The bridge to possible

Jupyter Notebooks + XR for Modern Learning Process

Taran Deshpande, Technical Marketing Engineer

Cisco Webex App

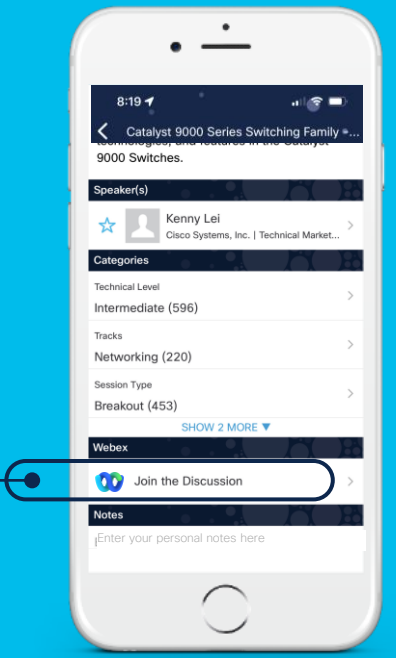
Questions?

Use Cisco Webex App to chat
with the speaker after the session

How

1. Find this session in the Cisco Live Mobile App
2. Click “Join the Discussion”
3. Install the Webex App or go directly to the Webex space
4. Enter messages/questions in the Webex space

Webex spaces will be moderated until February 24, 2023.



Agenda

- Introduction
- 8000 Emulator
- PyVXR Orchestration Tool
- Dcloud Offering
- Notebook Demo
- Conclusion

Cisco IOS XR – Industry's #1 Network OS

Cisco NCS 5xx

Cisco NCS 5xxx

Cisco ASR9000

Cisco 8000

Merchant Silicon

Custom Silicon

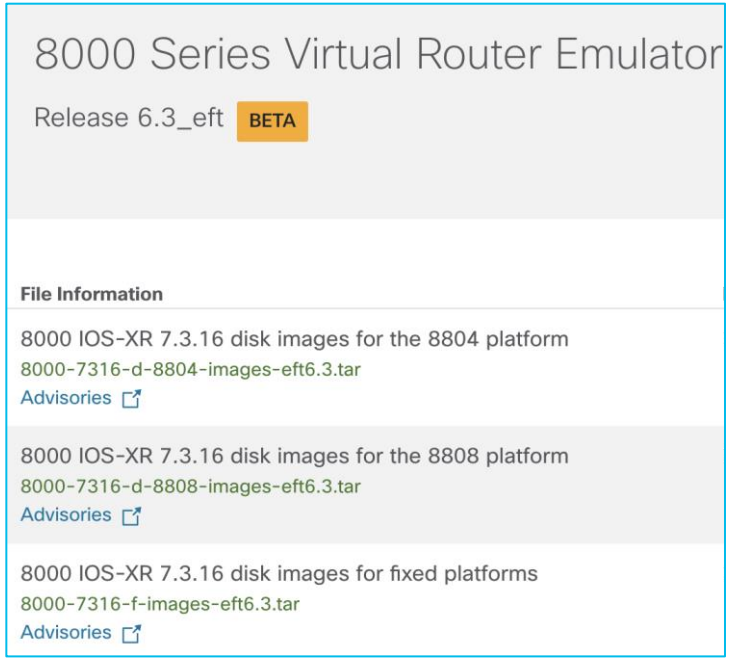
Session Topic

Legacy Platforms – GSR, CRS, NCS6K

Flexible Platforms – Virtual & more...

8000 Emulator

- Software simulation of Cisco 8000 available on Cisco Software Central for free with 8000 potential order;
- 8000 Simulator can emulate fixed or distributed devices;
- With current supply chain issues emulator can get you up to speed quicker with new platform.



