

>>> network .toCode()

Enabling Network Automation for Everyone with Nautobot Jobs

London Network Automation Meetup #14

Jason Edelman
@jedelman8

October 12, 2022

>>> Agenda

- Introduction
- Day in the Life of Getting Started with Network Automation
- Introduction to Nautobot Jobs
- Diving into Nautobot Jobs with Schedules, Approvals, and JobHooks
- Demos

>>> Who Am I?

Nights & Weekends...



Days, nights, & weekends

>>> network .toCode()

- Based in NJ (NYC Metro in US)
- Network Engineer (2000 - 2014)
 - Cisco, BlueWater, Presidio
 - CCIE 15394 (expired)
- Founder & CTO of Network to Code (2014 - current)
- Co-author O'Reilly Network Programmability & Automation

4th time speaking at LNAM
LNAM: #4, #9, #11, 14

>>> Hacktoberfest & Hackathon - October 2022

Two Events, One Month

Hacktoberfest

- Full month of October
- Awards based on collective contributions during the time period
- Participate as an individual

Hackathon

- October 14th & 15th
- Presentation submissions open from October 17th through 24th
- Participate as an individual or as a team



>>> network .toCode()



- Collab on submission with NTC advisors
- Join #hacktoberfest-2022 in NTC Slack (slack.networktocode.com to signup) for more details

- Contribute to Network to Code and/or Nautobot GitHub organizations
- Win NTC & Nautobot Swag & Prizes!!!



Day in the Life of Getting Started with Network Automation

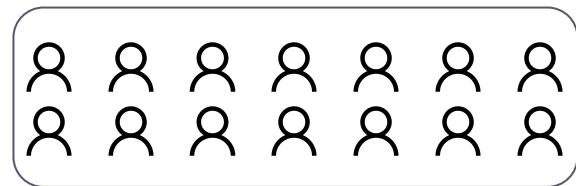
>>> Somewhere along the way...

- The network team has discovered that automation is their friend
- Several team members have built scripts that help them perform their daily tasks
- Other team members catch on and want to use these scripts
- The team does not have a great way to share these scripts
- The organization does not have a proper way to operationalize these scripts
- Team members begin to share files, perhaps even via git
- Scripts are hard to keep updated and sometimes the wrong version is used



Bob's scripts Mary's scripts Bill's scripts

Each engineer maintains and executes their own scripts not understanding or trusting others scripts. Each is usually stored locally with ad-hoc changes each day.



Every other IT team (Ops, Security, Cloud, Network) that could take advantage of the network teams scripts - no access or way to execute programmatically without deep tech skills.

