

**SOFTWARE DEFINED WIDE AREA NETWORKS**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ZAHRA KHALID

ZEYNEP TEMIZEL

CONVERGED OPTIMIZED NETWORKS 15 HP

HALMSTAD UNIVERSITY

SUPERVISOR: OVE ANDERSSON

**1.ABSTRACT**

This project is about showing how efficient software defined wide area networks are by comparing it with traditional wide area networks in DevNET’s virtual environment sandbox.

(we will write summary of the main findings and summarize problem method and rslt)

**2.TABLE OF CONTENTS**

Chapter Page

1. Abstract 2
2. Table of Contents 3
3. Introduction 4
   1. Problem Background and How to Solve it 4
4. **INTRODUCTION**
   1. PROBLEM BACKGROUND AND HOW TO SOLVE IT

In today’s word, companies are getting larger and more complicated, so their needs in data trafficking. Configurating hundreds, thousands of devices is money and time consuming, and it is not just that. With the evolving technology, social media platforms, software as a service (SaaS) cloud technologies, people realized they need to be as independent as they could be from hardware.

In traditional networks, every router has a control plane which commands the data plane how to behave. This control plane is the software part of the device where we configurate it via CLI and such. In the other hand data plane is the hardware part where the data coming from other devices routing to one another by the current configuration. These two different entities are placed in the same place in a switch, acting as one entity.

By that and many, people started to thing about "Control plane being on every each device is a problem." and began to create the SDN.

The concept of the SDN is data plane and control plane are seperated.

The software part is pulled to a central point, so that it can see and manage the whole network, evulate it and decide on it.Now the control plane is in a central location and it can see the whole network, it has a acommand on the neighbor and routing tables of all switches on the network. Decision-making becomes smarter. On the other hand SDN is abstracting the infrastructure of the network from the applications. In this way, SDN makes it possible to data planes to be programmable.

With a centered, executive control plane, we are free from the dependency of the hardware and head to the software to change the whole network as once.

1. **MAIN**
   1. **A SIMPLE LOOK TO MPLS AND COMPARATION TO SD-WAN**

Multi Protocol Label Switching (MPLS) is a highly scalable and protocol-independent data transport system which is highly using from enterprises. It is a 2.5 layer protocol. In MPLS networks, tags are assigned to data packets, and the packet transfer process is done only depending on the contents of these tags, regardless of the contents of the packet. This allows the establishment of point-to-point communication circuits regardless of the mechanism and protocols used.

The most asked question is ¨Should I switch to SD-WAN from MPLS?¨. What are the pros and cons and which one has the brightest future?

The reasons MPLS is so populer around the world are the Reliability and high performance.

* MPLS circuits are a special service with performance backed by SLAs. This ensures that traffic routed over these circuits is reliably transmitted to its destination.
* MLPS circuits remove much of the complexity of routing traffic through the Internet. This enables them to provide higher performance than traffic reliant upon other routing schemes.

However, MPLS is not a perfect networking solution. It has downsides balancing the advantages.

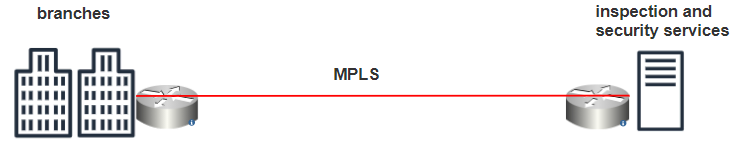
* **COST**: MPLS bandwidth is designed to provide high performance, reliable network connectivity. Even so this comes at a much higher price per bandwidth than a broadband Internet connection.
* **CENTRALIZATION**: Most MPLS connections are designed to send all traffic through a central hub then route to its destination. This can cause network latency due to inefficient routing.
* **GEOGRAPHIC LIMITATIONS**: MPLS circuits are a service offered by an ISP. If an MPLS provider is not available in a certain region, then it cannot be used to send traffic there.
* **SECURITY**: MPLS network doesn’t offer built-in data protection, and if incorrectly implemented, it can open the network to vulnerabilities.

