
CISCO
You make possible
#CLUS

Cisco Webex Teams

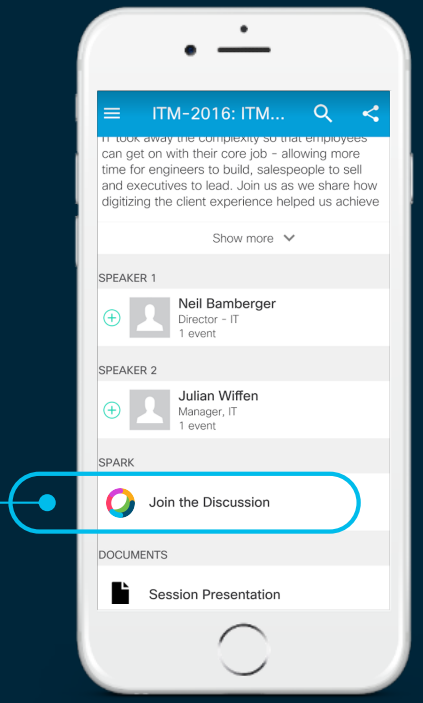
Questions?

Use Cisco Webex Teams 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 Webex Teams or go directly to the team space
4. Enter messages/questions in the team space

Webex Teams will be moderated by the speaker until June 16, 2019.



cs.co/ciscolivebot#DEVWKS-2594

Leveraging NX-API for Customized Operational Analytics

Dr Tim Miller, Virtual CSE DC/Cloud
@broadcaststorm

DEVWKS-2594

Cisco live!
June 9-13, 2019 • San Diego, CA
#CLUS

Agenda

- Introduction
- NX-API Overview
- Metrics and Monitoring
- Hands On
- Conclusion

The Remainder of our 45 Minutes

- Who You Are
  - Command Line Fighter Pilot
  - Know your network
  - Have your favorite set of metrics
- What You'll Learn
  - Translate CLI to Python programming
  - Plethora of monitoring tools
  - Parsing and graphing the metrics

Parsing CLI Output

Vintage style versus Hip style

```
Software
  BIOS: version 07.59
  NXOS: version 7.0(3)I7(3)
  BIOS compile time:  08/26/2016
  NXOS image file is: bootflash:///nxos.7.0.3.I7.3.bin
  NXOS compile time:  2/12/2018 13:00:00 [02/12/2018 19:13:48]


Hardware
  cisco Nexus9000 C9372PX chassis
  Intel(R) Core(TM) i3- CPU @ 2.50GHz with 16400992 kB of memory.
  Processor Board ID SAL18516SA8

  Device name: spine-1
  bootflash:   51496280 kB
Kernel uptime is 0 day(s), 0 hour(s), 5 minute(s), 17 second(s)
```


```
{
   "bios_ver_str": "07.59",
   "kickstart_ver_str": "7.0(3)I7(3)",
   "bios_cmpl_time": "08/26/2016",
   "kick_file_name": "bootflash:///nxos.7.0.3.I7.3.bin",
   "kick_cmpl_time": "2/12/2018 13:00:00",
   "kick_tmstmp": "02/12/2018 19:13:48",
   "chassis_id": "Nexus9000 C9372PX chassis",
   "cpu_name": "Intel(R) Core(TM) i3- CPU @ 2.50GHz",
   "memory": "16400992",
   "mem_type": "kB",
   "proc_board_id": "SAL18516SA8",
   "host_name": "spine-1",
   "bootflash_size": "51496280",
   "kern_uptm_days": "0",
   "kern_uptm_hrs": "0",
   "kern_uptm_mins": "5",
   "kern_uptm_secs": "31",
}
```

Parsing CLI Output

Vintage style versus Hip style

```
(server) $ ssh admin@switch "show version" > output.txt
Password:
(server) $ awk '/BIOS:/ { print $3; }' output.txt
07.59
```

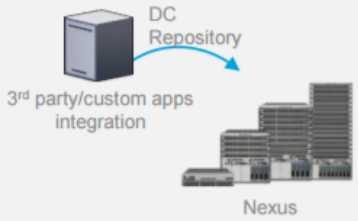
Vintage

Hip

```
print(output["bios_ver_str"])
```

Open NX-OS Provides

3rd Party Linux Applications

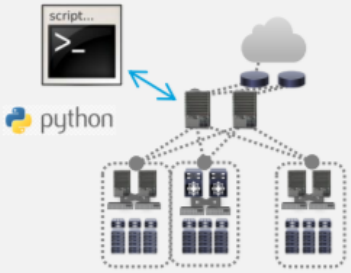


Use 3rd party applications using secure LXC containers

Linux Tools

Leverage Linux commands for config and troubleshooting

Programmable Open APIs



Object-based, model driven APIs (RESTful XML/JSON)

