



netlab

Bringing the joy back to virtual networking labs

Ivan Pepelnjak (ip@ipSpace.net)
Network Architect

ipSpace.net AG

Who is Ivan Pepelnjak (@ioshints)

Past

- Kernel programmer, network OS, and web developer
- Sysadmin, database admin, network engineer, CCIE
- Trainer, course developer, curriculum architect
- Team lead, CTO, business owner
- SDN skeptic and network automation evangelist

Present

- Network architect, consultant, blogger, open-source developer

Focus

- SDN and network automation
- Large-scale data centers, clouds, and network virtualization
- Scalable application design
- Core IP routing/MPLS, IPv6, VPN



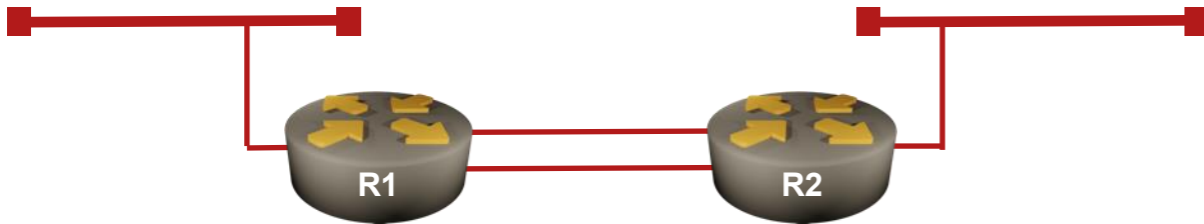
Also: I built too many labs for one lifetime, and hated that with passion

Based on a True Story

Pär Stolpe 01 September 2023 12:06

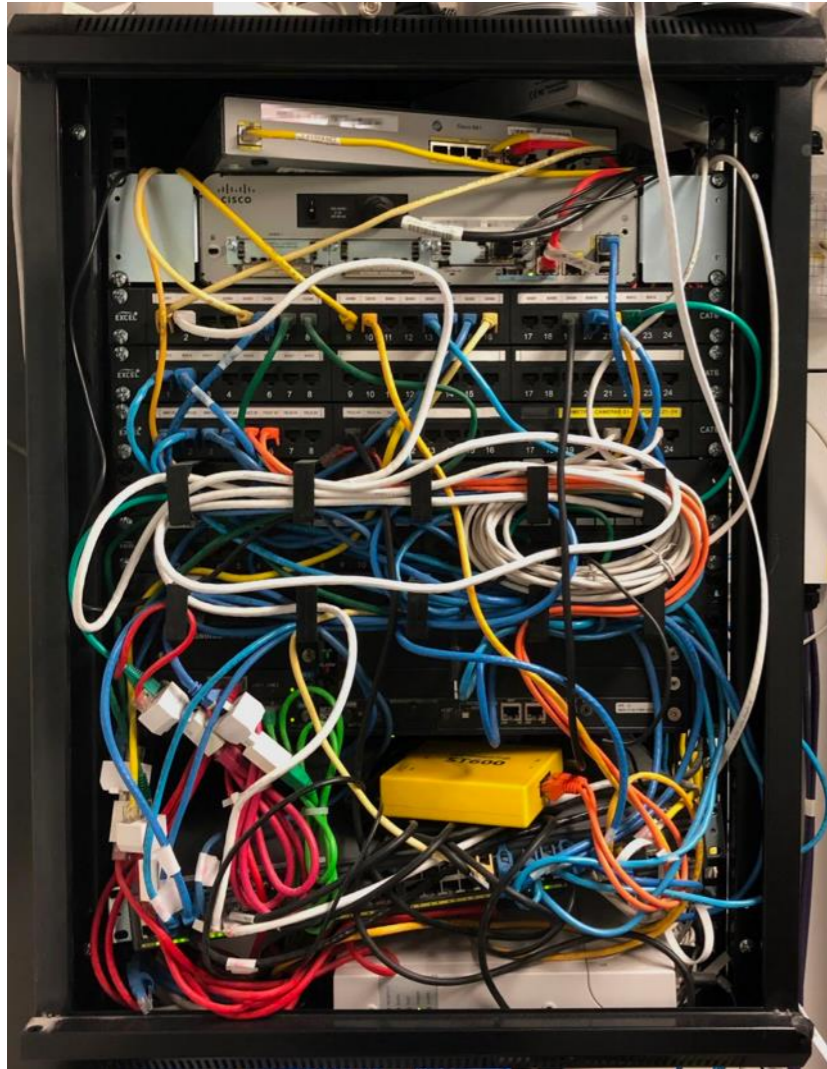
Beware of the fact that having more than one ospf link in between the same two nodes, together with unnumbered interfaces for multipathing purposes would most likely cause troubles. I don't know if any vendor have solved it or if they just recommend using link aggregation instead.

 **reply**



That should be trivial to test in a lab... However, someone has to build that lab...

The Reality Intervenes...



VM Maestro

Projects: My Topologies (routegen.virl, routeserver.virl, SpringBoard.virl)

Thbbft! virl:topology

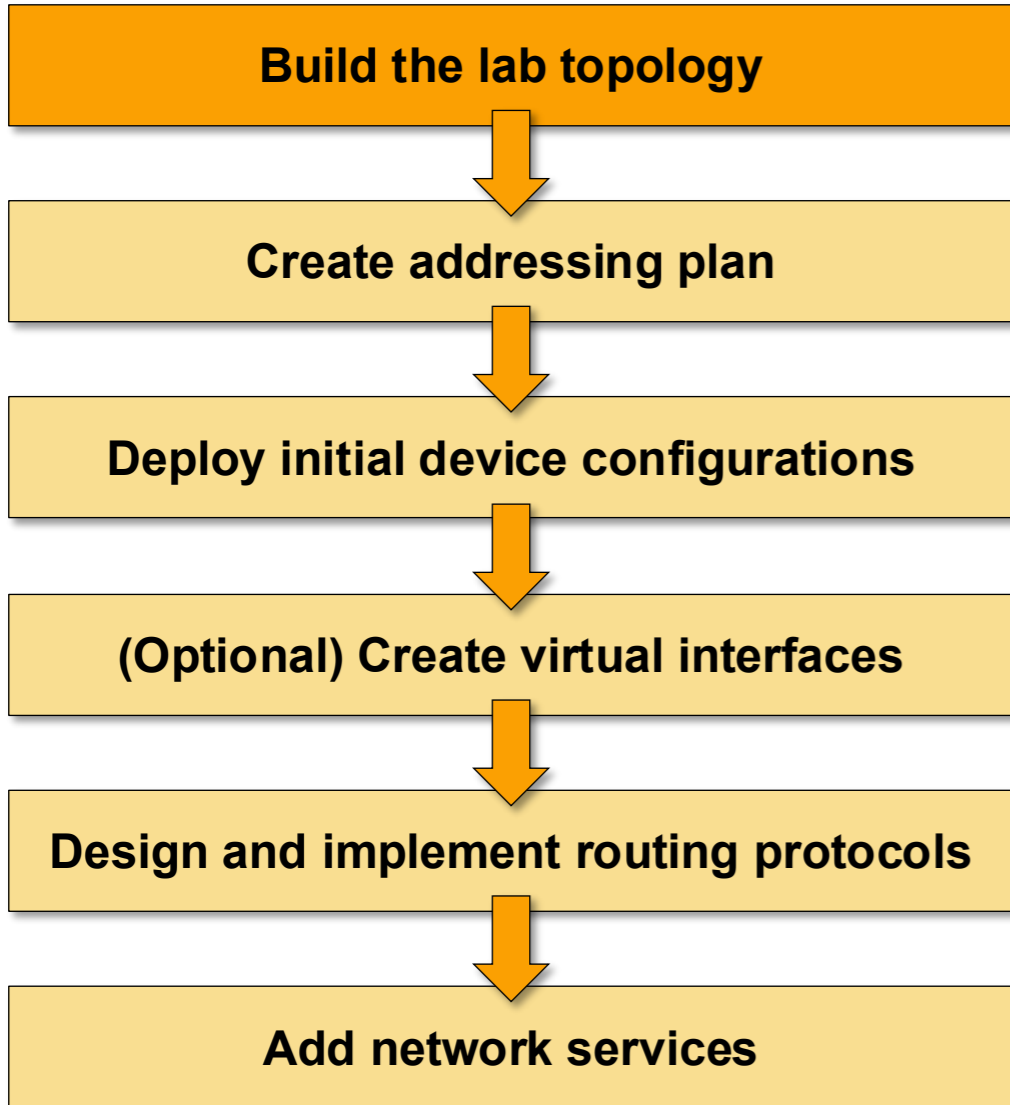
Simulations: Thbbft! (Last updated: Sun Jul 10 10:40:35 PDT 2016)

- guest
 - Thbbft!
 - Atlanta [ACTIVE]
 - Berlin [ACTIVE]
 - Boston [ACTIVE]
 - Dallas [ABSENT]
 - Denver [ACTIVE]
 - London [ACTIVE]
 - Milan [ACTIVE]
 - Paris [ACTIVE]
 - Seattle [ACTIVE]
 - Vienna [ACTIVE]
 - ~lxc-flat
 - External Address [172.16.1.51]
 - Forwarding Port on Server [10000]
 - ~mgmt-lxc interface [eth0]
 - ~mgmt-lxc [ACTIVE]

Console

```
Unknown simulation Thbbft!  
(INFO) [Jul/10/2016 16:46:30] Starting node "~mgmt-lxc"  
(INFO) [Jul/10/2016 16:46:45] Node "Berlin" state changed from BUILDING to ACTIVE  
(INFO) [Jul/10/2016 16:46:45] Node "Boston" state changed from BUILDING to ACTIVE  
(INFO) [Jul/10/2016 17:39:47] Stopping node "Dallas"  
(INFO) [Jul/10/2016 17:40:03] Node "Dallas" state changed from ACTIVE to ABSENT
```

A Networking Lab Is Much More than Topology





And now for something completely different

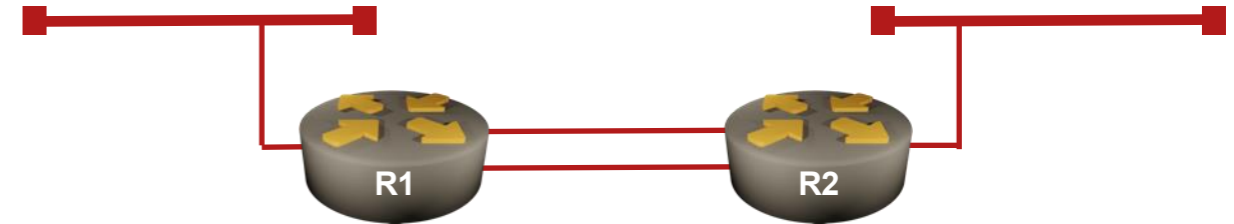
What Do We Need (in Unicorn Land of Infrastructure-as-Code)

Create a high-level description of the network

- Two devices: R1 and R2
- Let's make them Arista EOS containers
- They are running OSPF
- We need four links (two of them stub LANs)
- Oh, we're running unnumbered links...