PyVXR

- A package containing collection of tools to launch and manage virtual topologies
- Topologies can consist of generic and custom Cisco virtual routers
- How to invoke
  - Use PyVXR API within a Python Environment
  - PyVXR CLI tool
- Available to prospective Cisco 8000 customers

```
from pyvxr.vxr import Vxr
import logging
logging.basicConfig(level=logging.INFO)

cfg={
    'simulation': {
       'skip_auto_bringup' : True
    },

    'devices': {
      'router0': {
        'platform': 'spitfire_f' ,
        'image': '/auto/vxr/images/spitfire/8000-x64.iso',
     } }

vxr_sim=Vxr()
vxr_sim.no_image_copy=True

try:
    vxr_sim.start(cfg)
    ports=vxr_sim.ports()
except:
    print("Sim launch failed!")
    #get vxr log files for debugging
    vxr_sim.logs()
finally:
    vxr_sim.clean()
```

Topology defined in YAML

```
simulation:
  no_image_copy: True
  rollback_xr_user: True

devices:
  R0:
    platform: spitfire_f-baked
    linecard_types: ['8201-sys']
    image: /opt/cisco/images/8000/xr7/7.3.16/8000-x64-7.3.16.qcow2
    xr_config: |
       netconf-yang agent ssh
    xr_port_redir: [22, 161, 830]
...
connections:
    hubs:
       R0R1hub:
          - R0.FourHundredGigE0/0/0/0
          - R1.FourHundredGigE0/0/0/0
```

Specify LC type

QCOW image from 8000-7316-f-*** tar

Config applied on bring up.

Redirected ports

Connections between virtual devices

$ vxr.py start <config_yaml_file>

8000 Emulator In Action through PyVXR

Emulator Requirements

Emulator	Operating System	CPU	Memory	Min Memory	Disk
8201 (fixed)	IOS-XR7	4	20-32G	12G	30G+
8202 (fixed)	IOS-XR7	4	20-32G	12G	30G+
8804/8808 (modular)	IOS-XR7	8 (RP+LC)	64G	40G	30G+

System Requirements

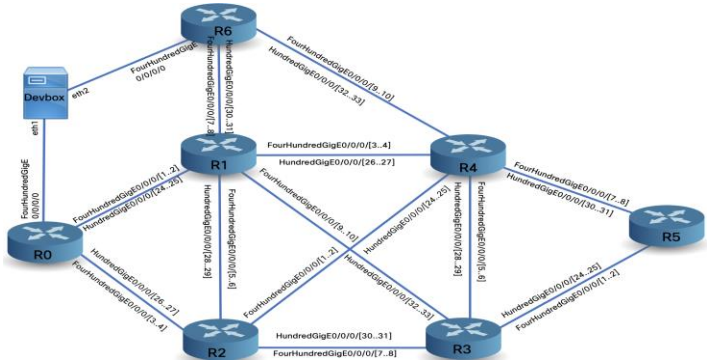
System	Type	Minimal System	Operating System
Dedicated Server	Bare Metal Server	16+ cores 64G+ Mem	Ubuntu18.04
AWS	Bare Metal Instance	Bare metal Instance M5d.metal	Ubuntu18.04
Azure	Virtual Machine	16+ cores 64G+ Mem	Ubuntu18.04
Google Cloud	Virtual Machine	16+ cores 64G+ Mem	Ubuntu18.04
ESXI	Virtual Private Cloud	16+ cores 64G+ Mem	Ubuntu18.04

dCloud Session

Installation and Hosting provided by dCloud

Bring your own use case with fixed topology

- PyVXR is a Python module that allows you to build a topology of virtual devices which are spun up as virtual