SD-WAN roughly cover this downsides but yet, it is still proggtressing

Traditional solutions on WAN connections allow us to interconnect main offices with branch and that allows us to share resources. We can share access to centrally located data services or applications. Traditionally dedicated circuits have been used to achieve this connectivity things such as a frame relay or MPLS, although these provide reliability and security for the connection our modern networks require some rethinking of this cloud usage as we know is on the rise so we need ways to simplify the management of wide area networks, we also need a way to include our cloud resources.

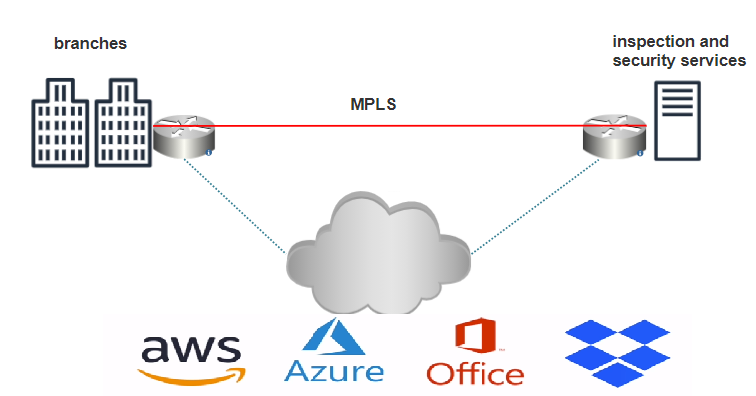
Software defined networks goal is to control and to manage the interactions between branch locations and our central resources. One another big advantages is that there's no longer need for backhauling your traffic

If we take a look at a very simplified topology,



We can see that we have a branch location connected over an MPLS circuit which is a traditional WAN connection method that's connected back to a data center. At the data center we have some advanced security and inspection happening so that's why we would tunnel all of our traffic from the branch back to the data center for inspection. In this traditional LAN setup all of the traffic from the branch is back hauled to the data center for those security services that includes traffic destined to the cloud or the public Internet and traffic destined locally within the organization as well. This can cause a myriad of performance issues, delay and depending on the circuit speed it can some bandwidth issues.

On the other hand in SD-WAN ,



Has the ability to interact with all kinds of cloud applications. more and more our applications are going cloud-based like how cheap and easy AWS storages now and how office 365 has turned to a cloud platform for enterprise email access SD wan can easily interact with applications like AWS Dropbox as your office 365 and many more. This means that hosting applications in either public or private clouds will allow direct traffic between the cloud application and the branch location. It will allow that rather than back hauling all of the traffic through a central data center. SD WAN can intelligently control the path of traffic in order to optimize traffic flow and to reduce unnecessary bandwidth in our networks. now you might