Next

- Save the file
- Execute **netlab up** and you'll get a running network (including IP addressing and OSPF)



topology.yml

```
nodes: [ r1, r2 ]

defaults.device: eos
provider: clab

module: [ ospf ]

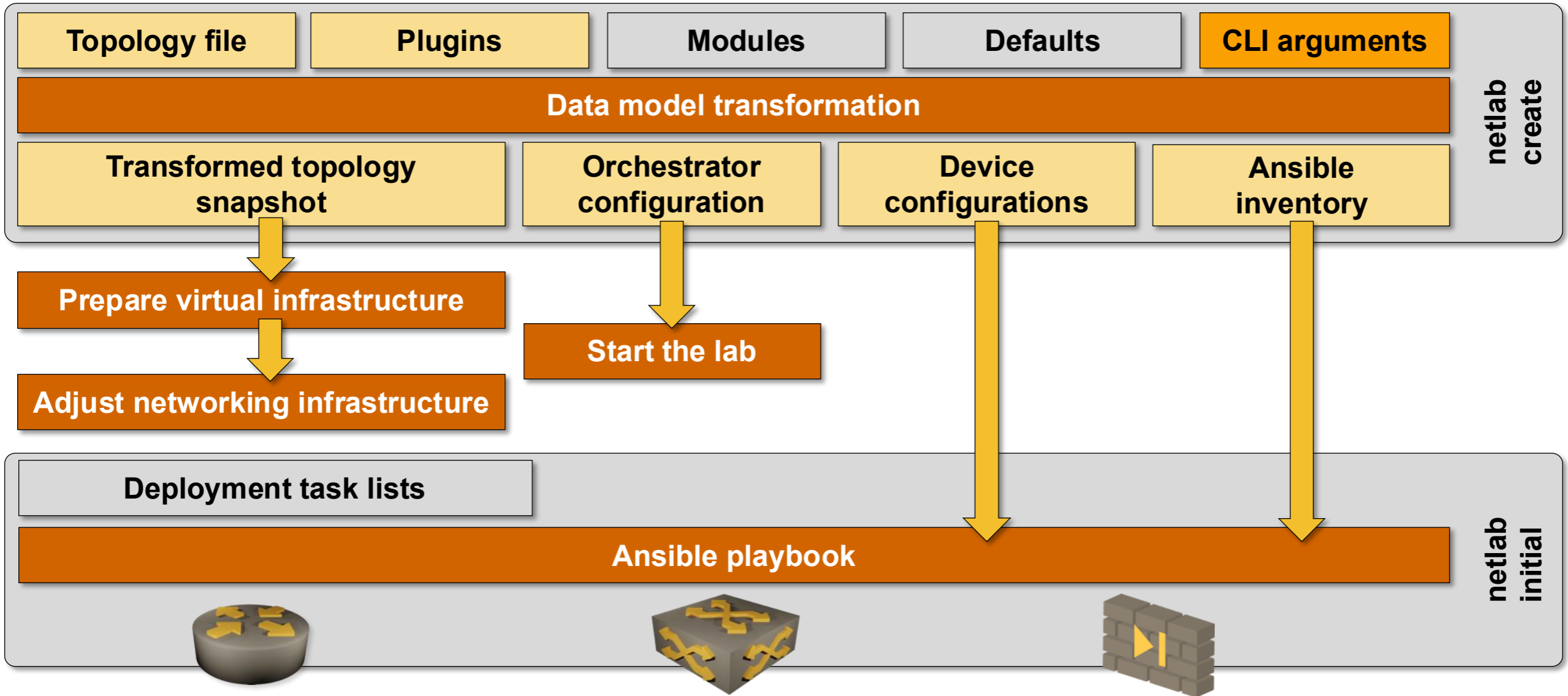
links: [ r1, r2, r1-r2, r1-r2 ]

addressing.p2p.ipv4: True
```

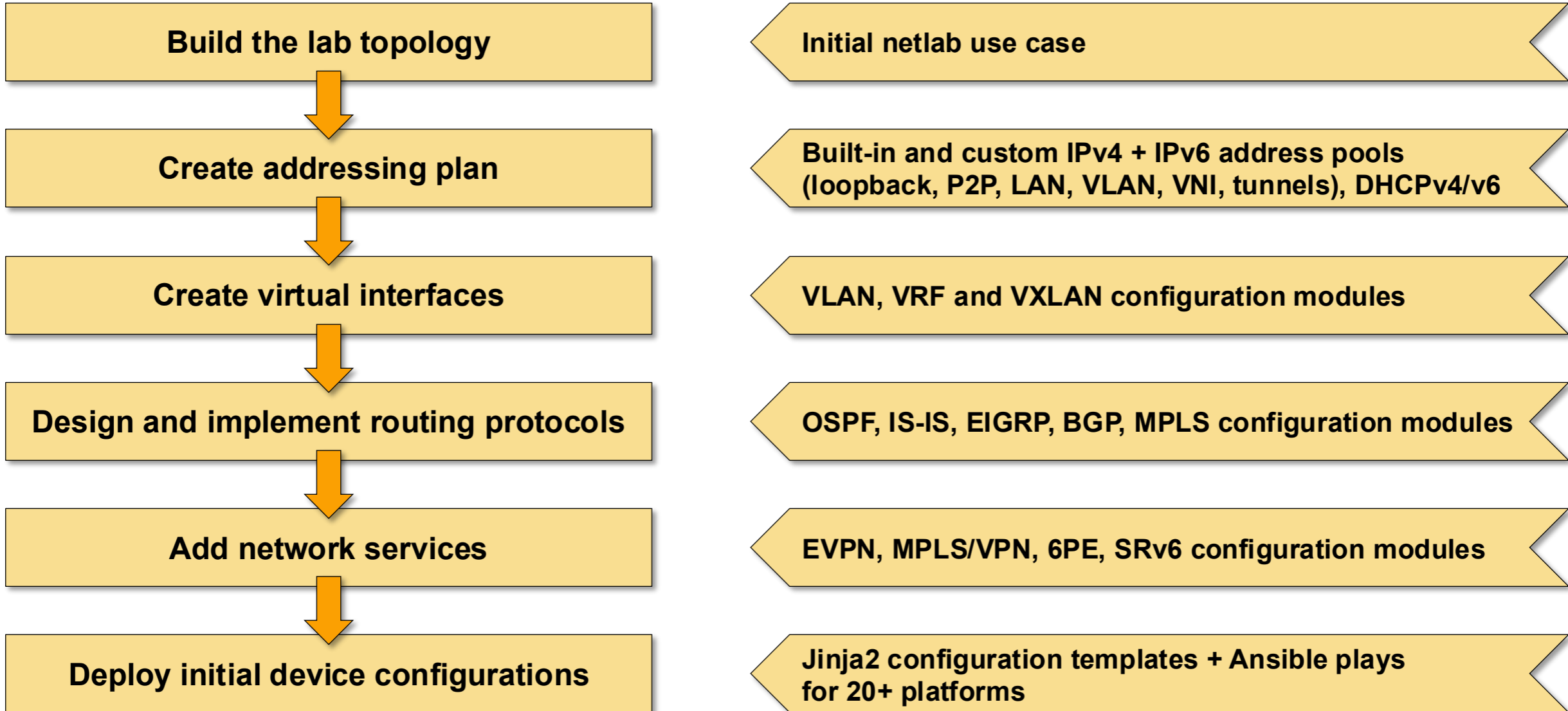
```
$ netlab up -p clab -d eos parallel.yml █
```

- Create configuration files
- Start the containers
- Start an Ansible playbook
- Initial device configuration
- Configuring OSPF
- Connect to the device
- ... and we have the answer!

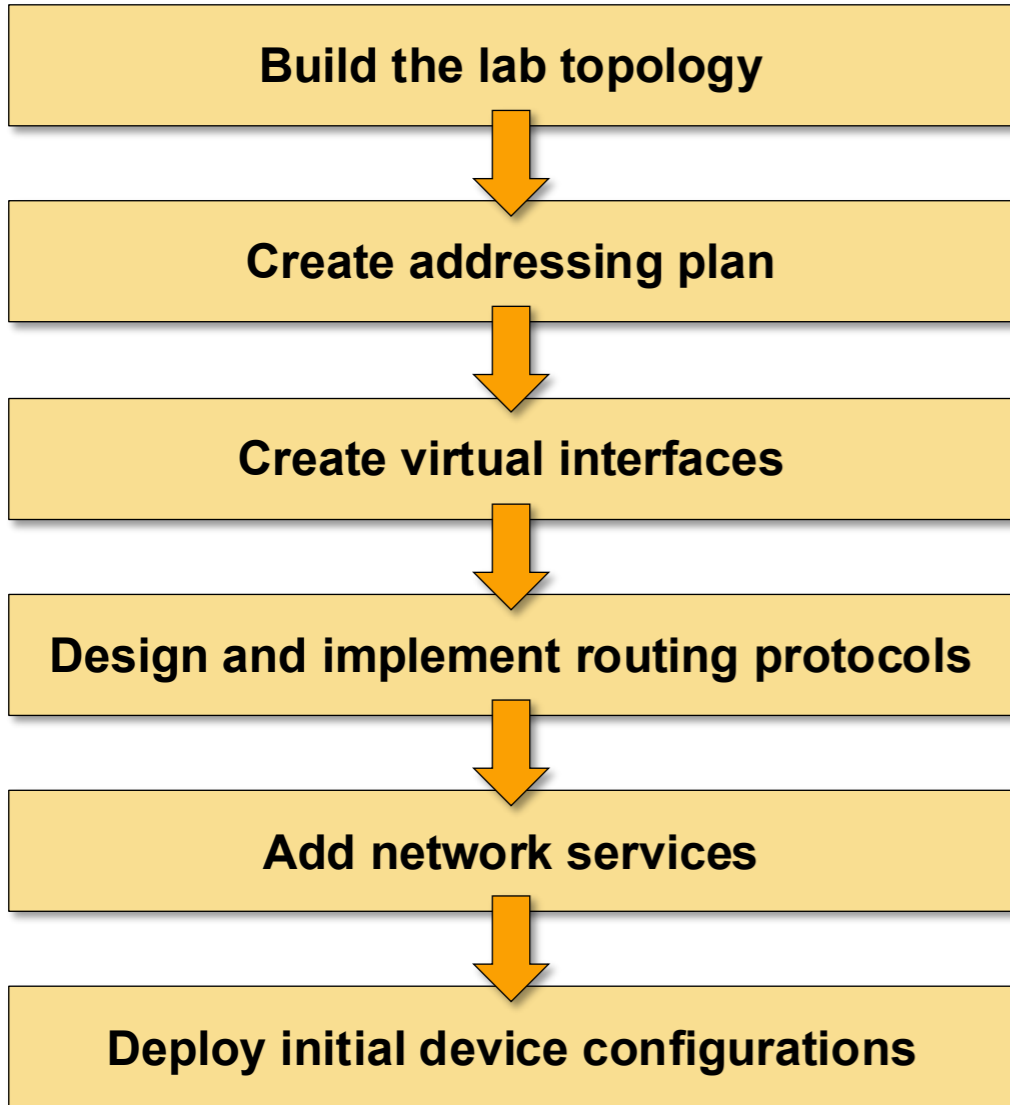
Wait, What Just Happened? netlab up Behind the Scenes



