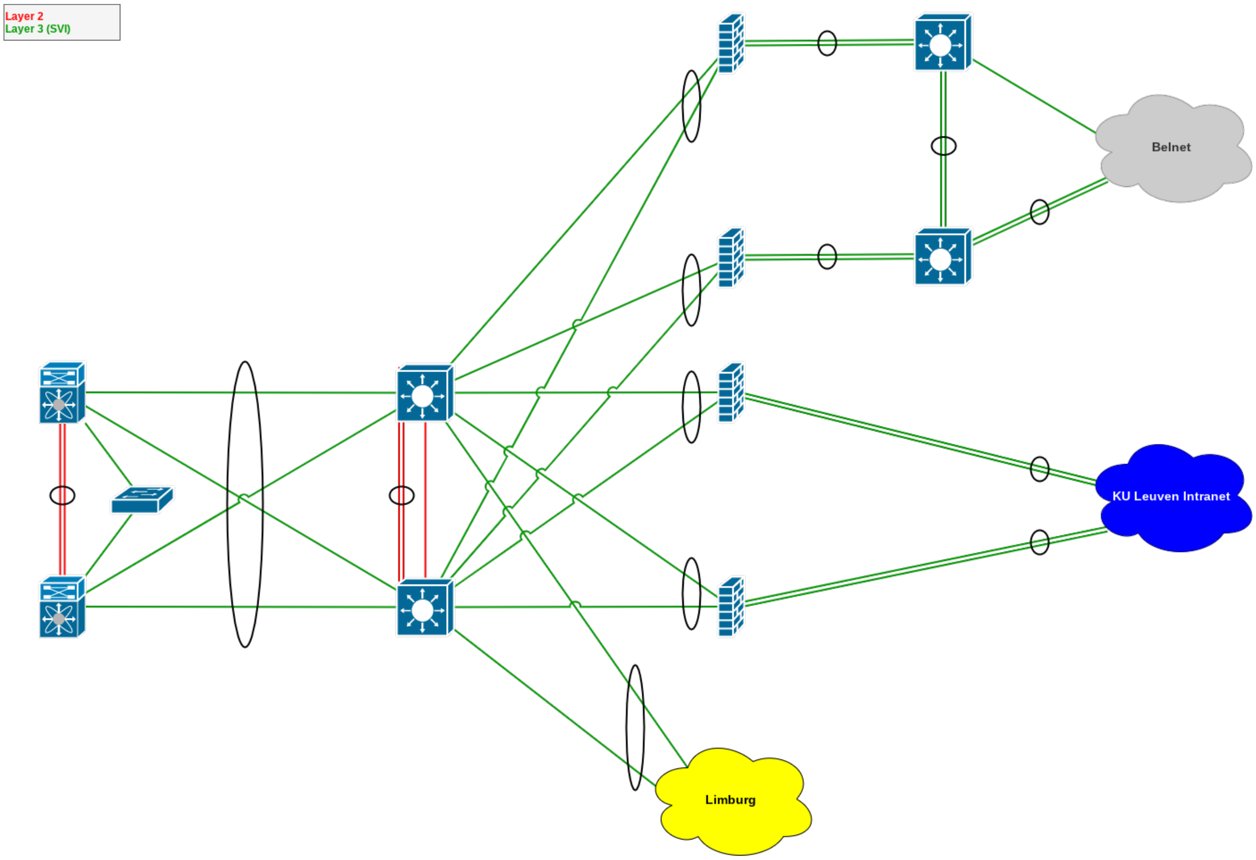
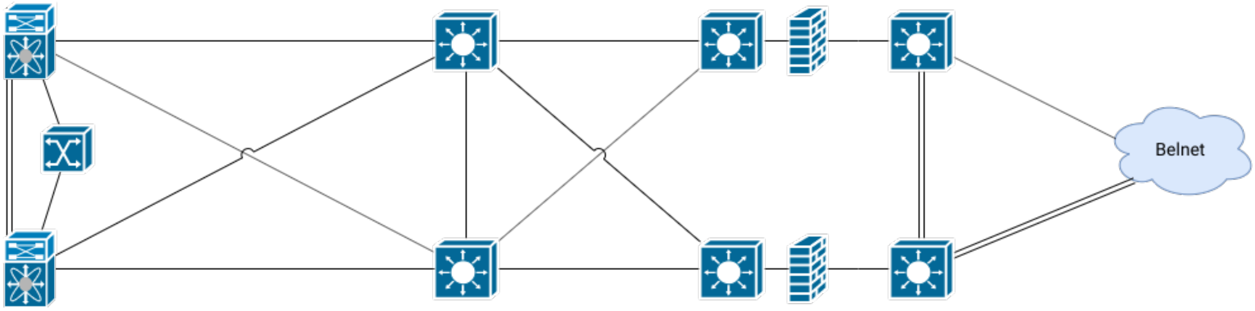