be wondering what happens to the

security inspection services since that

traffic is now not being routed through

the data center any longer that's been

addressed as well because most SD when

offerings now provide end-to-end traffic

encryption and inspection additionally

more and more next-generation features

are being added things such as

anti-malware systems and botnet

intervention many features that you

would see on next-generation security

devices another big advantage of SD lan

solutions is that they provide traffic

transport independence and in order to

understand that we need to understand

the difference in an overlay network and

an underlay network and underlay network

is simply the physical network

infrastructure responsible for the

delivery of packets we see that here in

this example we have several sites and a

data center all interconnected by

physical infrastructure sd1 on the other

hand that is a type of overlay network

and overlay networks are simply a

virtual network built on top of an

underlay network or built on top of the

actual physical infrastructure overlay

networks aren't anything new those have

been around for a long long time overlay

networks even include things like voice

over IP or VPNs those run on top of the

Internet as an overlay with sd1 as the

overlay network this gives us transport

independence and by this we mean that

our LAN connections can be made up of

all different types of connection

combinations we might have some LTE

connections we might have serial

connections wireless connections

satellite connections MPLS it doesn't

matter what's running as our underlay

network because SD Wang solutions can

very intelligently and coherently choose

the best data path transmission and

that's great news for

as network administrator's because it

simplifies our job even further now that

we understand what sd1 is and why that's

being increasingly used in our modern

enterprise networks in our next video we

want to take a look specifically at

Cisco's s dewayne solution and how we

would see that commonly implemented in

an enterprise network as we take a look

at sd1 implementation we want to

specifically examine the recommended

Cisco based solution at the time of this

recording that is Cisco's sd1 based on

VIP Atilla cisco acquired the VIP tella

company in 2017 which provides a

cloud-based sd1 solution this is

recommended to be used in conjunction

with Cisco's DNA Center and that will

allow you to leverage automation and

virtualization capabilities within Cisco

SD when we can break this down into four

planes the data plane the control plane

and the management plane which are

probably familiar with from traditional

networking models and the fourth plane

known as the orchestration plane this is

essentially shared with the management

plane there are four different solutions

within Cisco's SD when created to manage

each of these four different planes

the first is Cisco Vee managed which is

very simply the GUI interface for

managing the Cisco SD win solution this

is where you would perform configuration

monitoring and provisioning then there's

Cisco's V bond which controls the

orchestration plane it's the job of V

bond to understand how the network is

constructed and to make sure all of the

interconnected components can work

together one of the big capabilities

here is something called zero touch

provisioning this means that when an SD

when capable router is introduced into

the network Cisco's V bond can remotely

provision the router from anywhere

without the need for an administrator to

take any action at all so that's super

helpful for us as administrators next we

have

Cisco's V smart which resides within the

control plane this is thought of as the

brain of the sd1 solution as we create

policies envy manage as an administrator

creates those policies the V smart

component is responsible for the

enforcement of the policies those

policies are also shared with other SD

when routers and locations in our

network route information from branch

locations are received via the overlay

management protocol or OMP these Mart

can use the known policies against these

routes to control the traffic flow

through the SD WAM fabric and finally in

the data plane we have the actual LAN

edge routers themselves which are

responsible for establishing the network

and for forwarding traffic these devices

can be either physical or virtual or we

can have a combination of those these SD

when capable edge routers are referred

to as cisco v edge routers let's take a

look at a very common way that we might

see this implemented here we see a

sample topology within which we have a

main campus location a couple of branch

locations a physical data center and a

cloud data center all of these are

interconnected through various means we

see MPLS we see LTE and we see satellite

connections creating this network of

sites again remember that SD wanne is an

overlay technology and that provides

transport independence so in other words

it doesn't matter what the physical

underlying network infrastructure is as

we see here Sdn can work with any and

all of that at each of these locations

we would have a LAN edge router a cisco

v edge device and these routers form

IPSec tunnels with each other in order

to create the sd1 overlay network this

is going to make up the data plane that

we discussed a bit earlier also recall

that all of the control elements we

discussed earlier

Cisco's V managed V bond and V Smart

secure control channels

would be established between each of

these elements and each one of the LAN

edge routers and that would be used for

provisioning and for configuration of

the devices as for the edge routers

themselves these can be either a

hardware or software platform hardware

platforms includes cisco v edge routers

running on the VIP tella operating

system and certain integrated service

router and aggregation service router

models virtual platforms include the v

edge cloud router running VIP tella OS

and cloud services router models running

iOS xes deewan software so that's a look

at Cisco's SD wanne based on VIP tella

and how we would commonly see that implemented in an enterprise networ

1. **THE ACTION**

In this part of the report, we will apply the SD-WAN solution with Ansible using a sandbox and a VIRL supplied by Cisco.

The objectives of this act are:

* Use Ansible to dynamically build a VIRL topology from inventory data.
* Start a VIRL simulation using Ansible.
* Verify the VIRL simulation using virlutils.