3rd Party DevOps Automation Tools

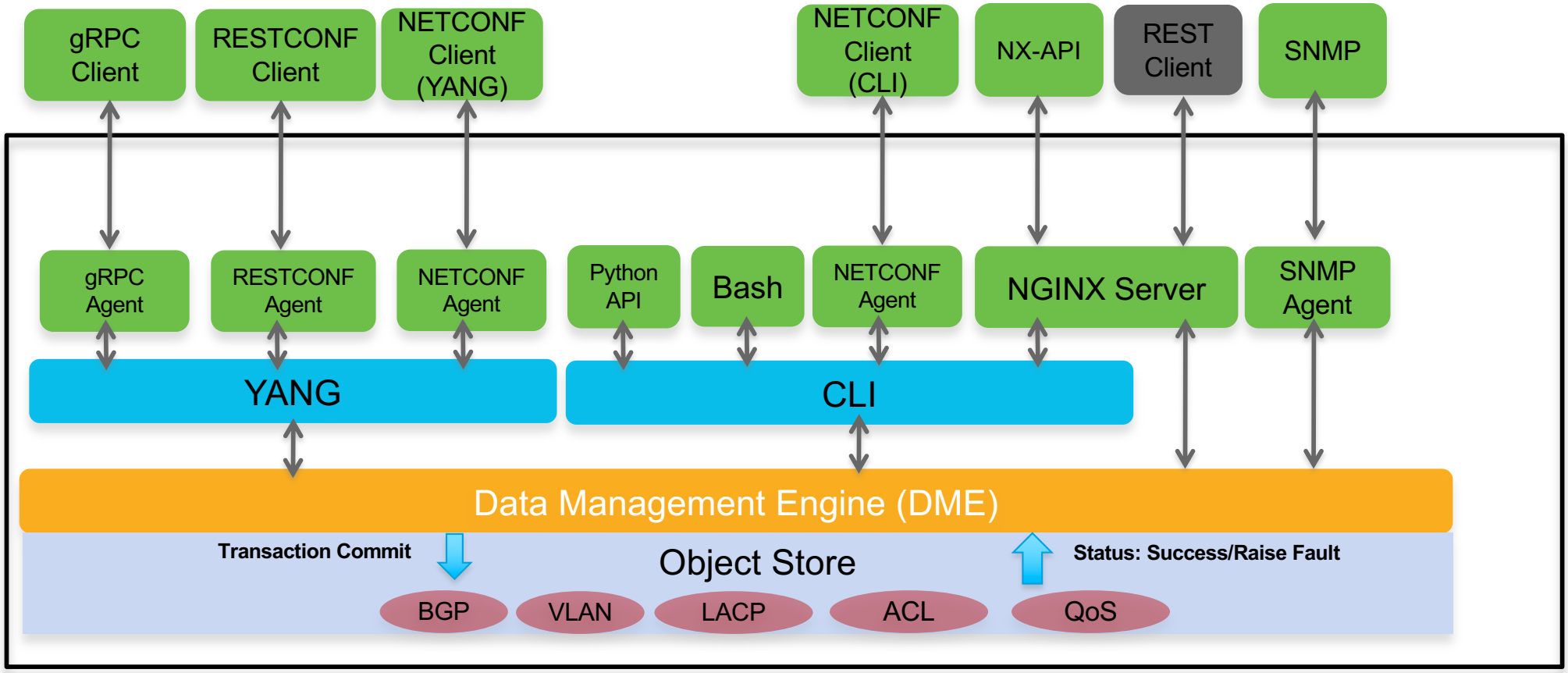
Leverage same software tools and expertise across different IT departments

NX-API Overview

You make networking **possible**

Cisco live!

NX-OS Software Architecture

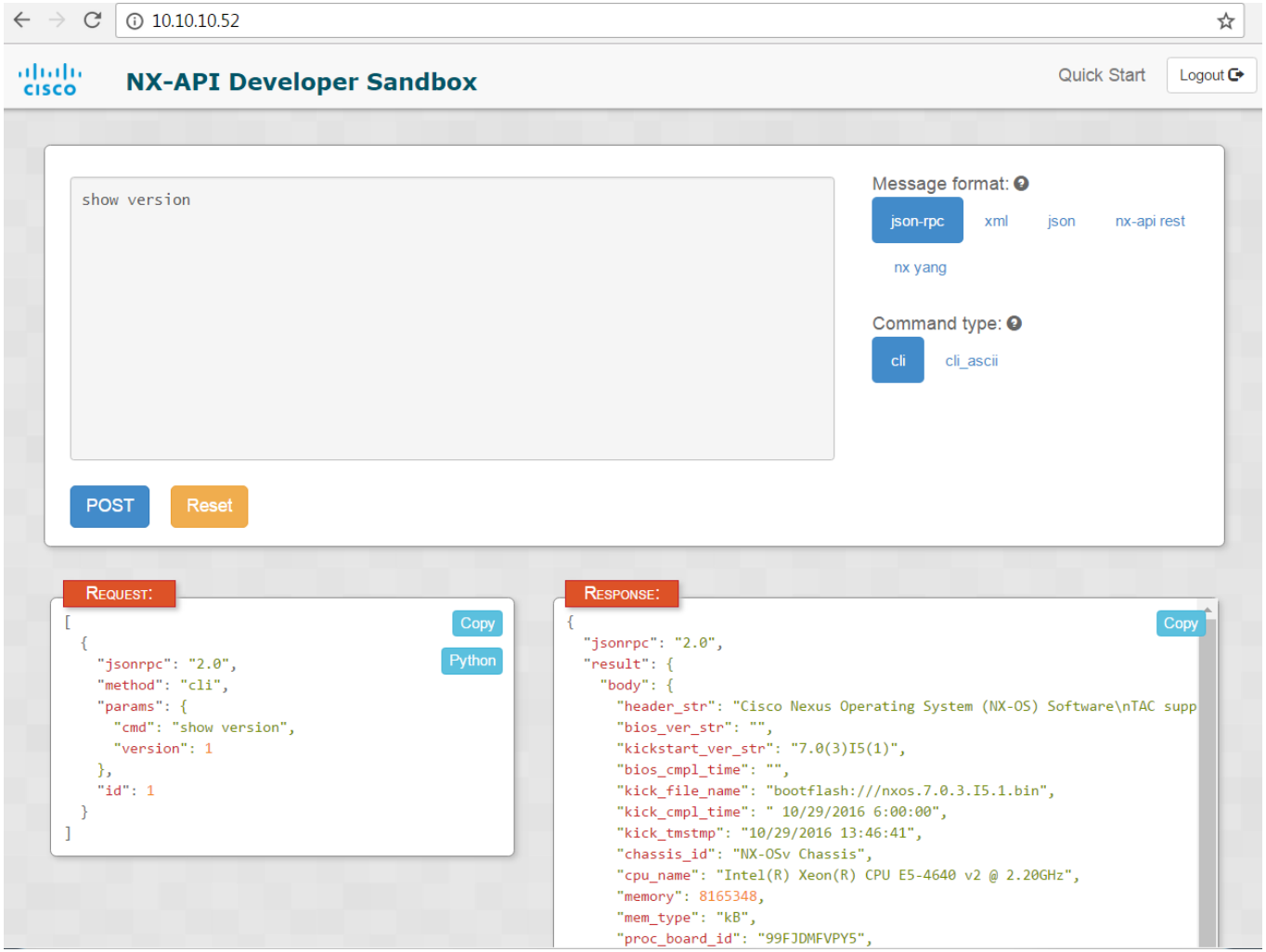


NX-API CLI Example

```
{
    "jsonrpc": "2.0",
    "method": "cli",
    "params": {
      "cmd": "show ver
      "version": 1
    },
    "id": 1
}
```