Figure 4: Layer two topology

5.2 Our implementation

5.2.1 The topology



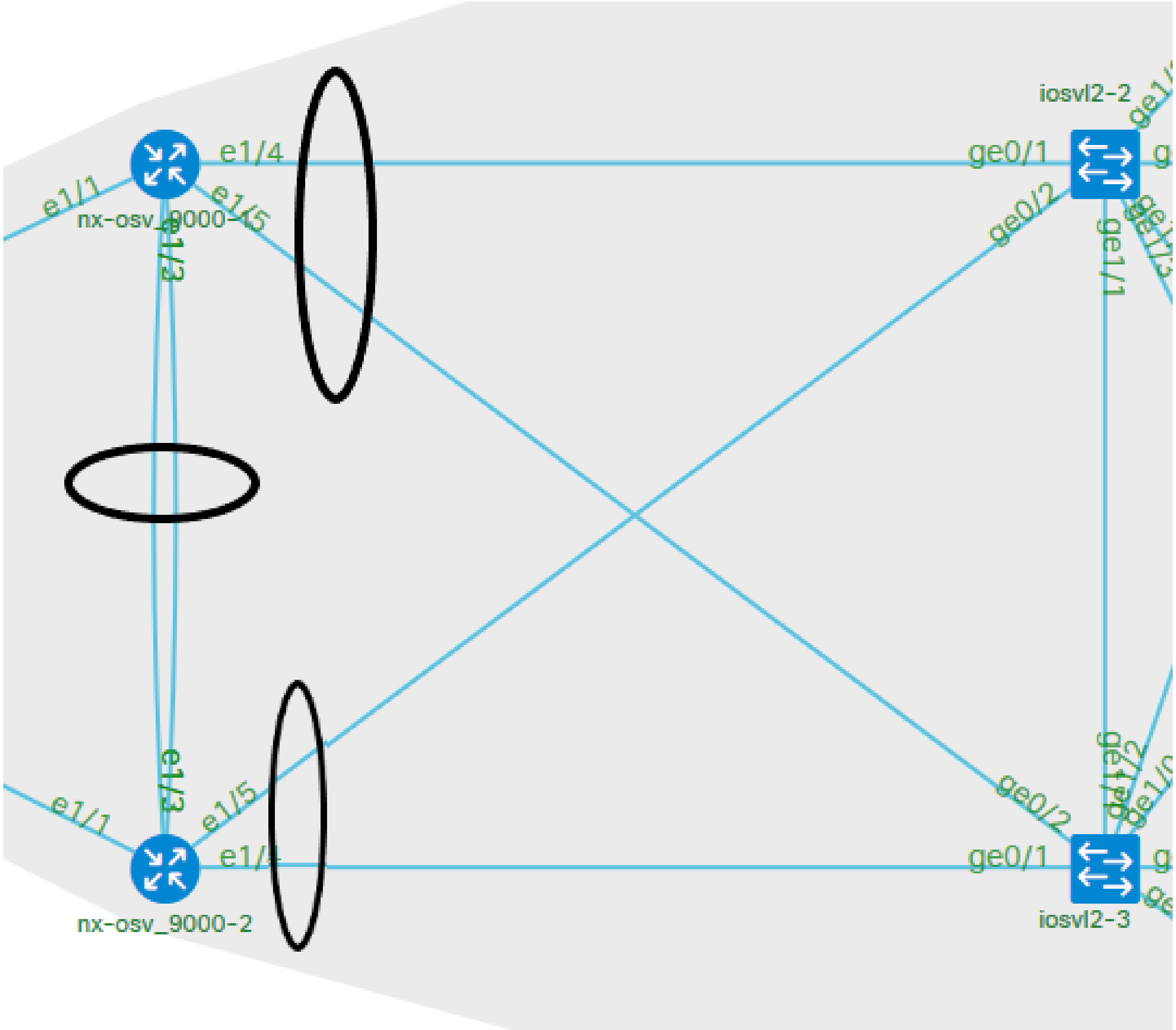
For our project we altered the network based on the project requirements, the most important parts to focus on and limitations in VIRL.