Building a Networking Lab with netlab



Every Lab Is Special



Add custom configuration

- Static configuration templates
- Configuration templates modifying built-in configurations
- Multi-vendor and multi-platform support

Modify data transformation with plugins

- Modify the built-in transformation rules
- Add new functionality (example: IP Anycast)
- Add new attributes or functionality to existing configuration modules (example: BGP add-path)

Current State of netlab (February 2026)

Network devices

- Arista vEOS/cEOS
- Aruba CX
- Cisco ASAv, Catalyst 8000v, Cisco 8000v, CSR 1000v, IOSv, IOSvL2, IOL, IOLL2, Nexus OS (9300v), IOS XRv/XRd
- Cumulus Linux 4.x and 5.x (NVUE)
- Dell OS10
- Fortinet
- FRRouting
- Juniper cRPD, vSRX 3.0, vMX, vJunos-evolved (vPTX), vJunos-switch, vJunos-router
- Mikrotik RouterOS 6 and 7
- Netscaler CPX
- Nokia SR Linux, SR-SIM, and SR OS
- Sonic
- VyOS 1.4 and 1.5

Hosts and daemons

- Generic Linux host or container
- OpenBSD
- BIRD
- dnsmasq
- Kubernetes in Docker (KinD) cluster

Virtualization providers

- KVM with libvirt (Vagrant)
- Docker (containerlab)
- Hardware labs (requires extra interface information)
- VirtualBox (Vagrant) (deprecated)

Multi-provider topologies

- Combine containers and virtual machines in the same lab
- Connect external devices with the virtual lab

Current State of netlab: Behind the Scenes

Network devices

- Arista vEOS/cEOS
- Aruba CX
- Cisco ASAv, IOSv, IOS XE (CSR), Nexus OS (9300v), IOS XR/XRd
- Cumulus Linux 4.x and 5.x (NVUE)
- Dell OS10
- Fortinet
- FRR
- Juniper vSRX 3.0, vMX, vPTX (vEVO)
- Mikrotik RouterOS 6 and 7
- Nokia SR Linux and SR OS
- VyOS 1.4 and 1.5

The Heroes



Stefano Sasso

Aruba, Junos, Dell OS10,
Mikrotik, VyOS



Jeroen van Bommel

Cumulus, Dell OS10, FRR,
Nokia SR Linux,
Nokia SR OS

Current State of netlab (February 2026)

Addressing

- IPv4 + IPv6, configurable IPv6 RA parameters
- Address pools and named prefixes
- VLAN-wide subnets
- Static subnets and interface addresses
- Unnumbered IPv4 and IPv6 (LLA) interfaces
- Layer-2-only interfaces
- DHCP (clients, servers, relays)

Data Plane

- VLANs and VRFs
- VXLAN (static ingress replication or EVPN)
- MPLS including SR-MPLS
- SRv6
- Tunnel interfaces
- Zone-based firewalls

Routing and Routing Protocols

- OSPFv2, OSPFv3, IS-IS, RIP, EIGRP, BGP
- BFD
- Static routes
- Routing policies
- Route redistribution and default routes
- Prefix filters, AS-path filters, BGP community filters
- VRRP and anycast gateways

MPLS Control Plane

- LDP, BGP-LU, SR-MPLS (OSPF or IS-IS), SRv6

Network Virtualization

- MPLS L3VPN and 6PE, L3VPN over SRv6
- EVPN over VXLAN or MPLS
(bridging, VLAN bundles, asymmetric and symmetric IRB, multihoming, most combinations of IGP and BGP)

Sample Platform Support

Operating system	Hostname	IPv4/IPv6 hosts	LLDP	IPv4 Loopback	IPv6 Loopback
Arista EOS	✓	✓	✓	✓	✓
Aruba AOS-CX	✓	✗	✓	✓	✓
Cisco ASAv	✓	✓	✗	✗	✗
Cisco IOS/IOS XE ³	✓	✓	✓	✓	✓
Cisco IOS XR ⁴	✓	✓	✓	✓	✓
Cisco Nexus OS	✓	✓	✓	✓	✓
Cumulus Linux	✓	✓ ¹²	✓	✓	✓
Cumulus Linux 5.x (NVUE)	✓	✓	✓	✓	✓
Dell OS10	✓	✓	✓	✓	✓
Fortinet FortiOS	✓	✗	✓	✓	✓
FRR	✓	✓ ¹²	✗	✓	✓
Generic Linux	✓	✓ ¹²	✓!	✓	✓
Junos ⁵	✓	✗	✓	✓	✓
Mikrotik RouterOS 6	✓	✓	✓!	✓	✓
Mikrotik RouterOS 7	✓	✓	✓!	✓	✓
Netscaler CPX	✓	✗	✗	✗	✗
Nokia SR Linux	✓	✓	✓	✓	✓
Nokia SR OS ⁶	✓	✓	✓	✓	✓
OpenBSD	✓	✓	✗	✓	✓
Sonic	✓	✓	✗	✓	✓
VyOS	✓	✓	✓	✓	✓

Operating system	OSPF	IS-IS	EIGRP	BGP	RIPv2/ng
Arista EOS	✓	✓	✗	✓	✓
Aruba AOS-CX	✓	✗	✗	✓	✗
Cisco ASAv	✓!	✓!	✗	✓	✗
Cisco IOSv/IOSvL2	✓	✓	✓	✓	✓
Cisco IOS XE ³	✓	✓	✓	✓	✓
Cisco IOS XR ⁴	✓	✓	✗	✓	✗
Cisco Nexus OS	✓	✓	✓	✓	✗
Cumulus Linux	✓	✗	✗	✓	✓
Cumulus Linux 5.x (NVUE)	✓	✗	✗	✓!	✗
Dell OS10	✓!	✗	✗	✓	✗
Fortinet FortiOS	✓!	✗	✗	✗	✗
FRR	✓	✓	✗	✓	✓
Junos ⁵	✓	✓	✗	✓	✗
Mikrotik RouterOS 6	✓	✗	✗	✓	✗
Mikrotik RouterOS 7	✓	✗	✗	✓	✗
Nokia SR Linux	✓	✓	✗	✓	✗
Nokia SR OS ⁶	✓	✓	✗	✓	✓
OpenBSD	✓	✗	✗	✗	✓
Sonic	✗	✗	✗	✓	✗
VyOS	✓	✓	✗	✓	✗

Operating system	BFD	EVPN	MPLS/VPN	FHRP
Arista EOS	✓	✓	✓	✓
Aruba AOS-CX	✓	✓	✓	✓
Cisco CSR 1000v	✓	✗	✓	✓
Cisco Catalyst 8000v	✓	✓	✓	✓
Cisco IOL/IOLL2	✓	✓	✓	✓
Cisco IOSv/IOSvL2	✓	✗	✓	✓
Cisco Nexus OS	✓	✓	✗	✓
Cumulus Linux	✓	✓	✗	✓
Dell OS10	✓	✓	✗	✗
FRR	✓	✓	✓	✗
Juniper vMX	✓	✗	✓	✓
Juniper vPTX	✓	✗	✓	✓
Juniper vSRX 3.0	✓	✗	✓	✓
vJunos-switch	✓	✓	✓	✓
vJunos-router	✓	✗	✓	✓
Mikrotik RouterOS 6	✓	✗	✓	✗
Mikrotik RouterOS 7	✓	✗	✓	✗
Nokia SR Linux	✓	✓	✗	✓
Nokia SR OS ⁶	✓	✓	✓	✓
VyOS	✓	✓	✓	✗

But Wait, There's More (February 2026)

External Connectivity

- Macvtap libvirt interfaces
- Macvlan container interfaces
- Port forwarding for VMs and containers

External Tools

- SuzieQ
- Graphite
- Edgeshark
- Cisco NSO
- Network Unit Testing System (NUTS)