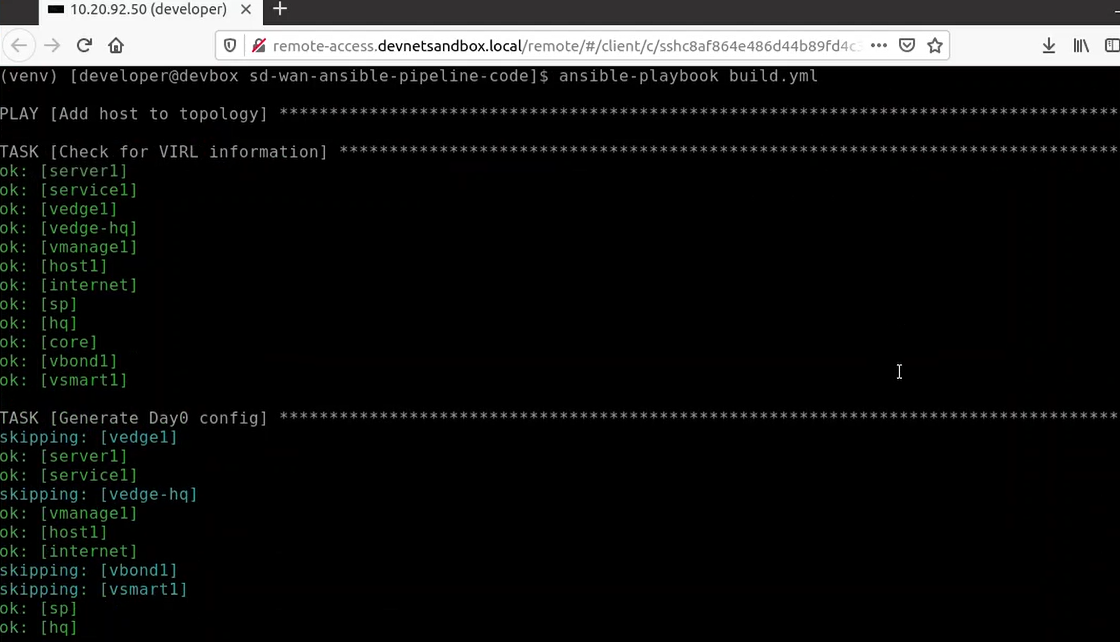
The basic process for automating VIRL topologies with Ansible is to:

* Create the required inventory data for each node in the simulation (stored in **virl.yml**)
* Run the **build.yml** playbook
  1. **STEP ONE**

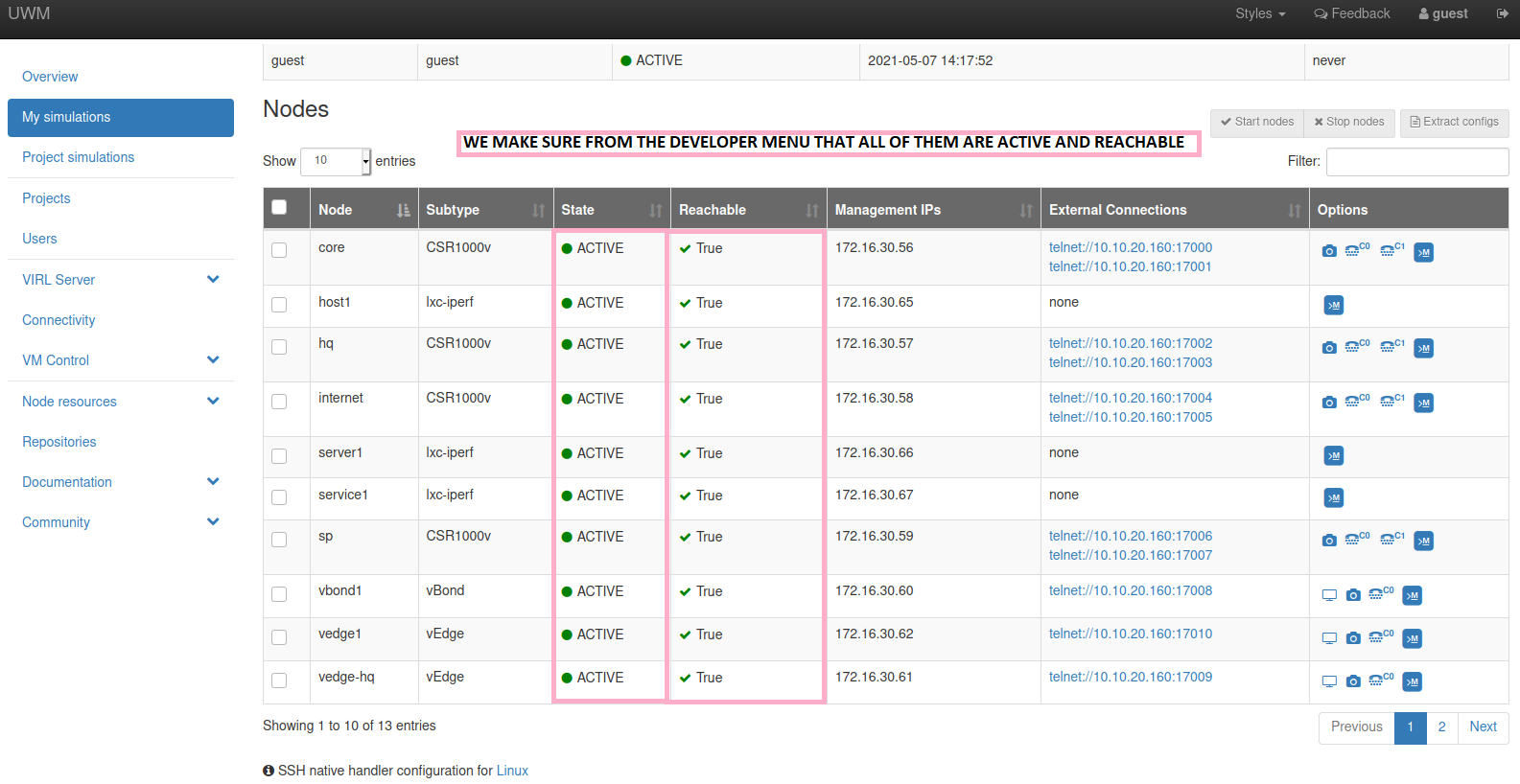
We connected to sandbox and VIRL via Cisco VPN so that we can access to SSH and developer mode