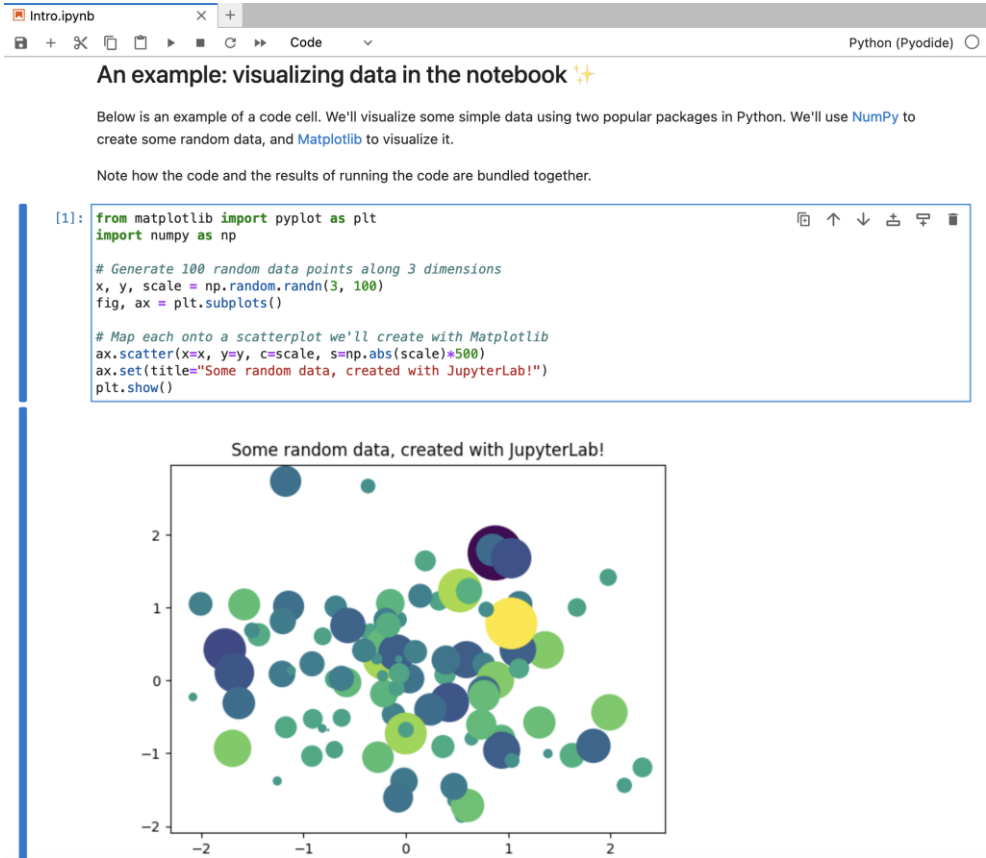
Curated experience with Jupyter Notebooks

- Set of validated notebooks from Documentation team with contributors from engineering/TME
- Available on GitHub: https://github.com/ios-xr/network-notebooks

Introducing Jupyter Notebooks

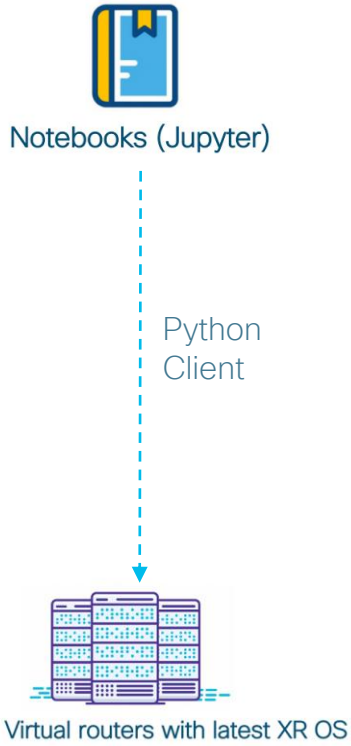
The Jupyter Notebook

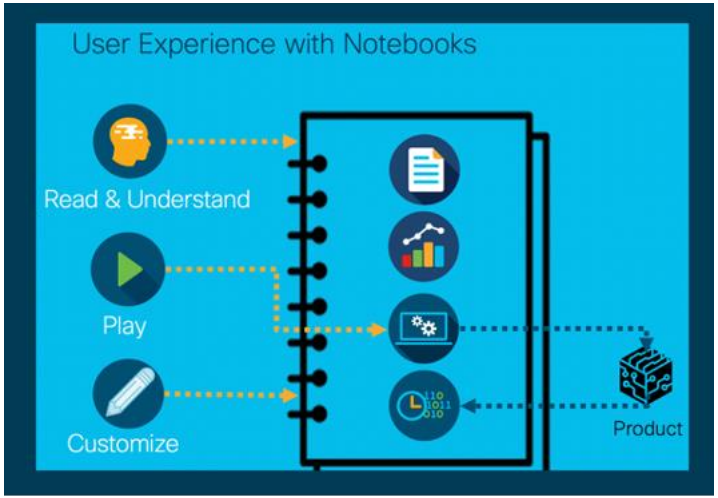
- Interactive authoring of documents combining text, mathematics, computations and rich media output
- Named after 3 core languages- Julia, Python, and R
- Components
  - Notebook Web Application
  - Kernels
  - Notebook Documents
- https://jupyter.org/



cisco Live!