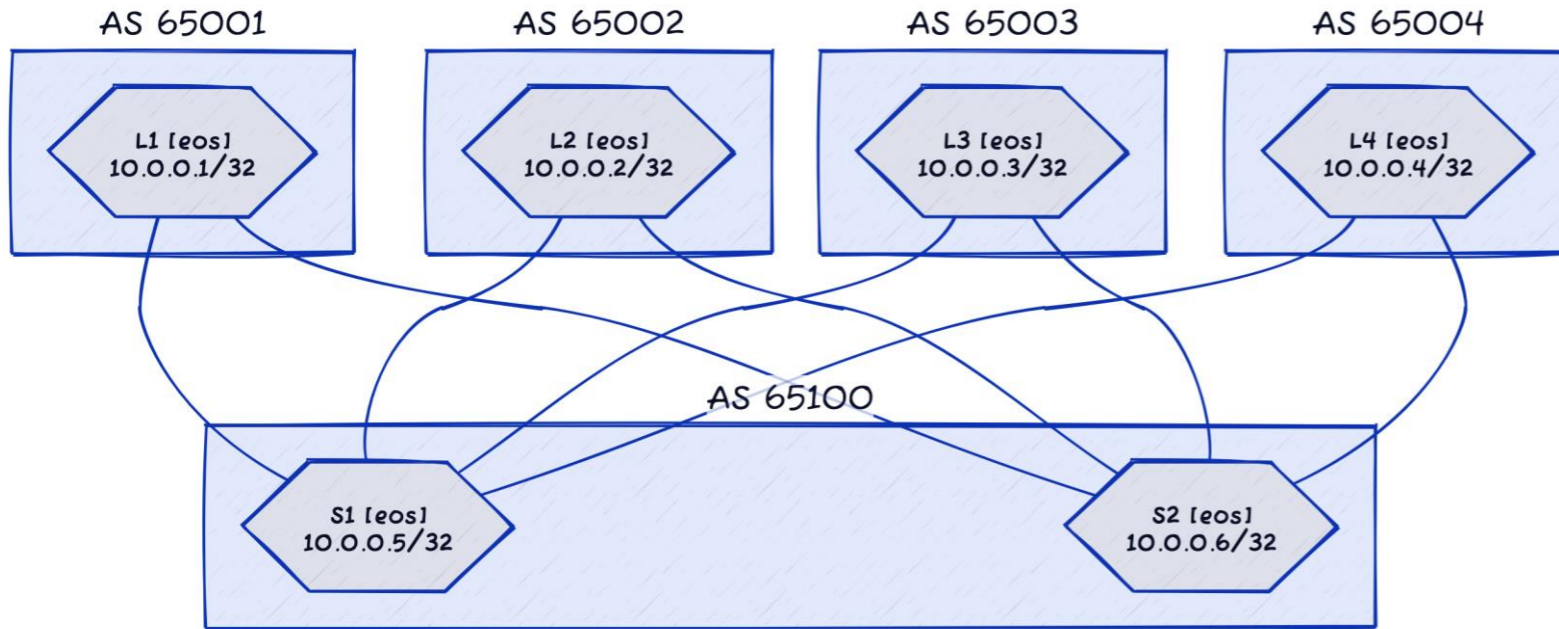
Ease-of-use

- Wiring, addressing, OSPFv2, OSPFv3, IS-IS, and BGP reports (text, Markdown, HTML)
- Graphs (Graphviz or D2)
- Graphite GUI 🍦 🍦 🍦
- Automated validation
- Multiple lab instances on a single server
- Restore previous configurations

Large topologies

- Topology components
- Staggered device start
- Link groups

Example: Graph (D2)



Example: Addressing Report (BGP)

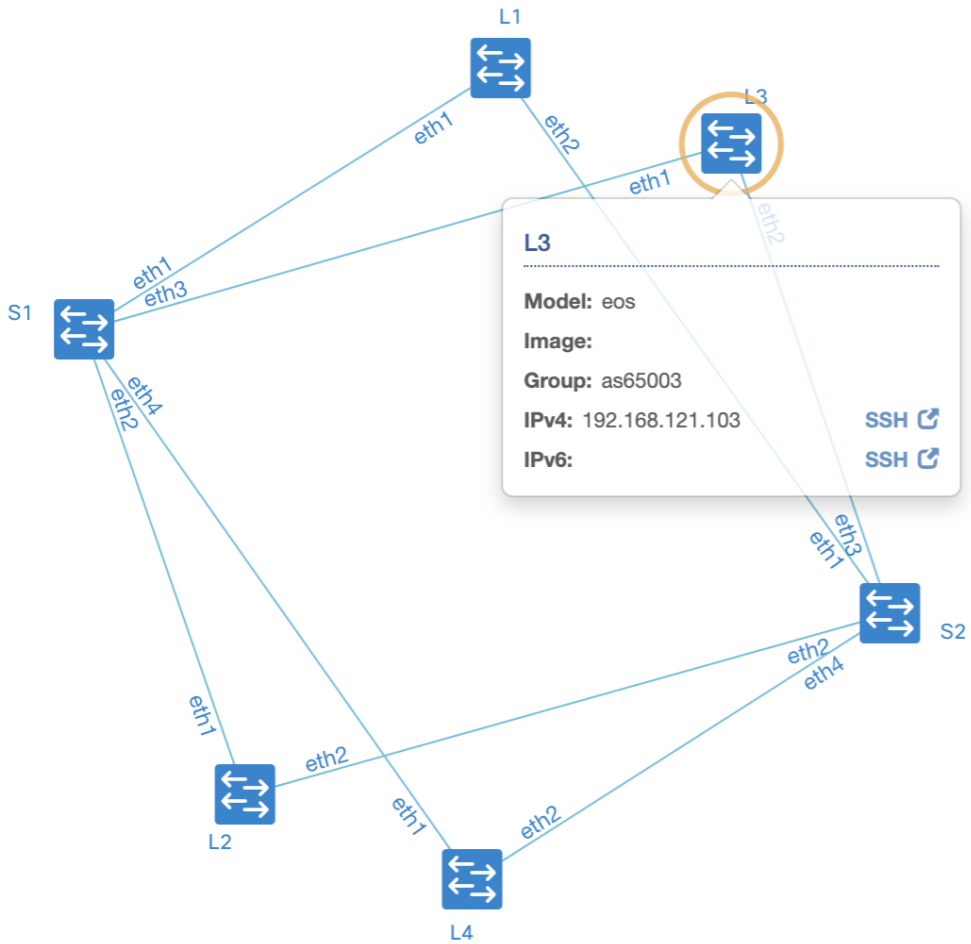
BGP AS Numbers

NODE/ASN	ROUTER ID	ADVERTISED PREFIXES
AS65001		
L1	10.0.0.1	10.0.0.1/32
AS65002		
L2	10.0.0.2	10.0.0.2/32
AS65003		
L3	10.0.0.3	10.0.0.3/32
AS65004		
L4	10.0.0.4	10.0.0.4/32
AS65100		
S1	10.0.0.5	10.0.0.5/32
S2	10.0.0.6	10.0.0.6/32

BGP Neighbors

NODE	NEIGHBOR	NEIGHBOR AS	NEIGHBOR IPV4
L1 (10.0.0.1 / AS 65001)			
	S1	65100	10.1.0.2
	S2	65100	10.1.0.6
L2 (10.0.0.2 / AS 65002)			
	S1	65100	10.1.0.10
	S2	65100	10.1.0.14
L3 (10.0.0.3 / AS 65003)			
	S1	65100	10.1.0.18
	S2	65100	10.1.0.22
L4 (10.0.0.4 / AS 65004)			
	S1	65100	10.1.0.26
	S2	65100	10.1.0.30

Example: Graphite GUI



L3

Model: eos
Image:
Group: as65003
IPv4: 192.168.121.103 [SSH](#)
IPv6: [SSH](#)

Example: Network Validation

```
$ netlab validate
[session] Check IPv6 EBGP session with RTR on ISP routers [ node(s): x1,x2 ]
[PASS] Validation succeeded on x1
[PASS] Validation succeeded on x2
[PASS] The IPv6 EBGP session with RTR is in Established state

[pfxcnt] Check whether RTR receives and sends IPv6 prefixes [ node(s): x1,x2 ]
[PASS] Validation succeeded on x1
[PASS] Validation succeeded on x2
[PASS] RTR is advertising IPv6 prefixes to ISP routers

[advroute] Check whether RTR advertises 2001:db8:1::/48 [ node(s): x1,x2 ]
[PASS] Validation succeeded on x1
[PASS] Validation succeeded on x2
[PASS] RTR is advertising 2001:db8:1::/48 to ISP routers

[SUCCESS] Tests passed: 6
$ █
```

Use Cases

Rapid Prototyping

” Ever had a networking question where you think "I wish I could test that in a lab"... and then remember how long it takes to spin one up from scratch?

I recently had an iBGP session refusing to establish between two devices. It took me longer to draw the topology in Excalidraw than to write the netlab topology file — 8 devices, a mix of eBGP/iBGP/IS-IS/OSPF — so I could recreate the scenario and test routing and redistribution. How long would that have taken manually? A lot longer than I'd like to admit.

Sebastien d'Argoeuves, Solutions Architect, IP Fabric



Rapid Prototyping

” I use netlab to quickly mock up scenarios a practitioner might face as I prepare episodes for the educational podcast N Is For Networking.

I’ve successfully stood up labs with dozens of virtual routers and switches. For me, the real netlab magic is the time savings. I spec the lab in YAML using just a few lines to describe NOS, which devices should be connected to which other ones, what modules I want to run (OSPF, BGP, VLANs, etc.) and then type “netlab up”. I have a fully functional lab in just a few minutes.

Ethan Banks, Packet Pushers



Mastering New Technologies or Vendor Platforms

” I use netlab as a tool to learn new technologies and doing labs. This is how I learned IS-IS, and how I learn EVPN through online labs. Netlab is the best in class here. Device abstraction, separation of topology from node types, auto-configuration, templating user configs, ready for Ansible/Nornir with automatic inventory creation.

Dan Partelly, Independent IT and Services Professional

” How do you configure MPLS on Juniper, OSPF on Fortigate, VXLAN on Aruba... without losing a weekend, or more? I'd rather spend my time *designing* the lab than figuring out the right command for vendor #7. Even when the CLI looks familiar, it's never *quite* the same.

Sebastien d'Argoeuves, Solutions Architect, IP Fabric



More @ <https://blog.ipSpace.net/2025/12/netlab-configuration-translator/>

Classroom Training

” I use netlab when teaching. It makes it very easy to set up labs for my students, and with the new **skip_config** option, I can let netlab configure my virtual devices, but leave some parts for the students to figure out :)

Sander Steffann, IPv6 and network automation consultant



” Netlab enables me to conduct training sessions efficiently, going straight to the relevant topics without wasting time repeatedly setting up basic configurations.

Stefano Sasso, Principal Solutions Architect (Telco/5G)



Online Labs



Severin Dellsperger · 1st
Lecturer | Network & Research Engineer
2d · 🌐

Today I prepared for my OSPF lecture next week. I'm a big fan of giving live demos to understand the theory behind a technology. I wanted to create an easy-to-understand lab on OSPF areas and LSA distribution. It was the first time I actually used GitHub Copilot to do the work.