With the code below the started the Viptela SD-WAN simulation.

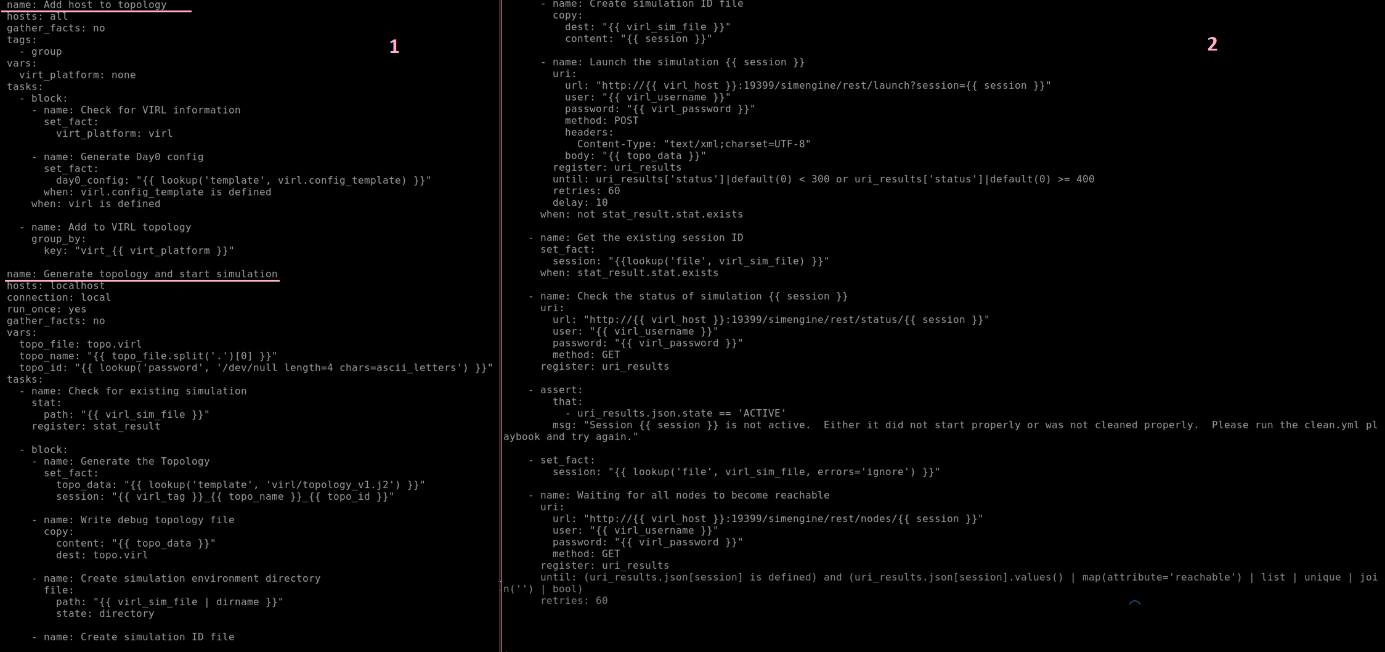
ansible-playbook build.yml







Now let’s take a look at the playbook, where the magic happens.