That being said we will shortly discuss the alterations we made:

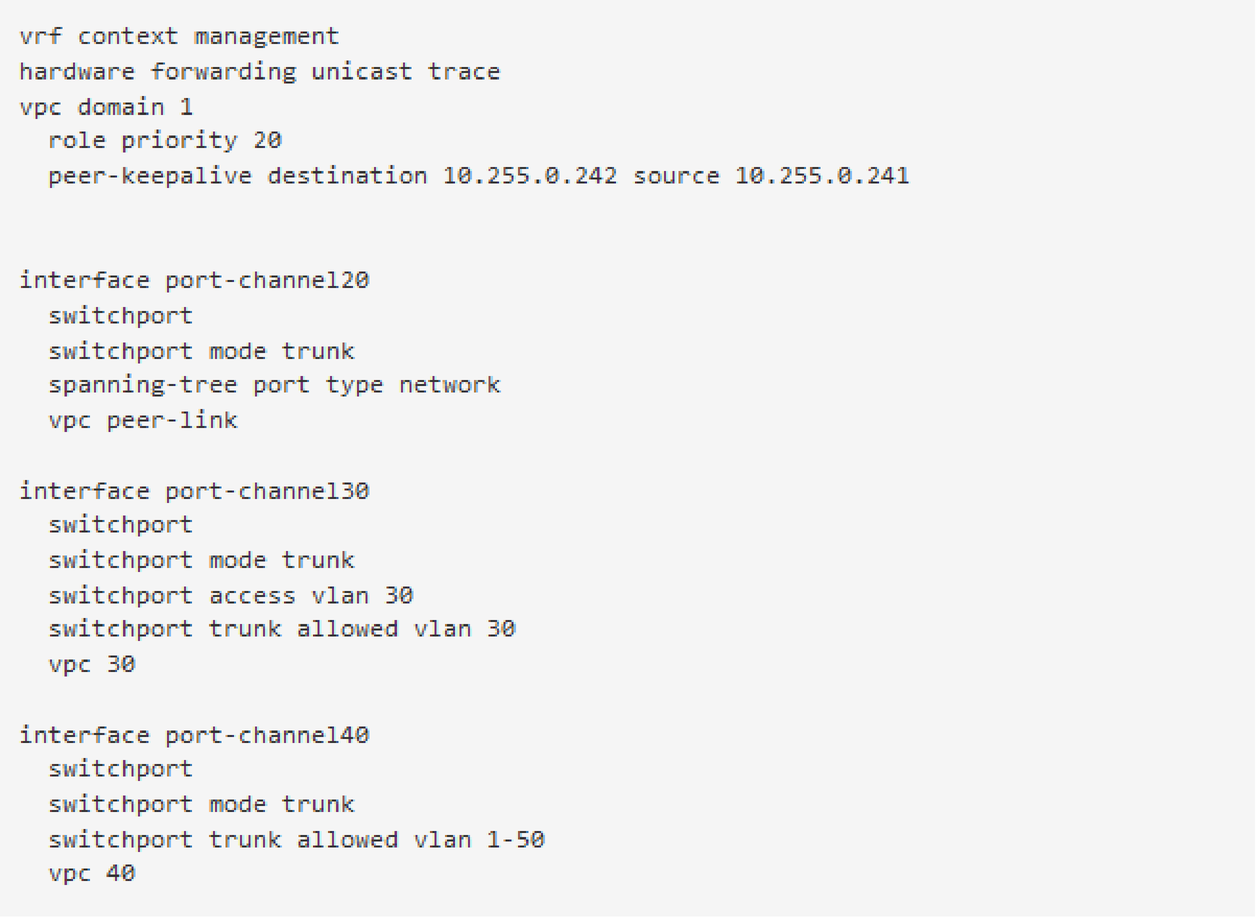
- Mainly of course we didn't try to remake the whole core network all at once. We firstly focused on the up-link to Belnet, the Firewall, the core switches and the data center.
- With our limited part of the network we still had to figure out this implementation in VIRL. The machines didn't always do what we wanted them to do. For example the ASA machines weren't able to make Etherchannels so we had to put some extra switches between them and the core switches.
- Next we ran into some trouble withe the core switches being unable to do VSS. So we tried HSRP instead.

5.2.2 vPC

On the 2 nexus switches in the topology, we configured vPC or Virtual Port Channel. Port-channel is a technology that allows to aggregate multiple interfaces. The traffic is then distributed between the connections. vPC however provides a way to configure a port-channel across multiple switches, also known as vPC peers. In our topology the 2 nexus switches appear as a single PortChannel to the layer3 switches.



Part of the topology with the two nexus switches.