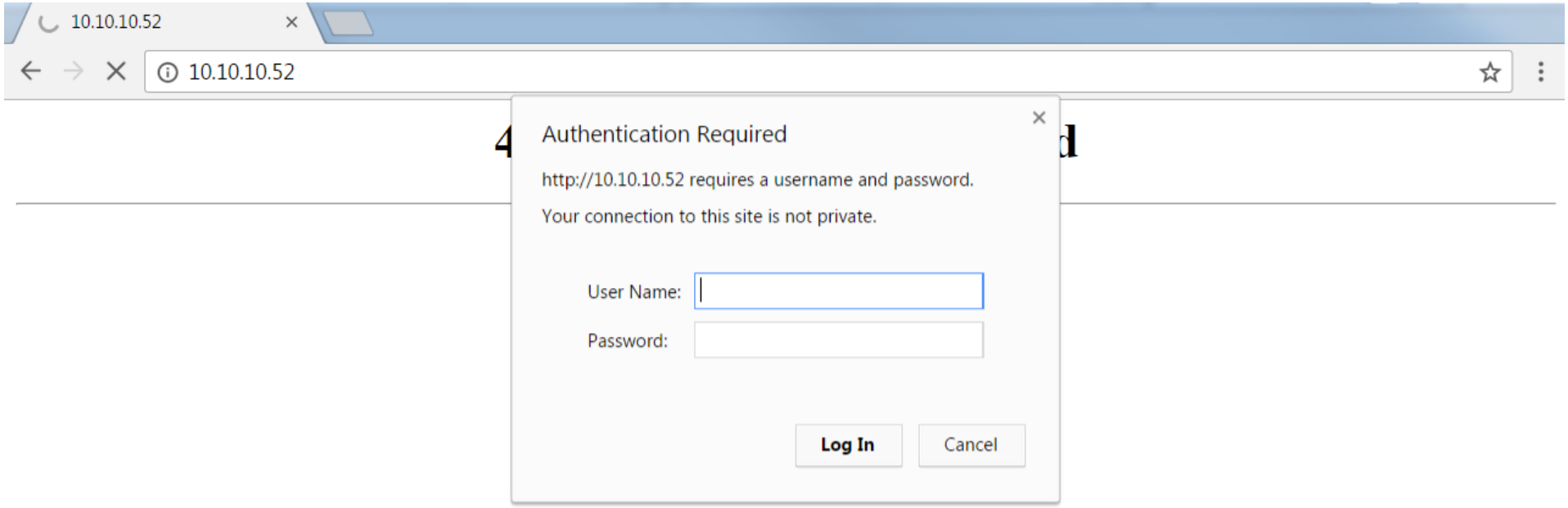
```
{
  "bios_ver_str": "07.59",
  "kickstart_ver_str": "7.0(3)I7(3)",
  "bios_cmpl_time": "08/26/2016",
  "kick_file_name": "bootflash:///nxos.7.0.3.I7.3.bin",
  "kick_cmpl_time": "2/12/2018 13:00:00",
  "kick_tmstmp": "02/12/2018 19:13:48",
  "chassis_id": "Nexus9000 C9372PX chassis",
  "cpu_name": "Intel(R) Core(TM) i3- CPU @ 2.50GHz",
  "memory": "16400992",
  "mem_type": "kB",
  "proc_board_id": "SAL18516SA8",
  "host_name": "spine-1",
  "bootflash_size": "51496280",
  "kern_uptm_days": "0",
  "kern_uptm_hrs": "0",
  "kern_uptm_mins": "5",
  "kern_uptm_secs": "31",
}
```

```
{ "jsonrpc": "2.0",
  "result": {
    "body": {
      ...
      "bios_ver_str": "07.59",
      "kickstart_ver_str": "7.0(3)I7(3)",
      "bios_cmpl_time": "08/26/2016",
```



NX-API CLI is Secure

- Users must have the correct device role to use NX-API CLI.
- For example, a read-only role will not be able to make changes using NX-API CLI.



NX-API CLI Has Many Use Cases

- Check versions of multiple switches in one command
- VLAN provisioning
- Poll routing table to watch for flapping routes
- Poll MAC address table for end point tracking
- Collect LLDP/CDP data to build wiring maps
- Couple collection of structured output with database backend for more advanced applications

NX-API CLI Demo!