For every host in inventory (hosts: all), we check to see if it has any VIRL inventory data (when: virl is defined). If so, then we add it to the list of nodes to simulate in VIRL and generate its Day Zero configuration in the VIRL topology file using the Jinja2 template specified in the virl.config\_template variable.

Now that we have a list of hosts that need to be simulated in VIRL, the **Generate topology and start simulation** play will transform the VIRL data into a valid VIRL topology file and launch the simulation. In the **Generate topology and start simulation** play, view the **Generate the Topology** task. Where we generate the VIRL topology and store it as an Ansible fact (topo\_data) using a Jinja2 template (**topology\_v1.j2**).

Now let’s view the topology by this command,

cat templates/virl/topology\_v1.j2



This template is what assembles all the VIRL inventory data into a XML format needed to launch a VIRL simulation. The template includes three main sections that are required for a valid VIRL topology:

* + **Network Nodes** - the nodes and their Day Zero configuration.
  + **Networks** - the networks.
  + **Connections** - the connections between nodes and networks.

Most of the work is done in the Network Nodes section were we:

* + Iterate over all hosts with virl data defined.
  + Set the VIRL node type (e.g. CSR1000v, vEdge, etc.).
  + Set the Day Zero config, if defined.
  + Create a list of required networks.
  + Create a list of required connections between nodes and networks.

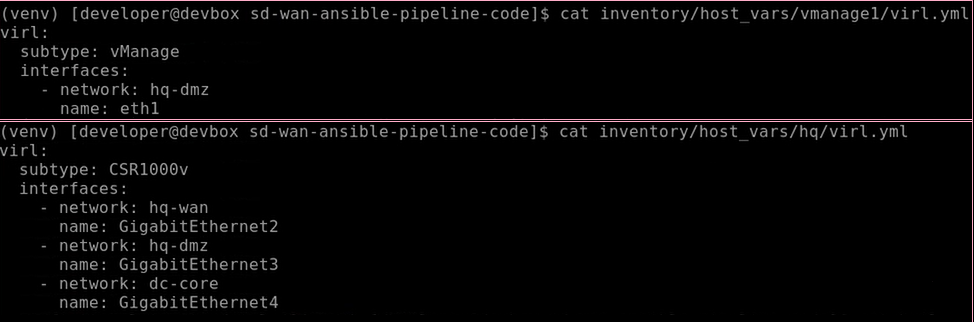
The final topology we created by build.yml as terminal output and schema.

[WE ARE CREATING THE TOPOLOGY SCHEMA AND WILL ADD IT HERE]

* 1. **STEP TWO**

The Jinja2 templates used by the build.yml playbook to dynamically generate a VIRL topology file based on Ansible inventory data are the 'magic' of the playbook. As a result, the Ansible inventory data becomes the foundation for our Infrastructure-as-code; if the inventory data changes, so does the resulting VIRL topology file. The relationship between Ansible inventory data and the VIRL topology is investigated in this step.

Each host in the Ansible inventory that we want to simulate in VIRL should have a **virl.yml** data file. In the photo down below you can see the VIRL inventory data for vmanage1 and the HQ router.



We specified a vManage node with its eth1 interface connected to the network hq-dmz. Note that we specify several properties under the 'virl:' block*:*

* *VIRL subtype*
* *interfaces/networks this node will have*
* *Day Zero template to use, if any*

The HQ node is type CSR1000v and has interface GigabitEthernet3 connected to the hq-dmz network. This is how we make connections in the VIRL topology between nodes. If two or more nodes have interfaces in the same network, then they will be directly connected. In this case, vmanage1 will be directly connected to the HQ router.

By combining the subtype, the interfaces, and the config\_template for each node, we can create any arbitrary VIRL topology with Ansible inventory data. In the next step, we examine how this data is converted into a VIRL topology file using the **build.yml** playbook and Jinja2 templates.