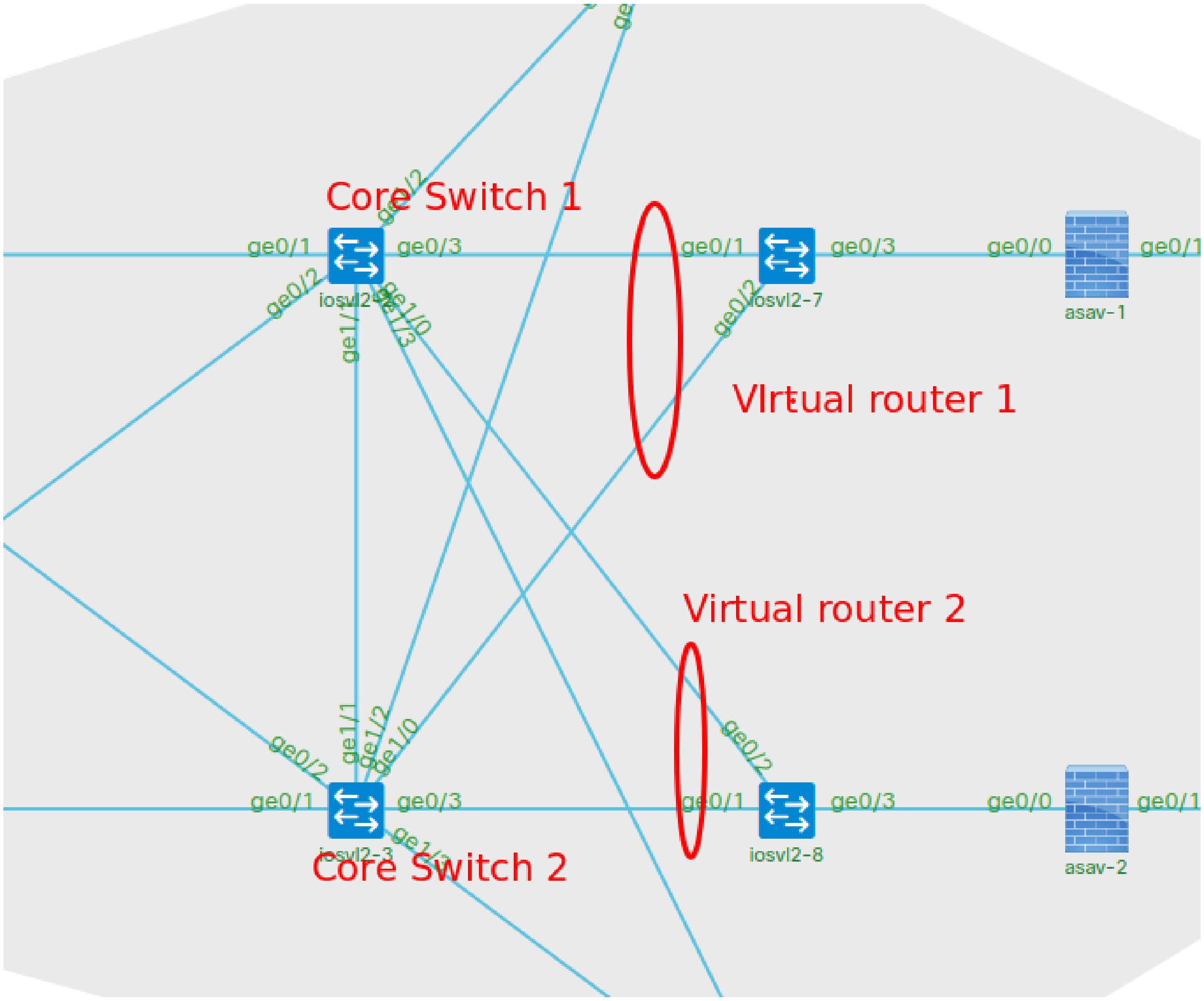


```
vrf context management
hardware forwarding unicast trace
vpc domain 1
  role priority 20
  peer-keepalive destination 10.255.0.242 source 10.255.0.241


interface port-channel20
  switchport
  switchport mode trunk
  spanning-tree port type network
  vpc peer-link

interface port-channel30
  switchport
  switchport mode trunk
  switchport access vlan 30
  switchport trunk allowed vlan 30
  vpc 30

interface port-channel40
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 1-50
  vpc 40
```