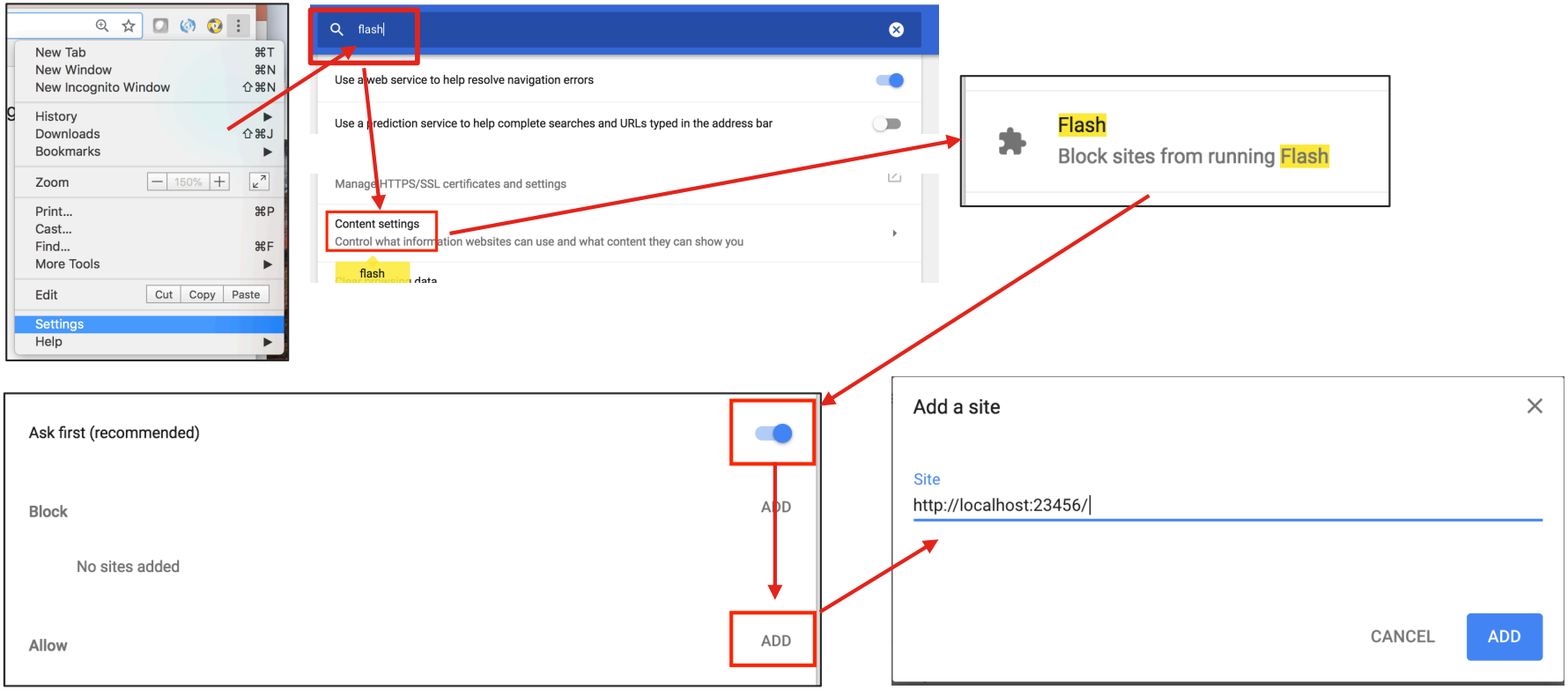
You make networking **possible**

Cisco live!

Connecting to Local N9KV Developer Sandbox

- Developer Sandbox is running on http://localhost:23456/
  - Username/Password is **admin/admin**
- May have to permit Flash in Chrome (next slide)
  - When connecting to the Developer Sandbox URL above, you'll get a warning that Flash is needed
  - OR… the "Python" button in the lower left request box does not produce Python text

Allowing Chrome to Use Flash for URL



Metrics

You make the power of data **possible**

Ciscolive!

Anatomy of Metrics Analytics

- Service to monitor
- Metric Generation
- Metric Collection
- Metric Storage
- Metric Visualization
- Browser to view it all

Metrics Generation

- Time Series Data
  - We are collecting measurements at regularly points in time
  - Name of Metric, Time Stamp, Metric Value, Metric Labels (units, source)
- Different Types of Metrics
  - Gauges
  - Counters
  - Timers (StatsD)
  - Histogram (Prometheus)
- Generators can be kernel level data, real time measurements, or calculated values

Prometheus

- Collection occurs via pull model over HTTP
  - Pushgateway exists for short-lived services or batch jobs
- Supported for service discovery (DNS, K8s, etc.)
- Data model for multi-dimension storage of time series data
  - Metric name, key/value pairs
  - Designed with microservices in mind

- https://prometheus.io/docs/introduction/overview/

One Python Script to Rule Them All

You make the power of data **possible**

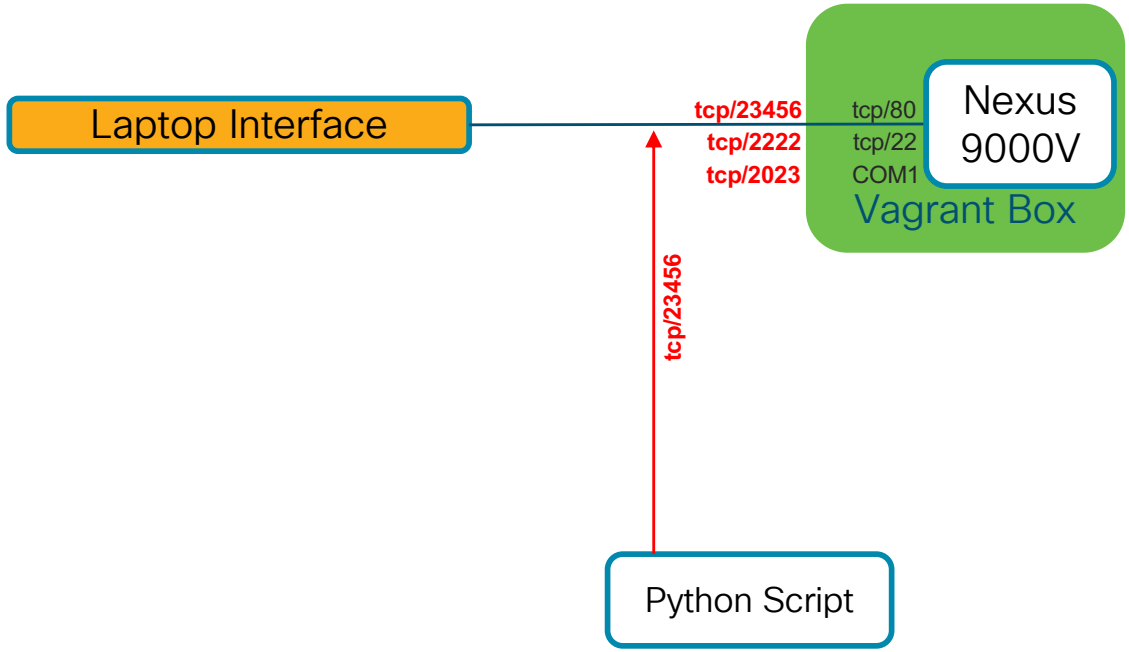
Cisco live!