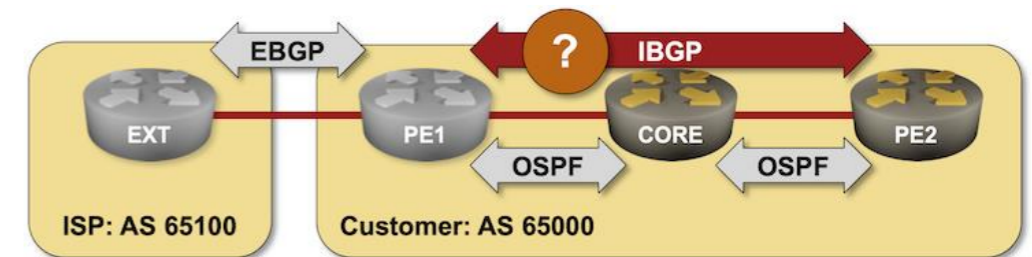
I'm impressed by the result: I was able to quickly setup a netlab lab. Especially, I was fascinated by how it could easily create a network diagram or the netlab config with a few hints here and there.

Is it perfect? Absolutely not, but I'm quite happy with how fast and effortless it was.

If you also want to refresh your OSPF knowledge, here is the GitHub link: <https://lnkd.in/e9NJrJAq>

More examples:

- bgplabs.net (50+ labs)
- isis.bgplabs.net (20+ labs)
- evpn.bgplabs.net (~ 20 labs)

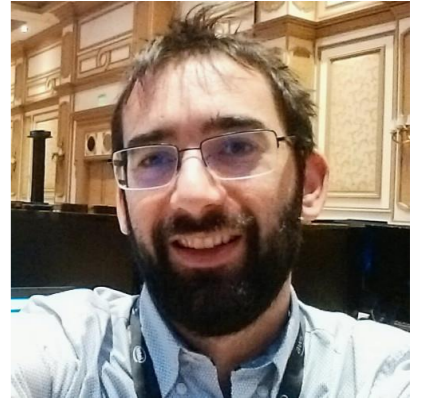


More @ <https://github.com/severindellsperger/netlab-ospf-lab>

Digital Twins (🧑🏻♂️)

” Netlab is, for me, the Swiss Army knife of networking and network automation. It allows me to focus on what truly matters: validating new configurations and creating digital twins of networks without the complexity of configuring everything manually from scratch.

Stefano Sasso, Principal Solutions Architect (Telco/5G)



” ... as part of change management in an enterprise network - validating production configuration changes before deploying them in the field. Specifically on complex legacy setups involving Dell OS10 and NVidia Cumulus devices.

Jeroen van Bemmelen, AI Systems Engineer



Setting Up Demos

” Netlab has been a lifesaver. Ivan’s entire approach, from the software to collecting instructions and providing a meaningful information trail, enabled me to go from zero to having a functional lab in minutes. It has been an absolute lifesaver.

I can be lazy with the infrastructure side, because he’s done all of the hard work. Now I get to concentrate on the value-added functionality of my own systems and test with the full power of an automated and modern network lab. Game-changing.

David Gee, Managing Director, Curvium

” Netlab provides a fast and effective way for me to demonstrate configurations or features of specific vendors.

Stefano Sasso, Principal Solutions Architect (Telco/5G)



Arguing with Vendor TAC

” I recently “leveraged” 🤪 to reproduce a topology and configuration that caused an outage that should not happen during a change. The team didn't have the luxury and time to extract tech support dumps from all the possibly involved devices. They had to roll back the change as expediently as possible.

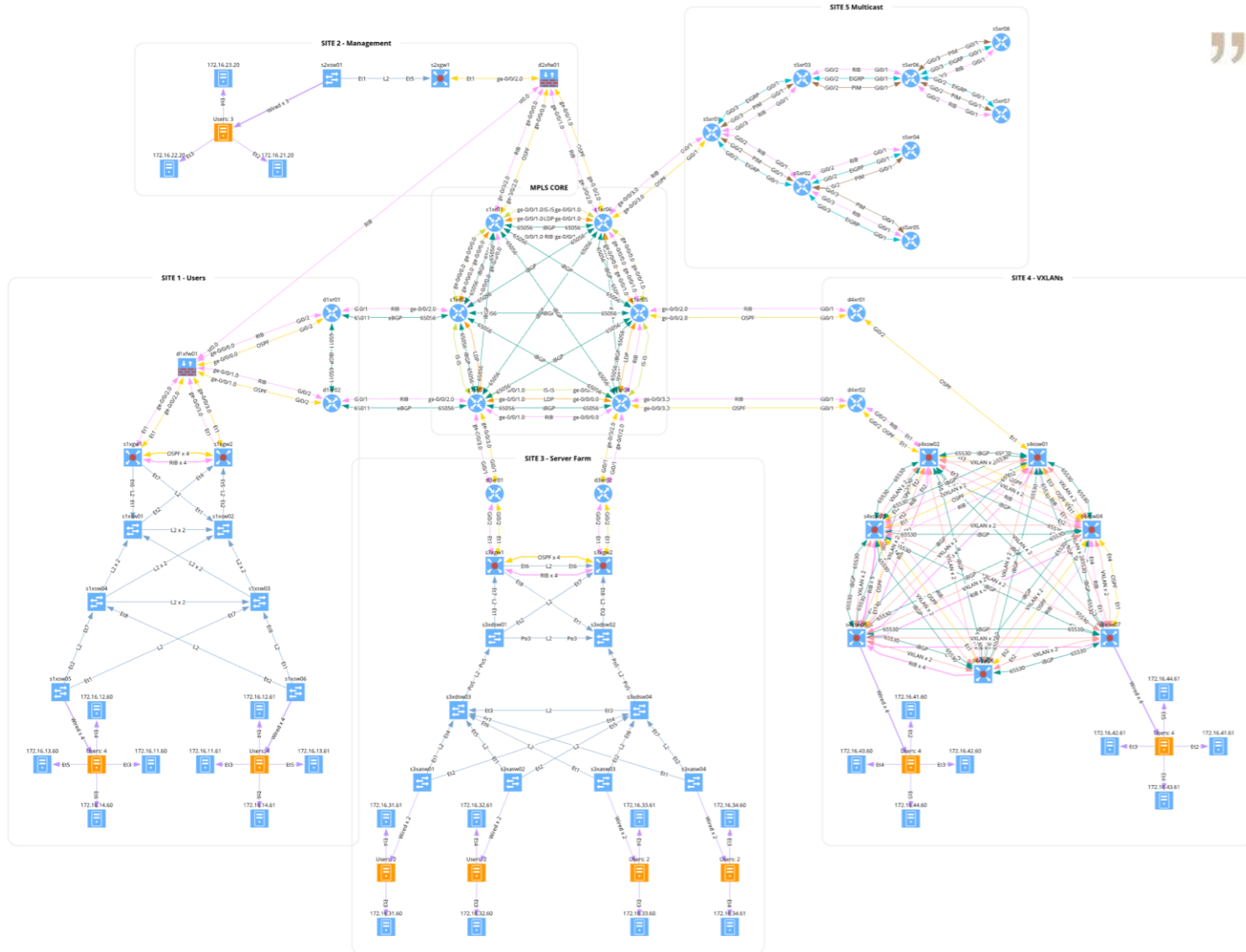
A TAC-loop ensued where we kept sending them drawings, a show command output illustrating the problem, configs and explanations, but they kept asking for more data. Of course, tech support dumps being amongst that. One TAC engineer even persisted in calling the situation 'works as expected'.

I subsequently recreated the physical topology in netlab with the minimum of devices involved. Devices from other vendors didn't cause an outage, but I could reproduce the outage with virtual devices from our vendor.

We're now going to send \$vendor the topology.yml instead of continuing with the rinse-repeat TAC cycle.



Large-Scale Labs for Software Testing



”

Our main lab: 8 vendors, 11 families, 65 devices. And when I say "infrastructure as code," I mean it — you can nuke the whole thing, and we'll rebuild it from scratch back to the defined state. netlab gets the lab to a working state fast, for all supported devices. IP addressing for point-to-point links? Management? Loopbacks? We don't care — just pick whatever. That's netlab's job, not ours.

Sebastien d'Argoeuves, Solutions Architect,
IP Fabric

Building Scenarios for Software Testing

” We use netlab for testing various scenarios for SuzieQ. Its hard for me to keep track of every release and OS's nits to figure out how to configure features on a platform. Furthermore, other people who don't know or have experience with specific platforms but understand the concepts can easily use netlab to setup test scenarios. Suffice it to say, without netlab, our lives would be very hard :)

Dinesh Dutt, CEO and cofounder, Stardust Systems



To Recap

” I’ve used GUI-based network labbing tools for decades, but defining my lab in YAML has proven to be the fastest mean time to productivity of any labbing platform I’ve tried. **Netlab is the “clicky-clicky” eliminator.**

Ethan Banks, Packet Pushers



Randy Bias ✓ • 2nd

Cloud Pioneer, Technology Disruptor, and Master Strate...

1w • 🌐

 **Connect**

If you haven't played with netlab yet and you're a network engineer; you are missing out.



Ivan Pepelnjak • You

Independent network architect, webinar author and prolific blogger at ipSp..

1w • 🌐

The changes I made in netlab releases 25.12 through 26.02 to stay as far away from Ansible as possible had an interesting side effect: it became much easier to generate device configurations with minimal installation effort.



Lou D. • 2nd

Network Security Engineer | Network Automation

4h (edited) ...

I fully agree with Ivan's point on the show about being exhausted building labs . Netlab is really one the best tools to help me build fairly complex network designs quick and also cheap ! I don't need so much resource provisioning to get something started a tiny aws Ubuntu box will build an decent amount of nodes .

Like · 🗳️ 2 | Reply · 1 Reply

A man with dark hair and a beard is sitting at a bar, looking out a window with a thoughtful expression. He is holding a glass of dark beer with a white head of foam. The scene is dimly lit, with warm light coming from the window and some blurred lights in the background.