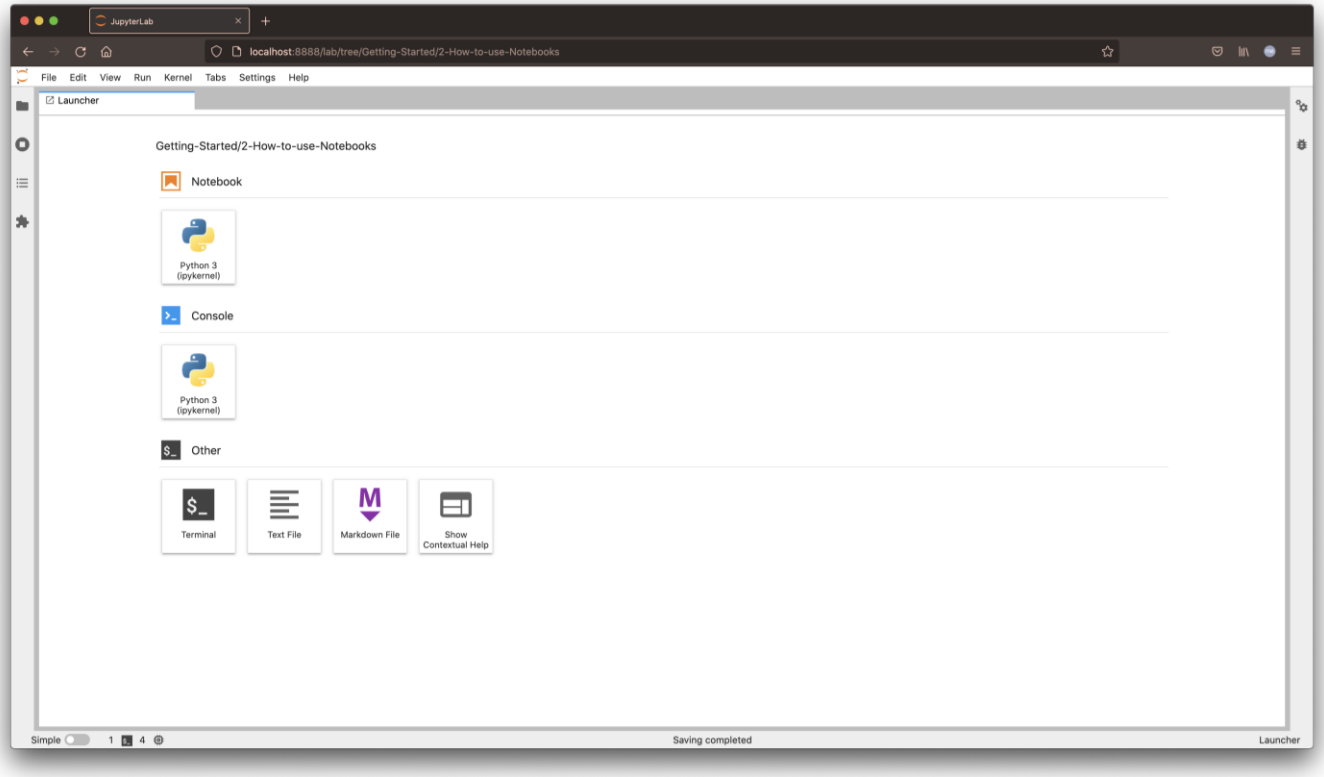
Interaction with Simulator – Networks within Notebooks





Web UI for Notebooks

- Not much different from Office document
- Create new, execute notebook, run scripts, etc.
- Typically available on localhost:8889



JupyterLab

localhost:8888/lab/tree/Getting-Started/2-How-to-use-Notebooks/Notebook-magic.ipynb

File Edit View Run Kernel Tabs Settings Help

Notebook-magic.ipynb

Markdown

Toolbar

Python 3 (ipykernel)

Kernel

Let's see how this actually happens. In this notebook, we will be using a 2 router back-to-back network topology, like below.

TOPOLOGY

Images

> In the cell below, the python module **b2b_routers** contains the code to set up the python enviroment, the simulated 2-router topology as depicted in the topology diagram along with the base configuration of the routers. The variable **sim** is initiailized to the CiscoVxr module. The **sim.init** and **sim.up** API calls initialize the 2 routers and brings up the network. For more informaiton on the router configuration and topology, refer to the **b2b_routers.py** file in this folder.

> Play the below cell to bring up the network. Please note that the bring up can be slow, sometimes taking 10 minutes or more.