Stop! Python Time...

You make multi-cloud **possible**

Ciscolive!

Metric Collection Service Connectivity Diagram

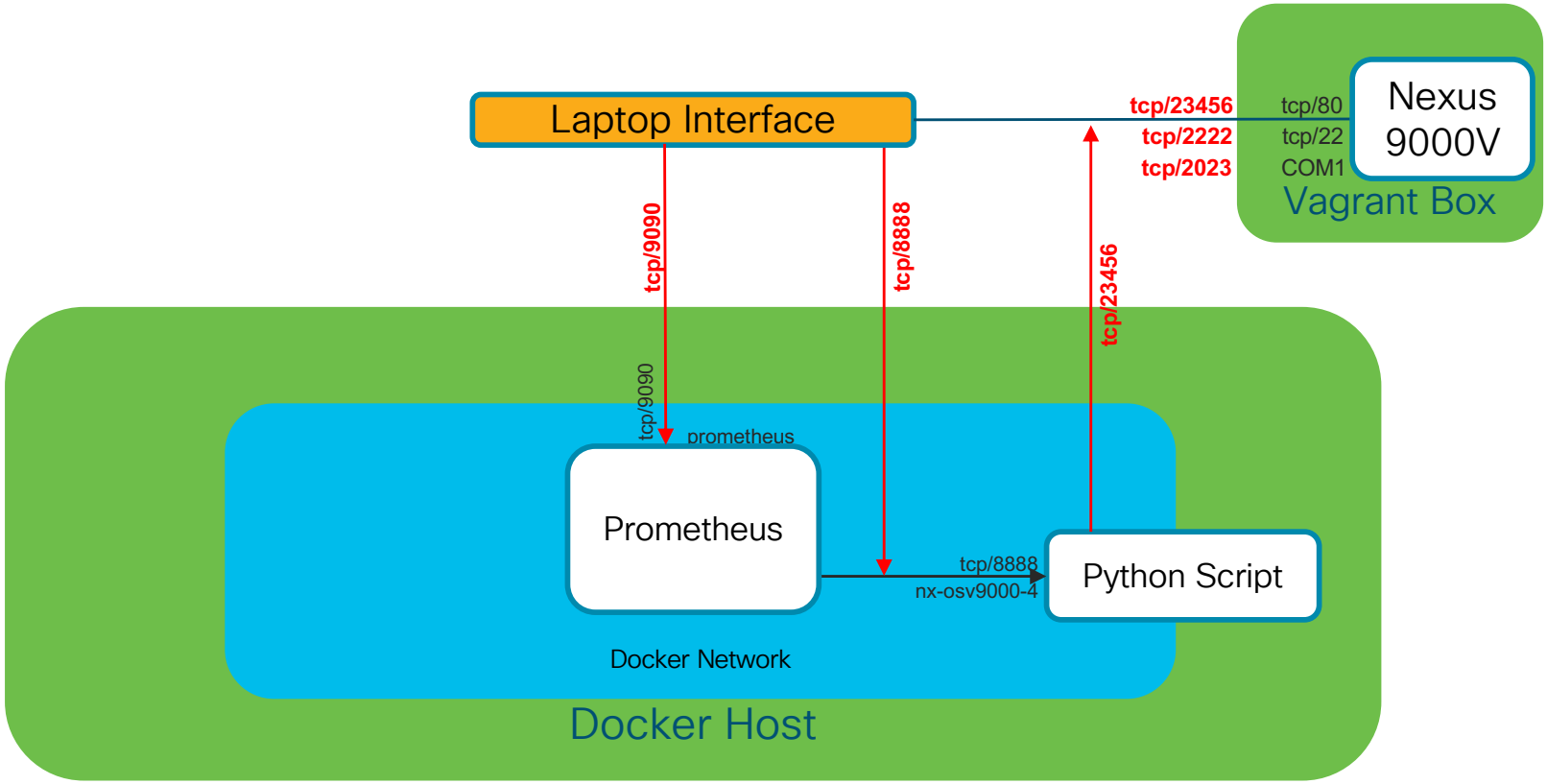


Commands found in README.md

Step-01 – Verify and Test Metric Generation

- Done for you:
  - Vagrant Box Nexus 9000v started
  - NXAPI and iCAM NX-OS features enabled
  - Workshop Git repo checked out to workspace folder
  - Python 3 modules populated
- Run script
  - `source ${HOME}/workspace/python/venv3/bin/activate`
  - `cd ${HOME}/workspace/DEVWKS-2594-CLUS19/nxapi_cli/step-01`
  - `python generate_l2table.py`

Metric Collection Service Connectivity Diagram



Commands found in README.md

Step-02 – Collect metric in Prometheus

- Build Docker image of Collector script

```
docker build -t devwks-2594/publish_l2table:latest -t devwks-2594/publish_l2table:1 .
```