SO WHAT ARE YOU WAITING FOR?



pip3 install networklab

Deployment Scenarios

**Open-Source BGP
Configuration Labs**

[Deploy BGP in Your Network](#)

[BGP Routing Policies](#)

[Challenge Labs](#)

[Recent Labs](#)

Open-Source BGP Configuration Labs

This series of BGP hands-on labs will help you master numerous aspects of EBGP, IBGP, and BGP routing policy configuration on a [platform of your choice](#)¹, including:

- Arista EOS
- Aruba AOS-CX
- Cisco ASAv, IOSv, IOS XE, IOS XR and Nexus OS
- Cumulus Linux and FRR
- Dell OS10
- Juniper vSRX, vMX and vPTX
- Mikrotik RouterOS
- Nokia SR OS and SR Linux
- Vyatta VyOS

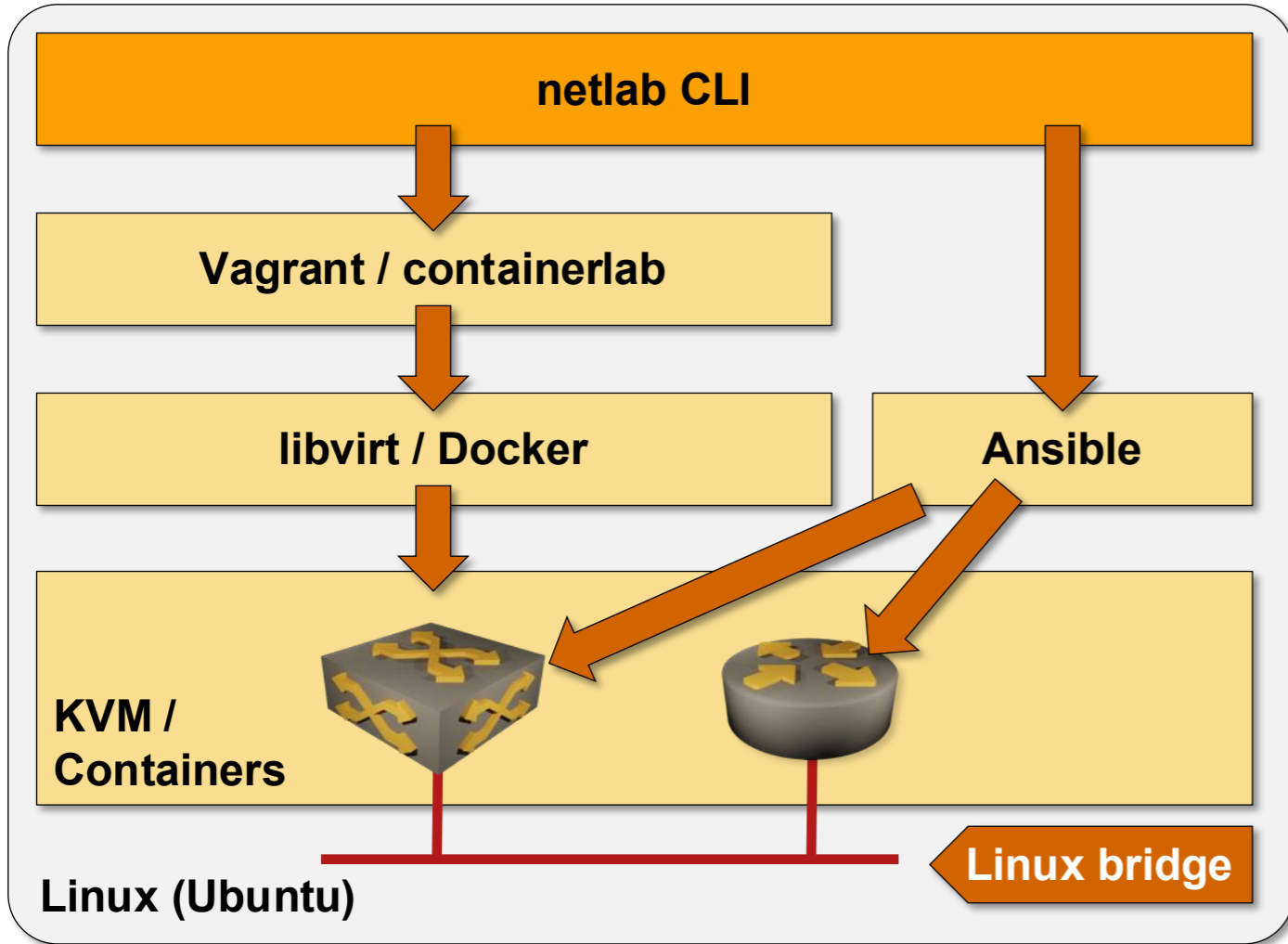
Dozens of labs are already waiting for you (with more [coming soon](#)), but if this is your first visit to this site, you should start with the [Installation and Setup](#) documentation or [run BGP labs in GitHub codespaces](#).

Deploy BGP in Your Network

In the first set of the BGP labs, you'll master these skills:

- [Configure and monitor routing daemons on Cumulus Linux and FRRouting](#)
- [Configure BGP sessions and advertise IPv4 and IPv6 prefixes](#)
- [Protect BGP sessions](#)
- [Run BGP in networks with more than one BGP router](#)
- [Manipulate BGP AS numbers or AS paths](#) with nerd knobs like **as-override** and **local-as**

Recommended: Ubuntu, KVM, libvirt, Docker



Prerequisite software

- Python3
- Ansible (to configure the devices)

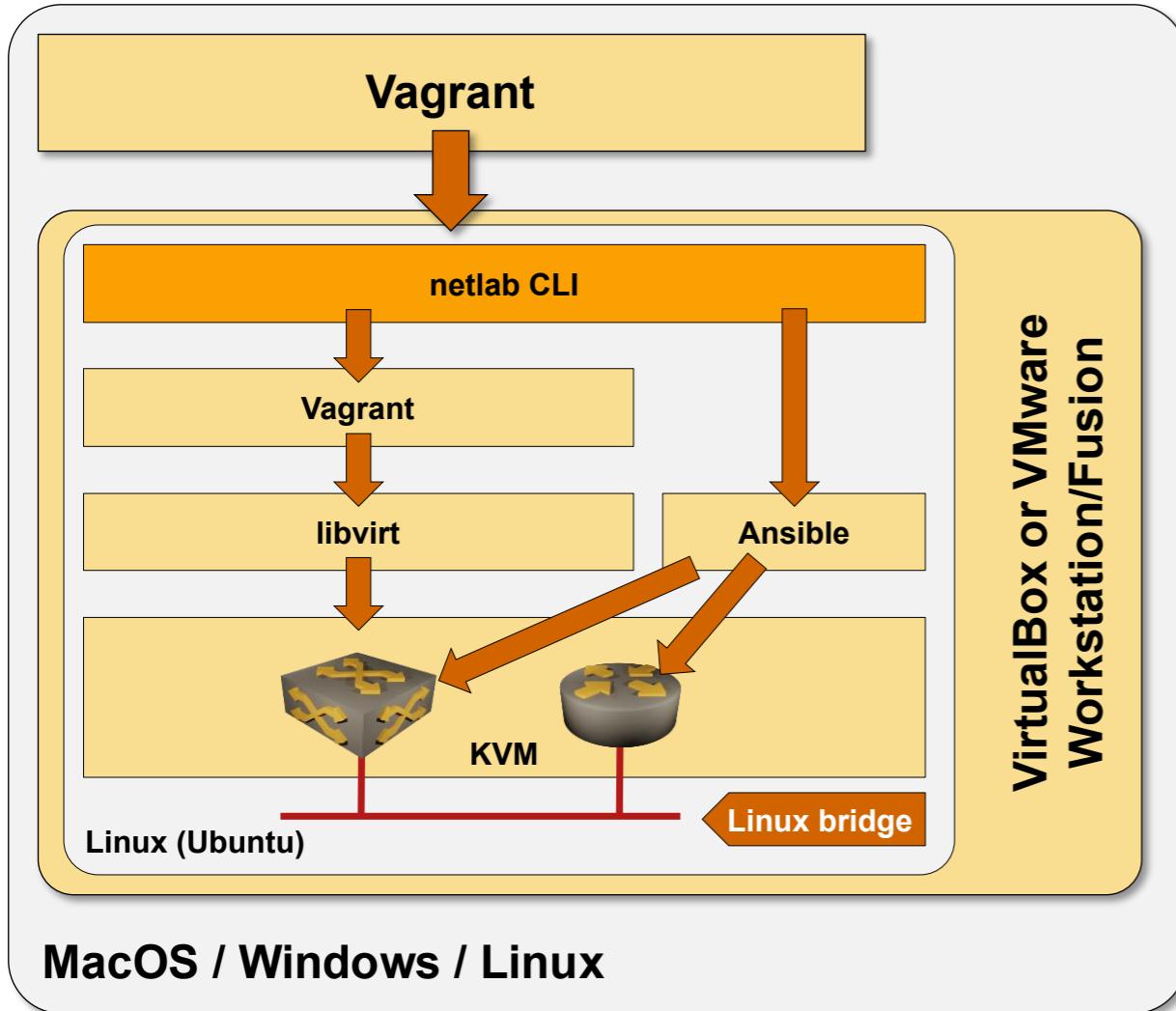
Devices as virtual machines

- KVM
- libvirt
- Vagrant with vagrant-libvirt plugin

Containers

- Docker
- Containerlab

Use Existing x86 Device: Ubuntu VM



Requirements

- You can run containers with any VM virtualization product
- Nested virtualization is required to run network device VMs