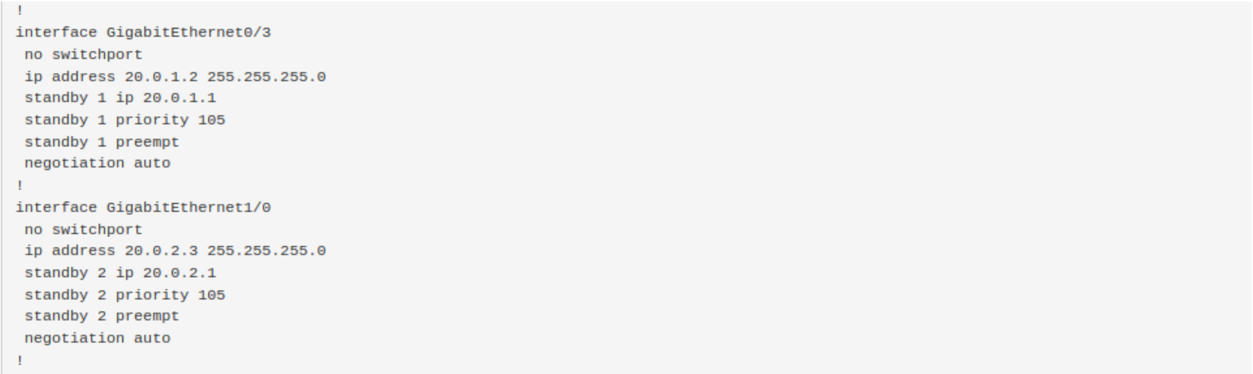
Configuration file of one of the nexus 9k switches.

5.2.3 HSRP

Instead of VSS we set up HSRP on the core switches. We set up a virtual router toward asav-1 and another virtual router towards asav-2. Core switch 1 is for both virtual routers the active.



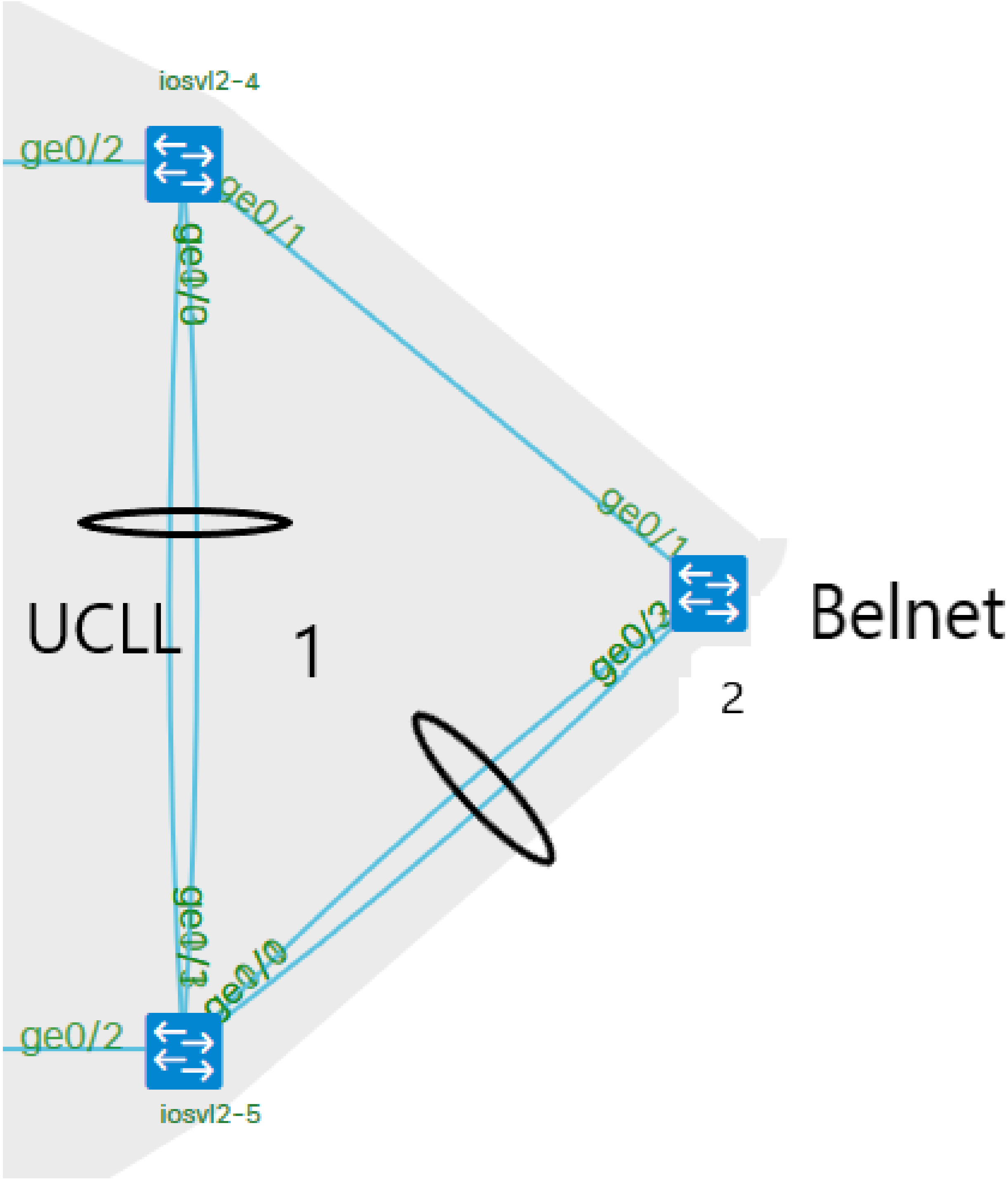
In the running-configuration you indeed see that the core switch 1 has the elevated priority and it will preempt when the priorities would switch.