- Create Docker network

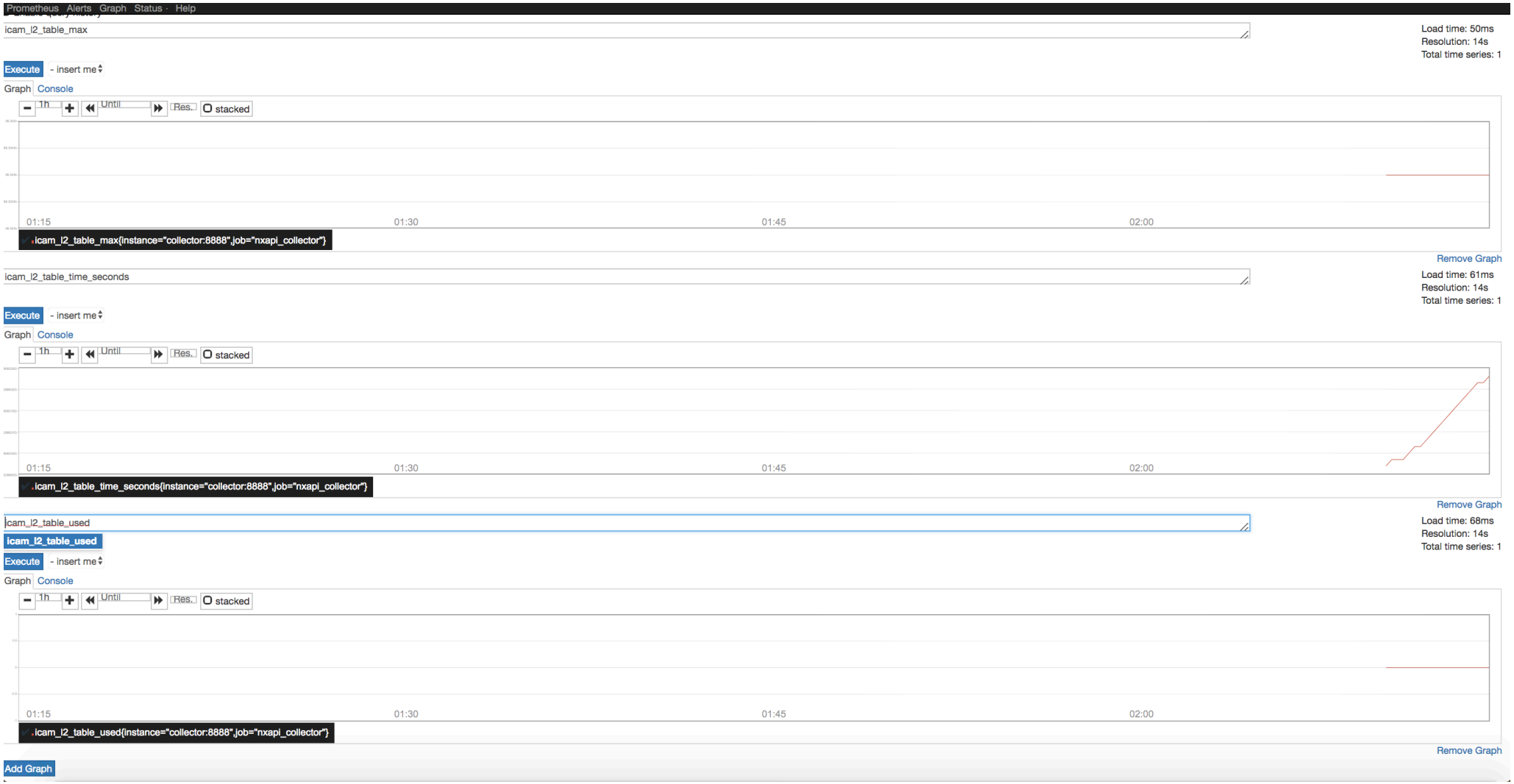
```
docker network create --driver=bridge --subnet=192.168.254.0/24 \
                      --gateway=192.168.254.254 --attachable demo0
```

- Deploy Prometheus container

```
docker run --name prometheus -d --network demo0 -p 127.0.0.1:9090:9090 \
           -v ${PWD}/prometheus.yml:/etc/prometheus/prometheus.yml \
           quay.io/prometheus/prometheus
```