Virtualization solution with nested virtualization

- Hyper-V (WSL)
- KVM
- VirtualBox
- VMware Workstation/Fusion

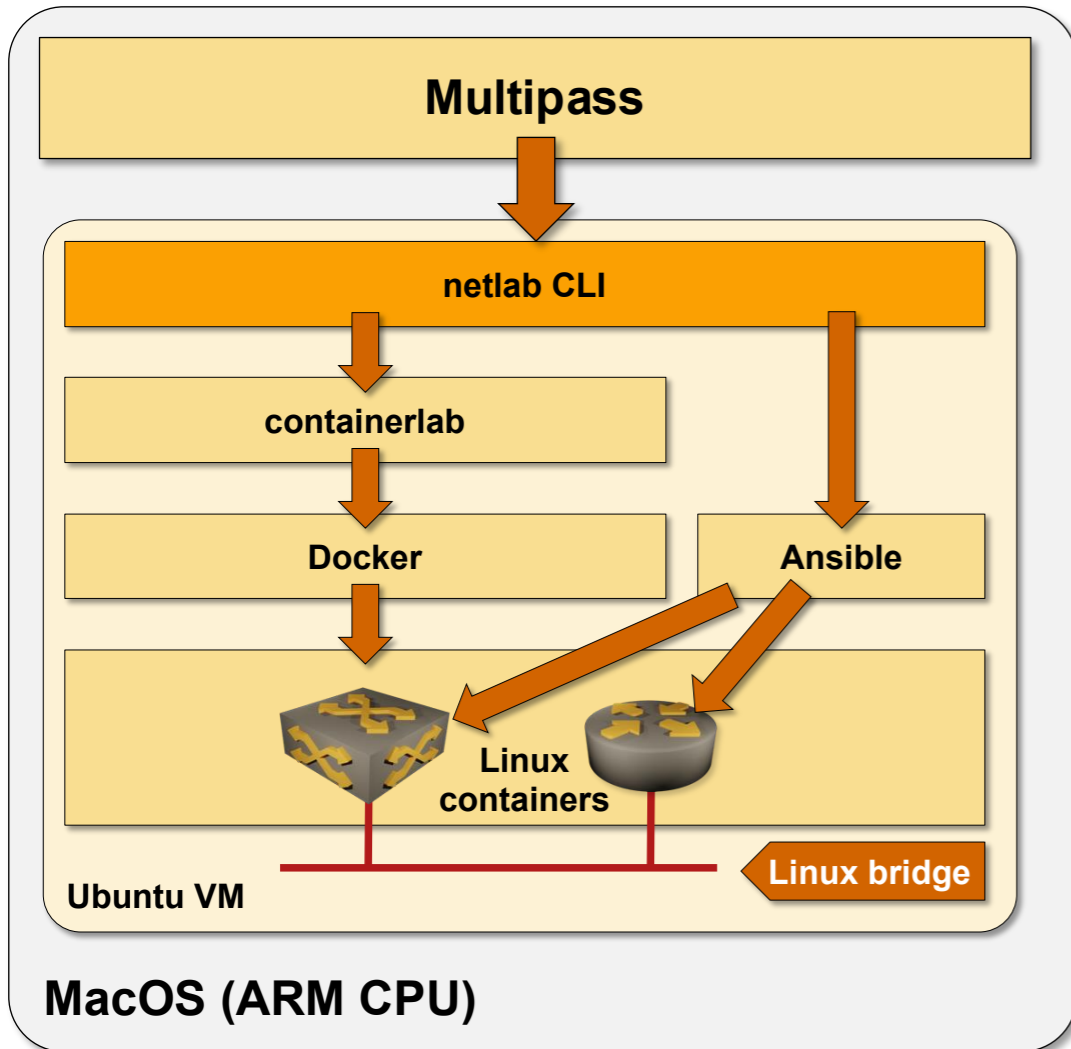
Optional

- Start the VM with Vagrant (simplifies the operations)

Alternative

- Cloud deployment

Ubuntu VM on Apple Silicon



- **multipass** starts an Ubuntu VM on an ARM CPU
- Nested virtualization is not supported → containers only
- Container images must be built for the ARM CPU → Arista EOS, FRRouting, and SR Linux

Interesting use cases

- Run BGP, VXLAN, or EVPN labs on your Apple laptop
- FRRouting and Arista configuration syntax are pretty close to the *industry standard CLI*



Some Assembly Required (Thank You, Vendors)

Automatically downloadable images and containers

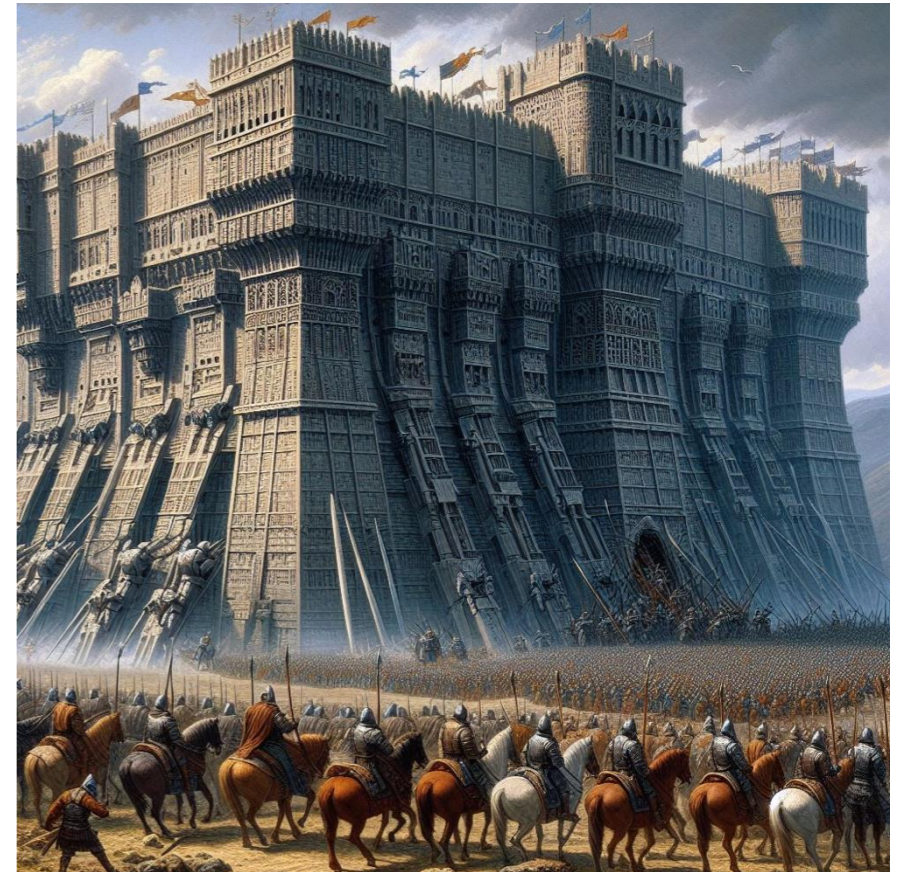
- FRRouting
- Linux, OpenBSD
- Nokia SR Linux
- VyOS

Easy to download (no registration required)

- Juniper (some images), Dell OS10, Mikrotik RouterOS7

Most everything else (from bad to worse)

- Registration (Arista EOS, Aruba CX, Cisco Nexus OS, Cumulus)
- Download tied to a valid support contract (Cisco CSR)
- Begging your SE
- Available only if you know the right dev person



But Wait, That's Not All

After you download a container image (Arista cEOS)

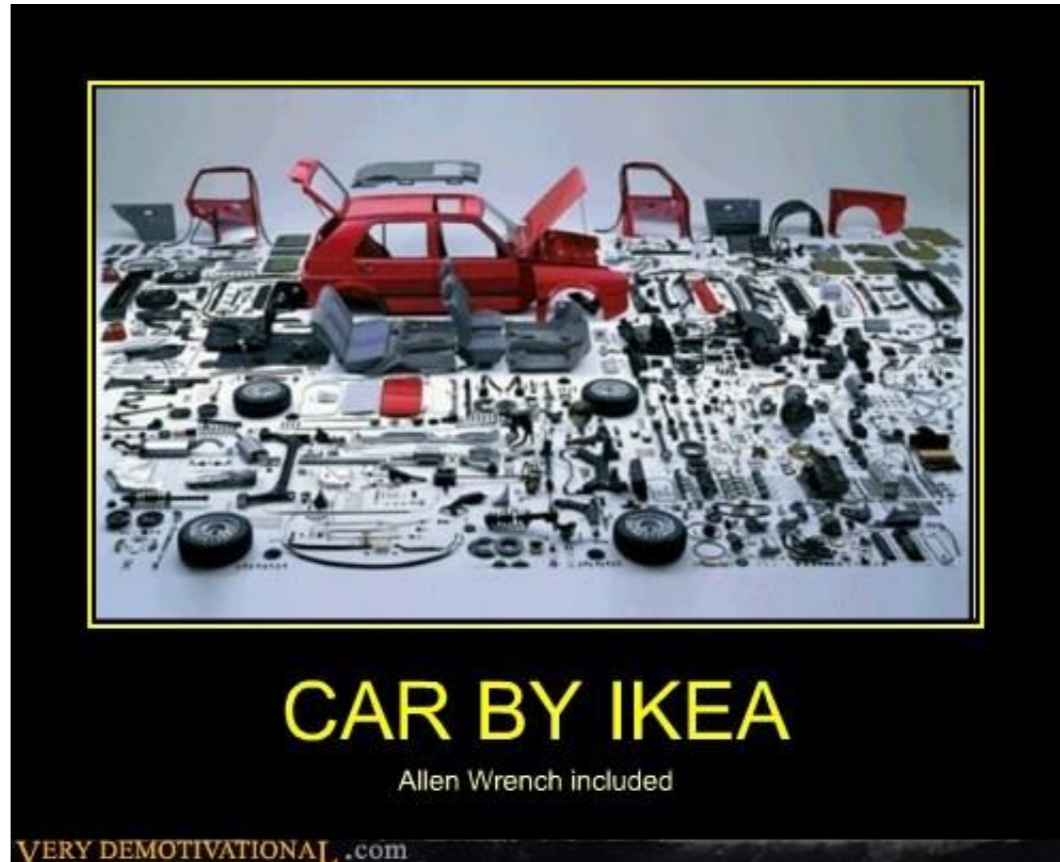
- Unpack and install it (easy)

Virtual machines are a nightmare

- Download a virtual disk
- It boots without a configuration and expects stuff on serial port
- Exception: Junos and ASA can take configuration from a mounted CD-ROM

Building a Vagrant box

- What we need to automate lab startup
- Start the VM, answer a dozen questions
- Copy-paste initial configuration
- Save the configuration, shut down the VM
- Package as a Vagrant box, hope it works



Getting Started

Getting Started: Hardware

Get decent hardware

- Network devices consume between 256MB (Mikrotik) and 12GB (Cisco Nexus 9300v release 10.3) per node
- Containers consume between 200MB (FRR) and 1GB (Arista cEOS, Nokia SR Linux)

Decide which deployment architecture you want to use

- Run x86 containers in GitHub Codespaces
- Existing x86 hardware using Windows/Mac: Linux VM (might need nested virtualization support)
- ARM CPU: Linux VM running FRR or SR Linux containers
- Standalone Linux server on dedicated hardware: use a fresh copy of Ubuntu