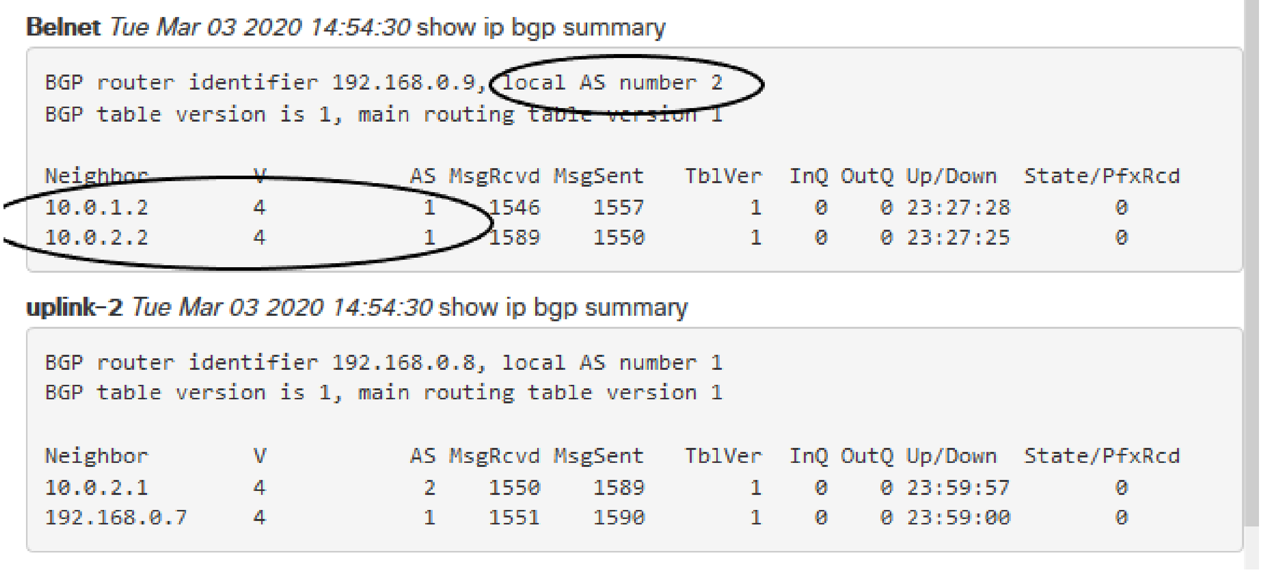


```
!
interface GigabitEthernet0/3
 no switchport
 ip address 20.0.1.2 255.255.255.0
 standby 1 ip 20.0.1.1
 standby 1 priority 105
 standby 1 preempt
 negotiation auto
!
interface GigabitEthernet1/0
 no switchport
 ip address 20.0.2.3 255.255.255.0
 standby 2 ip 20.0.2.1
 standby 2 priority 105
 standby 2 preempt
 negotiation auto
!
```

5.2.4 BGP

BGP or Border Gateway Protocol is a protocol to exchange routing information between autonomous systems (AS) on the internet. In our topology we have two autonomous systems, Belnet and the UCLL. We gave Belnet the AS number two and the UCLL number one. Between the 2 edge switches on the UCLL network we run iBGP. We also configured Etherchannel between the 2 edge switches and between Belnet and one of the switches.





Belnet *Tue Mar 03 2020 14:54:30* show ip bgp summary

```
BGP router identifier 192.168.0.9, local AS number 2
BGP table version is 1, main routing table version 1

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.0.1.2        4            1    1546    1557        1    0    0 23:27:28        0
10.0.2.2        4            1    1589    1550        1    0    0 23:27:25        0
```

uplink-2 *Tue Mar 03 2020 14:54:30* show ip bgp summary

```
BGP router identifier 192.168.0.8, local AS number 1
BGP table version is 1, main routing table version 1

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.0.2.1        4            2    1550    1589        1    0    0 23:59:57        0
192.168.0.7     4            1    1551    1590        1    0    0 23:59:00        0
```

In these configuration file from the Belnet switch and one of the edge switches from the UCLL you can see their AS number, the neighbors, the AS number from the neighbors . . .