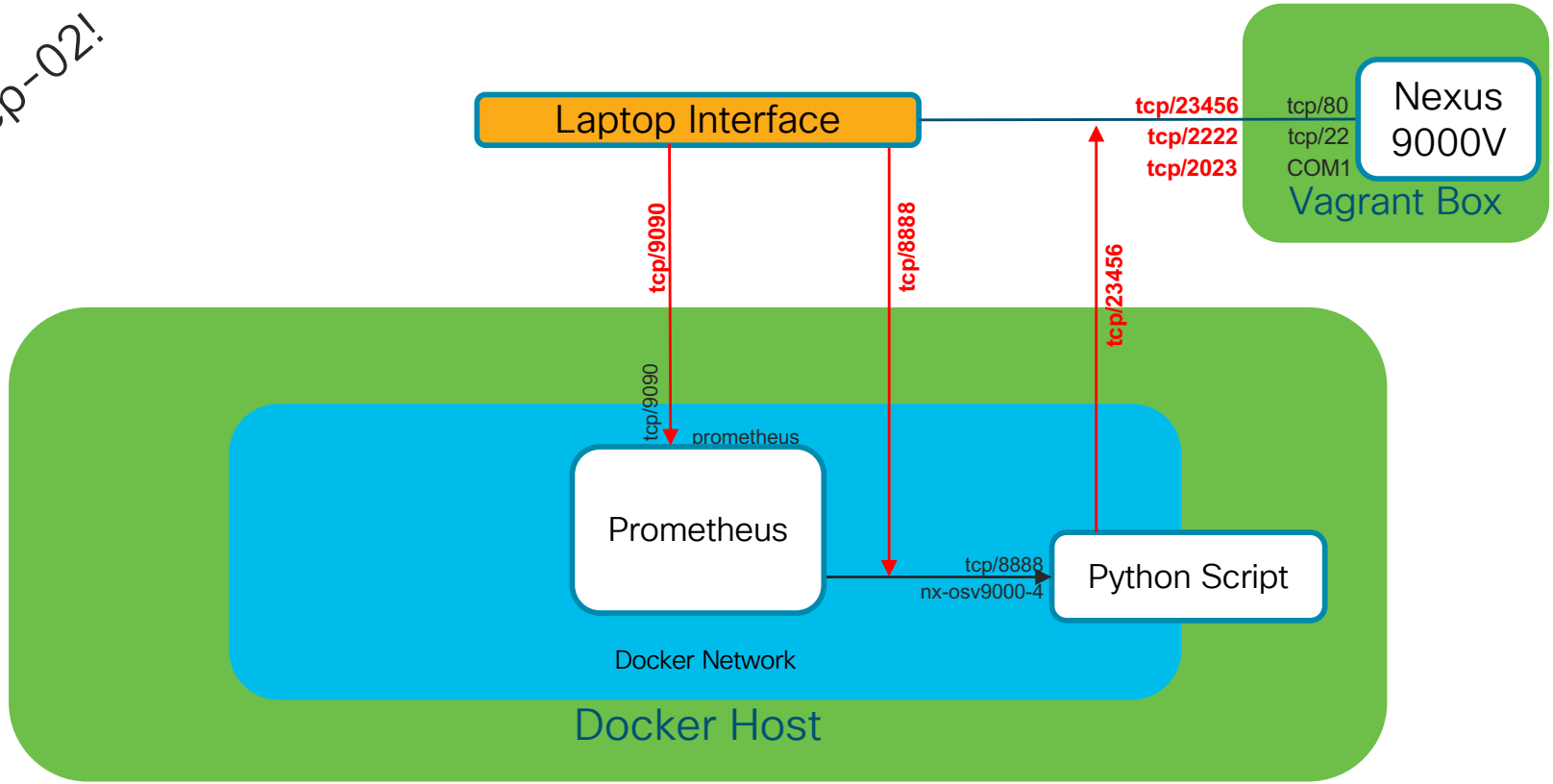
- Deploy Collector container

```
docker run --name collector -d --network demo0 -p 127.0.0.1:8888:8888 \
               devwks-2594/publish_l2table
```



Metric Collection Service Connectivity Diagram

Same as Step-02!



Step-03 – Expand metrics collected

Refactor Python Code

- Clean up from step-02 (cleanup_step02.sh)
- Build Docker image of Collector script

```
docker build -t devwks-2594/step-03:latest -t devwks-2594/step-03:1 .
```

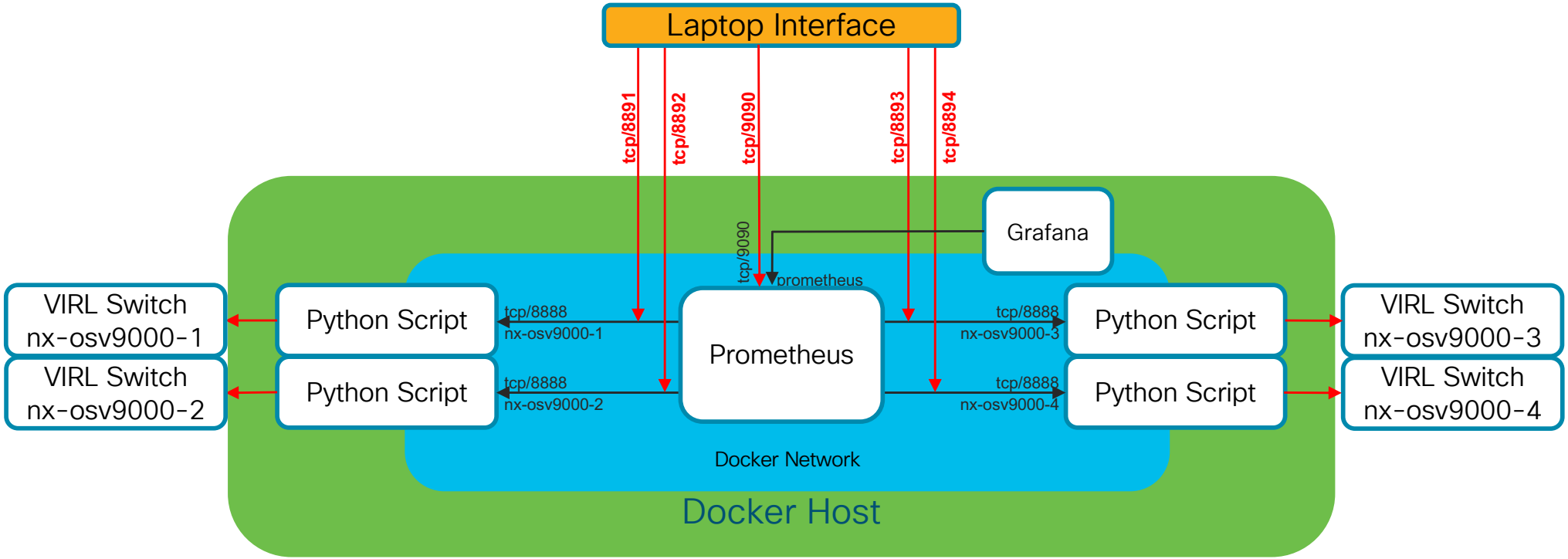
- Deploy Prometheus container

```
docker run --name prometheus -d --network demo0 -p 127.0.0.1:9090:9090 \
           -v ${PWD}/prometheus.yml:/etc/prometheus/prometheus.yml \
           quay.io/prometheus/prometheus
```

- Deploy Collector container

```
docker run --name icam -d --network demo0 -p 127.0.0.1:8888:8888 \
              devwks-2594/step-03
```

Metric Service Connectivity Diagram



Step-04 – Routing Metrics and Sandbox

- Clean up from step-03 (cleanup_step03.sh)
- VPN to the DEVNET Sandbox using instructions

- Build Docker image of Collector script

```
docker build -t devwks-2594/step-04:latest -t devwks-2594/step-04:1 .
```