Standalone server recommendations

- Intel NUC works great (but is no longer available)
- Run containers (Arista, Cumulus, FRRouting) on a VM in the cloud (example: Oracle Cloud Free Tier)
- Some cloud providers offer nested virtualization (Google Cloud, Packet, Digital Ocean)

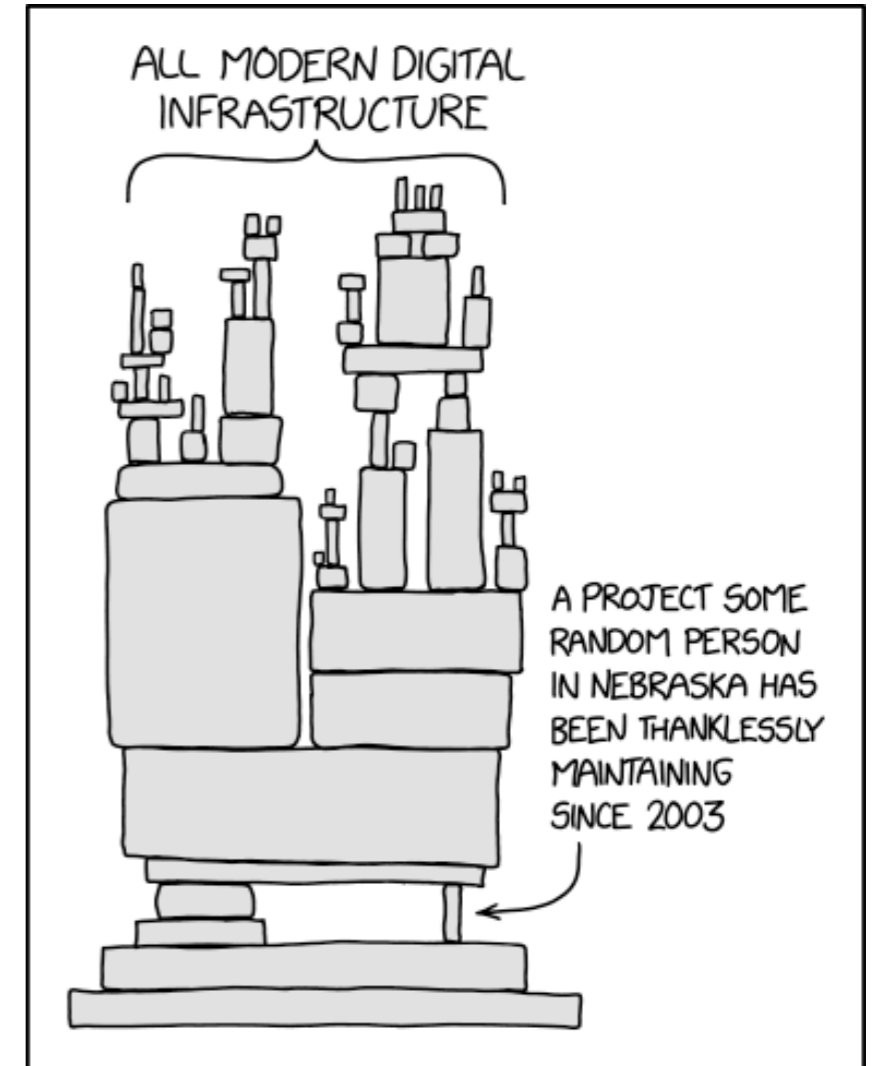
Getting Started: Installation

Ubuntu

- Start with a (fresh) Ubuntu VM or a bare-metal server
- Install Python3 if needed
- Install *networklab* package with **pip**
- Use **netlab** installation scripts to install libvirt/KVM, Vagrant, Ansible, Docker, containerlab...
- Download or build Vagrant boxes or containers (tons of recipes on netlab.tools/labs/libvirt and netlab.tools/labs/clab)

Other Linux distributions

- Everything *should* work (apart from installation scripts)
- Box-building tools have been tested on Ubuntu
- We won't be able to help you 🙄♂





ANOTHER TOOL...

... ANOTHER CONFIG FILE



UNICORN

AS A

SERVICE

No Presentation Is Complete Without Mentioning AI/ChatGPT

“ netlab is a lab creation tool documented at <https://netlab.tools/> with additional examples provided at <https://blog.ipSpace.net/tag/netlab/> and sample lab topologies at <https://github.com/ipSpace/netlab-examples>.

Can you use that documentation to create a netlab topology file for a network with two routers (R1 and R2) connected with two parallel links and running OSPF. Each router should also have an extra LAN interface.

- Results suck unless you mention the URLs in the prompt
- The initial result usually doesn't work. Run **netlab** and copy-paste error messages to ChatGPT
- Like any persistent sloppy clueless intern, it will eventually get it right
- It took ChatGPT four retries (in May 2025) to get a working 2-router-with-OSPF lab topology

Back to Reality

Is It Worth It? What Others Are Saying...



Lou D. • 2nd

Network Security Engineer | Network Automation

4h (edited) ...

I fully agree with Ivan's point on the show about being exhausted building labs . Netlab is really one the best tools to help me build fairly complex network designs quick and also cheap ! I don't need so much resource provisioning to get something started a tiny aws Ubuntu box will build an decent amount of nodes .

Like ·  2 | Reply · 1 Reply



Ethan Banks • 1st

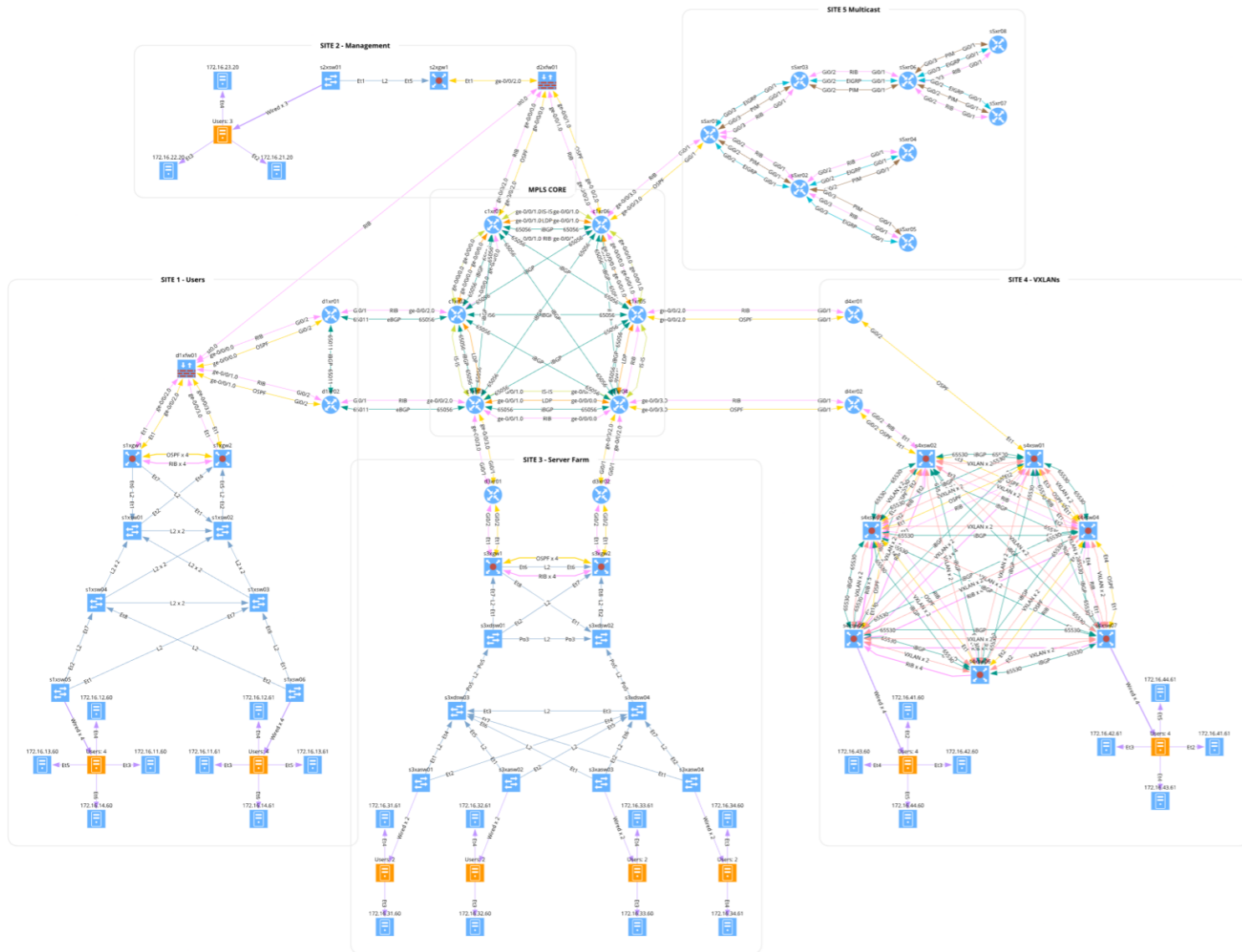
IT Podcast Host & Industry Analyst

1mo · 

...

If you work with networking labs but haven't worked with the netlab project, you need to. YAML-defined labbing, essentially. So much of the tedious work is just done for you. You don't have to drag icons around on a canvas and make links and setup IP addressing and configure routing and and and. Just tell netlab what you want it to build in the YAML file. Then "netlab up".

What Others Are Building with netlab



How Can You Help?

- Spread the word ;)
- Use the tool to build your labs
- Ask questions and report bugs

Want to contribute?

- Fix documentation
- Fix bugs
- Add new functionality to existing devices
(example: VXLAN on IOS XR, EVPN on Nexus OS)
- Add new devices

Still not enough?

- Develop new plugins (hint: OSPF areas)
- Develop new modules (IP multicast, Babel...)



Questions?

Documentation: [netlab.tools](#)
Blog posts: blog.ipSpace.net/tag/netlab.html
Source code: github.com/ipSpace/netlab
Examples: github.com/ipSpace/netlab-examples
Sample project: bgplabs.net (BGP labs)
isis.bgplabs.net (IS-IS labs)
evpn.bgplabs.net (EVPN/VXLAN labs)

To reach me

Web: ipSpace.net
Email: ip@ipSpace.net