5.2.5 The ASA cluster

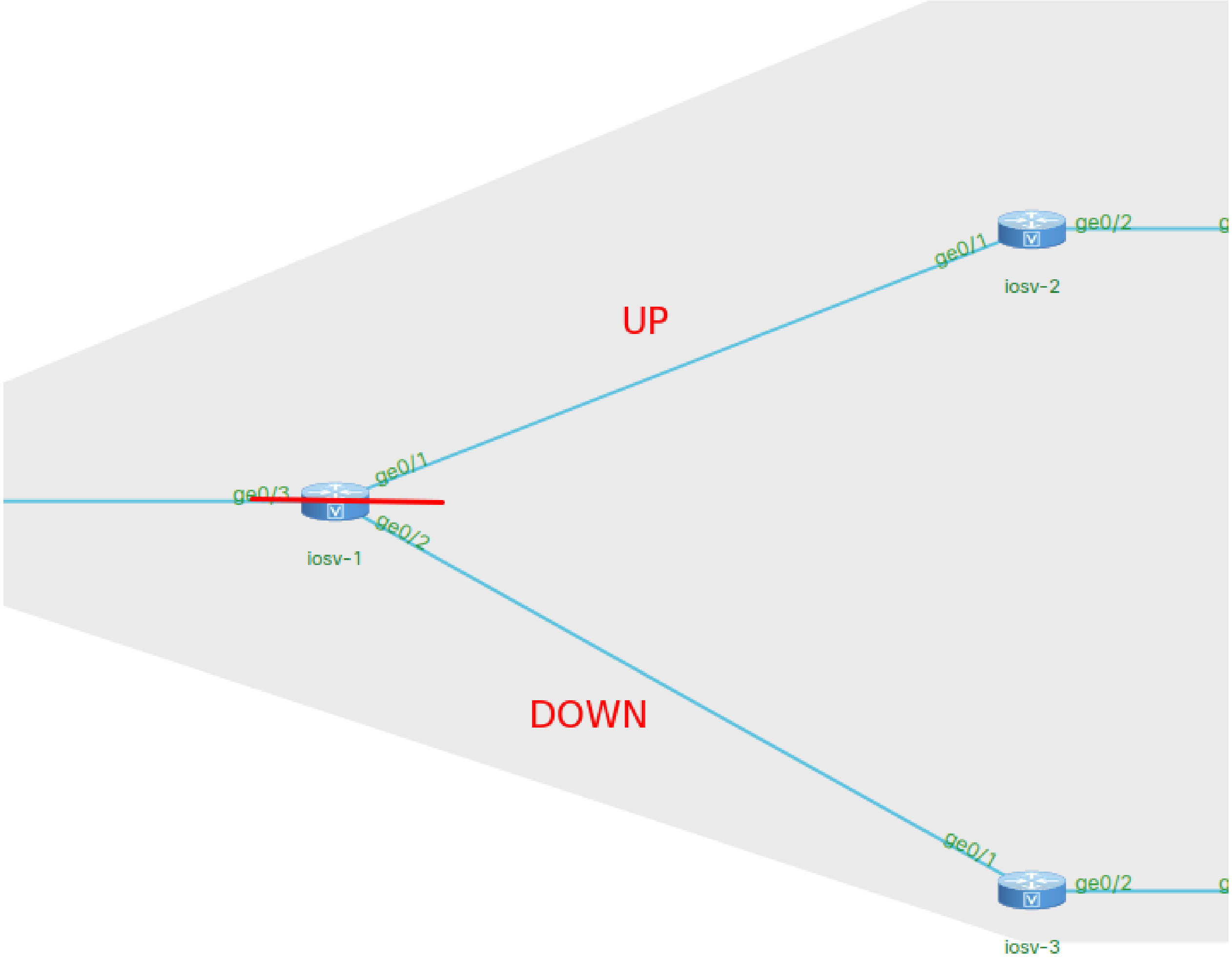
We encountered some trouble trying to configure the ASA machines on VIRL to form an ASA cluster. Firstly there was a lot of confusing information on the internet. Next we encountered trouble with the ASA machines not being able to form Etherchannels. So we added extra switches between the HSRP switches and the cluster.