[ ]:
```
from b2b_routers import *
sim = CiscoVxr()
sim.clean()
cfg = {'devices':
       {'r0': {'platform':'spitfire_f'},
       }
     }
print(cfg)
sim.init(cfg)
sim.up()
```

Code

> At this point, console access to the routers is available. Play the below cell and obtain the telnet and ssh connection details of the routers. You can access the simulated router consoles directly from your laptop through telnet/ssh or you can just continue playing the subsequent code-cells of this notebook and watch the notebook access the telnet console and execute commands on the router.

[ ]:
```
print('Consoles can be reached by:')
print('PE1:', get_telnet_cmd(sim, 'pe1'), '\nPE2:', get_telnet_cmd(sim, 'pe2'))
print('or better:')
print('PE1:', get_ssh_cmd(sim, 'pe1'), '\nPE2:', get_ssh_cmd(sim, 'pe2'))
print('The password is cisco123')
```

> When the below cell is played the notebook accesses the telnet consoles of the routers.

Simple 1 4 Python 3 (ipykernel) | Idle Saving completed Mode: Command Ln 1, Col 1 Notebook-magic.ipynb

Demo 2: Jupyter Notebook for GRE Tunnel Configuration

Existing Notebooks

Setup Network for Transport

- Simplify Your Network Using EVPN VPWS
- Segment Routing v6 Over An IS-IS Network

Secure Network

- Integrity Measurements To Handle Threats (IMA)
- Manage Your Router Database Securely (MongoDB)
- Type 6 Password Authentication For BGP

Traffic Management

- Configure Policies for Quality of Service
- Prioritize Delay-Sensitive Traffic Using QOS
- Configure QoS Using NSO
- Filter Network Traffic Using Access Control List
- Filter Network Traffic Using ACL Yang

Existing Notebooks (cont.)

Network Monitoring

- Traffic Monitoring With Encapsulated Remote Switched Port Analyzer (ERSPAN)
- Stream CPU Utilization Data Using Model-driven Telemetry
- Deploy YANG Data Model to Stream CPU Utilization Data Using Model-driven Telemetry
- Setup a Pipeline and Stream CPU Utilization Data Using Model-driven Telemetry

Service-Layer APIs

- Super-charge Your Router Performance With Service-layer APIs

Utilities

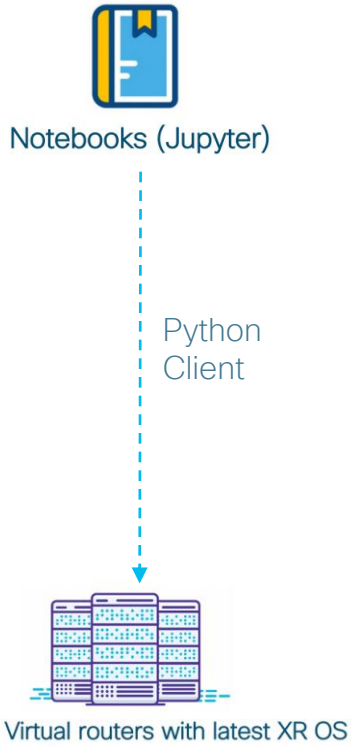
- Generate Real-time Network Traffic On Simulated networks
- Analyze Bit Level Information of Traffic Flows
- Utilize Linux Servers Effectively Within Simulated Networks

Other Virtualization Options

- XRv9000
- XRd
- Cisco Modeling Labs
- GNS3
- EVE-NG
- Kubernetes Network Emulator (KNE)

Key Takeaways

- Automate the creation and bring-up of the topology on the simulator
- Design and share proof-of-concepts, use cases, or features
- Use as “Live” Documentation
- Leverage it as a tutorial for educational purposes



Call to action

Contribute to our Network Notebooks Github

For access to PyVXR, contact your Cisco account team or myself

View the Emulator datasheet

Complete your Session Survey

- Please complete your session survey after each session. Your feedback is important.
- All surveys can be taken in the Cisco Events Mobile App or by logging in to the Session Catalog and clicking the "Attendee Dashboard" at https://www.ciscolive.com/emea/learn/sessions/session-catalog.html

Continue Your Education

Visit the Cisco Showcase for related demos.

Book your one-on-one Meet the Engineer meeting.

Attend any of the related sessions at the DevNet, Capture the Flag, and Walk-in Labs zones.

Visit the On-Demand Library for more sessions at ciscolive.com/on-demand.


cisco
The bridge to possible
Thank you
cisco Live!


CISCO Live!
ALL IN