- Deploy Prometheus container

```
docker run --name prometheus -d --network demo0 -p 127.0.0.1:9090:9090 \
           -v ${PWD}/prometheus.yml:/etc/prometheus/prometheus.yml \
           quay.io/prometheus/prometheus
```

Step-04 – Routing Metrics and Sandbox

- Deploy 4 collector container instances

```
docker run --name nx-osv9000-1 -d --network demo0 -p 127.0.0.1:8891:8888 \
        -e "NXAPI_HOST=172.16.30.101" -e "NXAPI_PORT=80" \
        -e "NXAPI_USER=cisco" -e "NXAPI_PASS=cisco" \
        devwks-2594/step-04
docker run --name nx-osv9000-2 -d --network demo0 -p 127.0.0.1:8892:8888 \
        -e "NXAPI_HOST=172.16.30.102" -e "NXAPI_PORT=80" \
        -e "NXAPI_USER=cisco" -e "NXAPI_PASS=cisco" \
        devwks-2594/step-04
```

Step-04 – Routing Metrics and Sandbox

- Deploy 4 collector container instances

```
docker run --name nx-osv9000-3 -d --network demo0 -p 127.0.0.1:8893:8888 \
        -e "NXAPI_HOST=172.16.30.103" -e "NXAPI_PORT=80" \
        -e "NXAPI_USER=cisco" -e "NXAPI_PASS=cisco" \
        devwks-2594/step-04
docker run --name nx-osv9000-4 -d --network demo0 -p 127.0.0.1:8894:8888 \
        -e "NXAPI_HOST=172.16.30.104" -e "NXAPI_PORT=80" \
        -e "NXAPI_USER=cisco" -e "NXAPI_PASS=cisco" \
        devwks-2594/step-04
```

Step-04 – Routing Metrics and Sandbox

- Deploy Grafana instance

```
docker run --name grafana -d --network demo0 \
           -p 127.0.0.1:3000:3000 \
           grafana/grafana
```

Summary

You make networking **possible**

Cisco live!

Summary

What We Have Done

- Use your favorite CLI show commands to script metric generation
- Leverage containers to perpetually generate them
- Collect and visualize those metrics via next generation tools

What You Can Do

- Leverage DevNet resources to learn more about all of this
  - Python, Containers, Kubernetes, Git, Visualization, etc.
- Ease your operational pains

Complete your online session evaluation

- Please complete your session survey after each session. Your feedback is very important.
- Complete a minimum of 4 session surveys and the Overall Conference survey (starting on Thursday) to receive your Cisco Live water bottle.
- All surveys can be taken in the Cisco Live Mobile App or by logging in to the Session Catalog on ciscolive.cisco.com/us.

  Cisco Live sessions will be available for viewing on demand after the event at ciscolive.cisco.com.

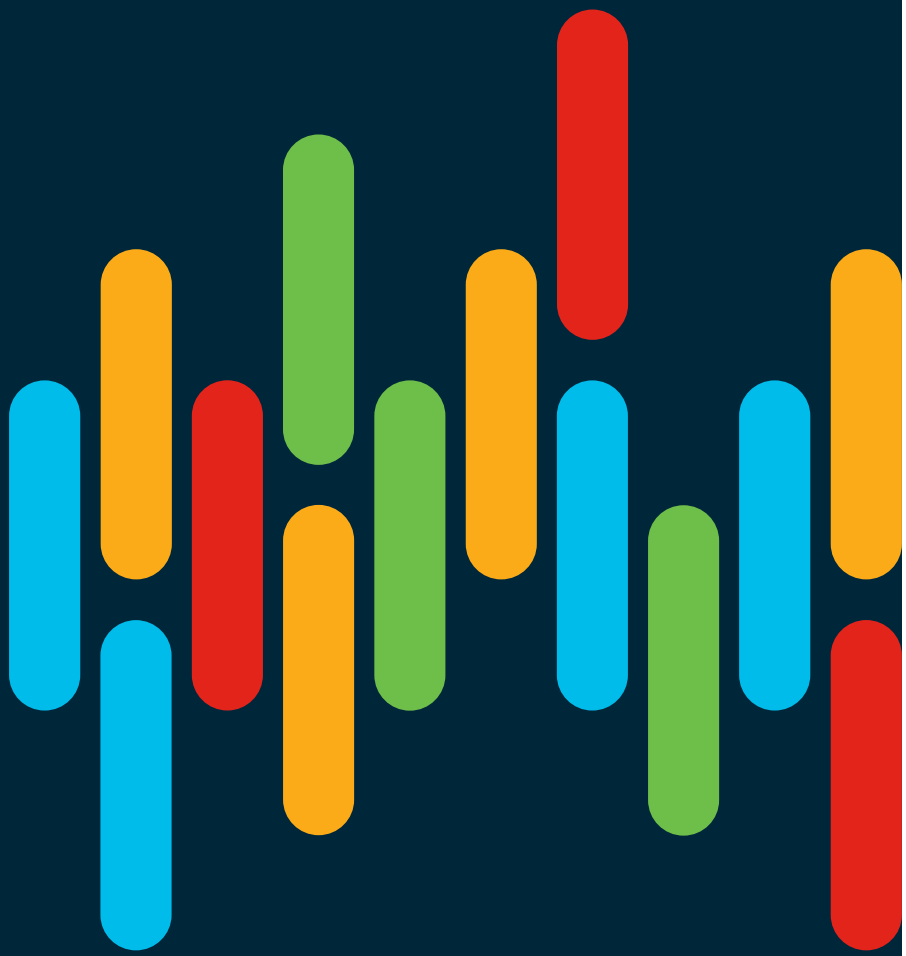
Continue your education

Demos in the Cisco campus

Walk-in self-paced labs

Meet the engineer 1:1 meetings

Related sessions



Thank you

Cisco live!

#CLUS

You make **possible**