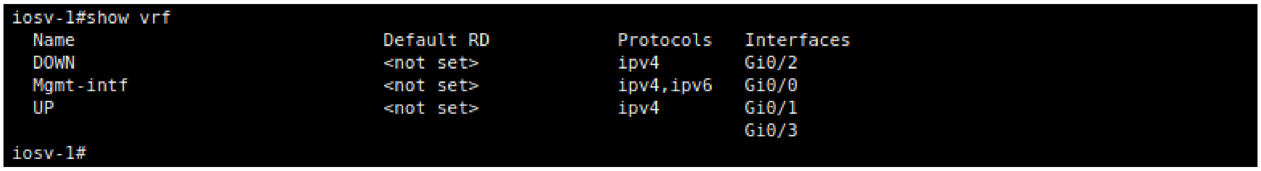
After consultation it appeared better to try and configure the ASA cluster with the ASDM GUI.So pretty late in the project we researched a little topology to connect to the GUI but we lacked the resources to run it on our machine next to our main topology and it was not possible to do on the Devnet solution. So sadly we had to leave this configuration unconquered. It should be possible to configure them except for the fact that they can’t configure Etherchannels.

5.2.6 Vrf lite

Since the core network runs multi area OSPF per Vrf we also tried to implement some vrf's. With a vrf you are splitting up a router into multiple virtual routers. But since vrf works on layer 3 and vlans on layer 2 it gave us some trouble at the end of our project. We decided to not focus our time in solving this problem since we had already succeeded in setting up Vrf all by it self.

In the next images you will see the vrf set up in another topology. In this topology we made 2 Vrf's on the iosv-1 router. One towards iosv-2 called UP and one towards iosv-3 called DOWN. The Mgmt-intf vrf is set by the ANK.

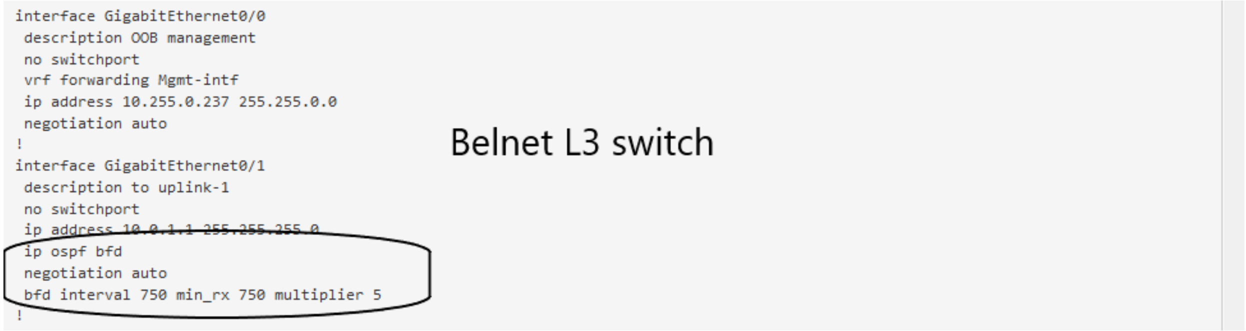




```
iosv-1#show vrf
  Name                             Default RD          Protocols   Interfaces
  DOWN                             <not set>           ipv4        Gi0/2
  Mgmt-intf                        <not set>           ipv4,ipv6   Gi0/0
  UP                               <not set>           ipv4        Gi0/1
                                                                   Gi0/3
iosv-1#
```

5.2.7 BFD

BFD or Bidirectional forwarding detection is a detection protocol to provide a fast way to detect a path failure. BFD can also be used for network administrators because it detects failures at a uniform rate. We used BFD on the Layer three switch from Belnet and on one of the L3 edge switches from the UCLL where we also configured BGP and OSPF.



```
interface GigabitEthernet0/0
 description OOB management
 no switchport
 vrf forwarding Mgmt-intf
 ip address 10.255.0.237 255.255.0.0
 negotiation auto
!
interface GigabitEthernet0/1
 description to uplink-1
 no switchport
 ip address 10.0.1.1 255.255.255.0
 ip ospf bfd
 negotiation auto
 bfd interval 750 min_rx 750 multiplier 5
!
```

6 Links

Cisco configuration pages `https://www.cisco.com/`

Cisco DevNet `https://devnetsandbox.cisco.com`

Cisco LearningNetwork `https://learningnetwork.cisco.com`

VIRL setup documentation `http://get.virl.info/index.php`

ESXi `https://my.vmware.com/en/web/vmware/evalcenter?p=free-esxi6`

VyOS `https://www.vyos.io/`

RANCID `https://www.shrubbery.net/rancid/`

LibreNMS `https://www.librenms.org/`