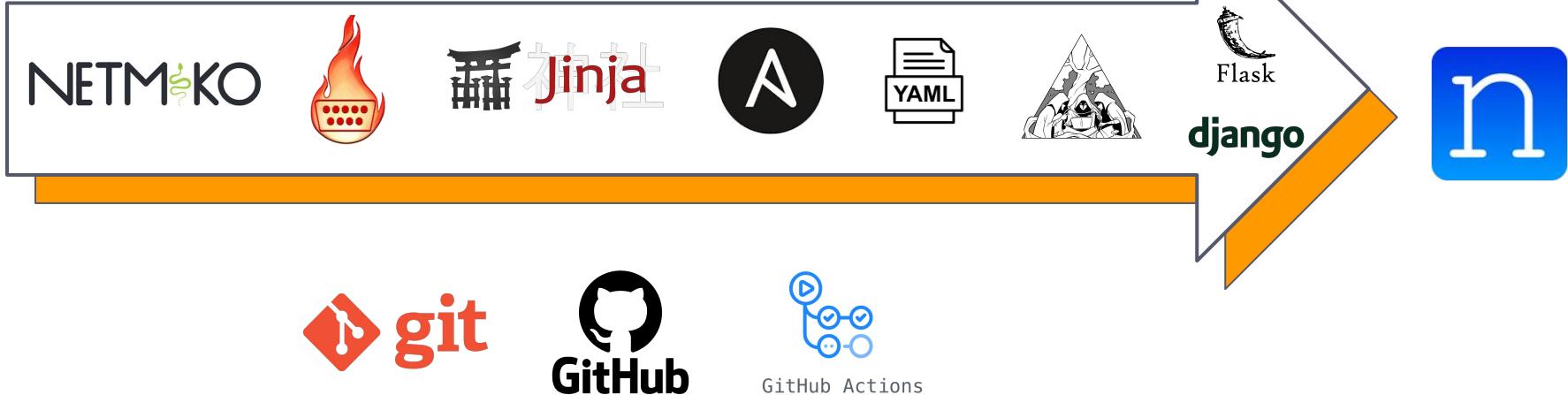
>>> Looking at the Technical Journey

*This is so simple,
amazing!*

*This is even simpler,
more amazing!*

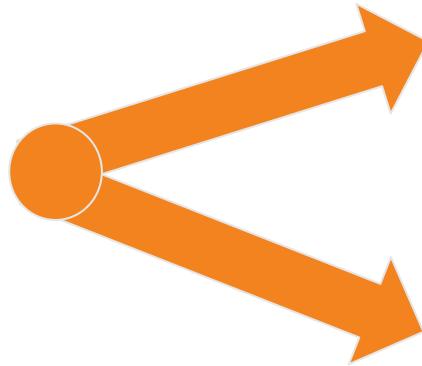
*How does this
scale?*

*How do we
operationalize this
all?*

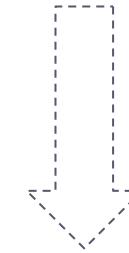


>>> Nautobot 101

All Open Source



Source of Truth

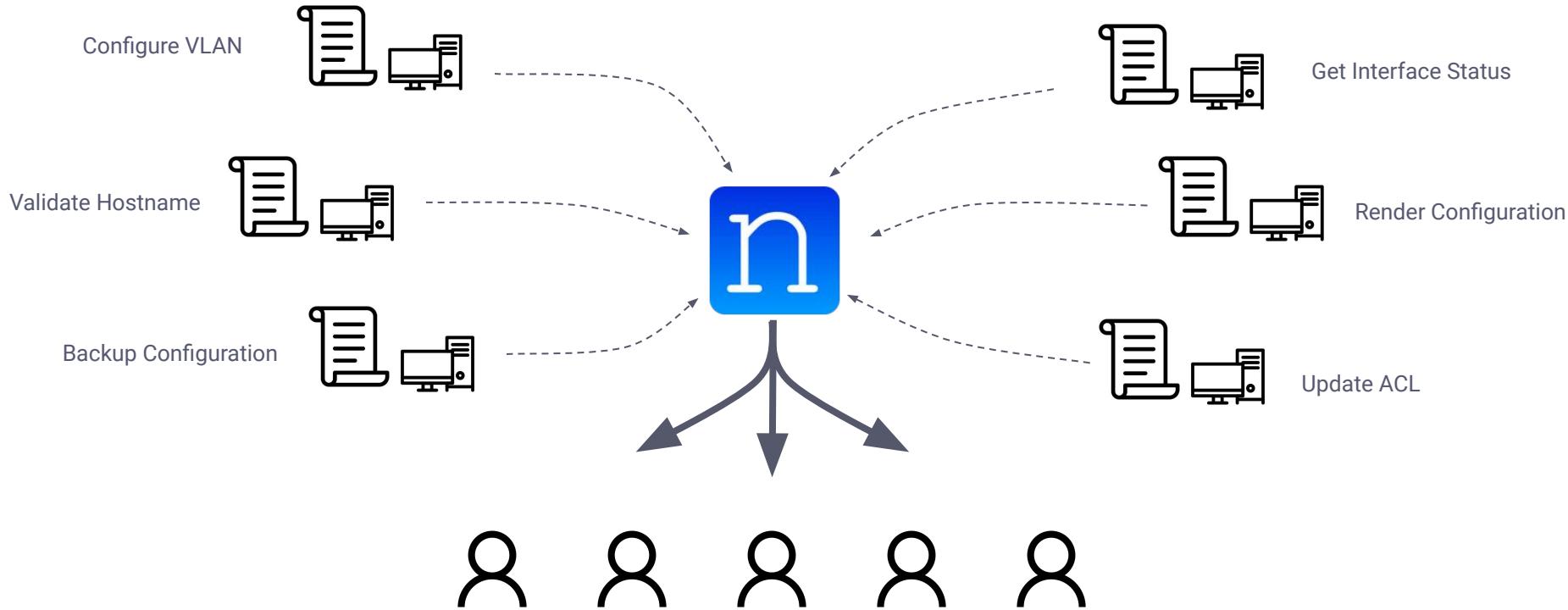


*Data-Driven
Automation*

Automation Platform

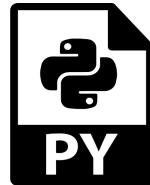
Our Focus Today

>>> Nautobot as the Center of Automation

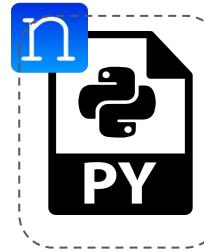


*Move your automation scripts to **Nautobot Jobs** and consume them centrally*

>>> Operationalize Python Scripts as Nautobot Jobs



Add Nautobot Python objects



Standalone Python Scripts

- Executed Locally
- Lacks auditability & permissions
- No way for business to consume

Nautobot Jobs

- Easily convert scripts to Jobs
- Executed via Self-Service Form
- Executed via API
- UI-based Git integration
- Support for Schedules, Approvals, and more

The background of the slide features a nighttime cityscape with numerous skyscrapers and illuminated windows, creating a professional and modern atmosphere.

>>> Nautobot Jobs Overview

>>> Nautobot Jobs

Jobs offer the ability for users to define Python scripts which take user input before running.

- **Inspect** and **validate** data over time
- **Report** only, or change data if desired
- Perform **network automation tasks**

Great way to provision data in a standard way

- Enforces patterns and conventions
- Circumvents some data issues

Also great for continuous data reporting

A central platform for shared automation scripting

The screenshot shows the 'Create a Bridge Domain in ACI' job configuration page. It includes fields for 'Bridge Domain Name' (with placeholder 'Enter Bridge Domain'), 'ACI Tenant:VRF' (with placeholder 'Select ACI Tenant and VRF within the tenant'), 'Prefix' (set to '10.0.10.0/24'), and 'Prefix Length' (set to '/24'). A note at the bottom states 'Next available subnet will be allocated from this prefix'. There are 'Run' and 'Source' tabs at the top of the form.

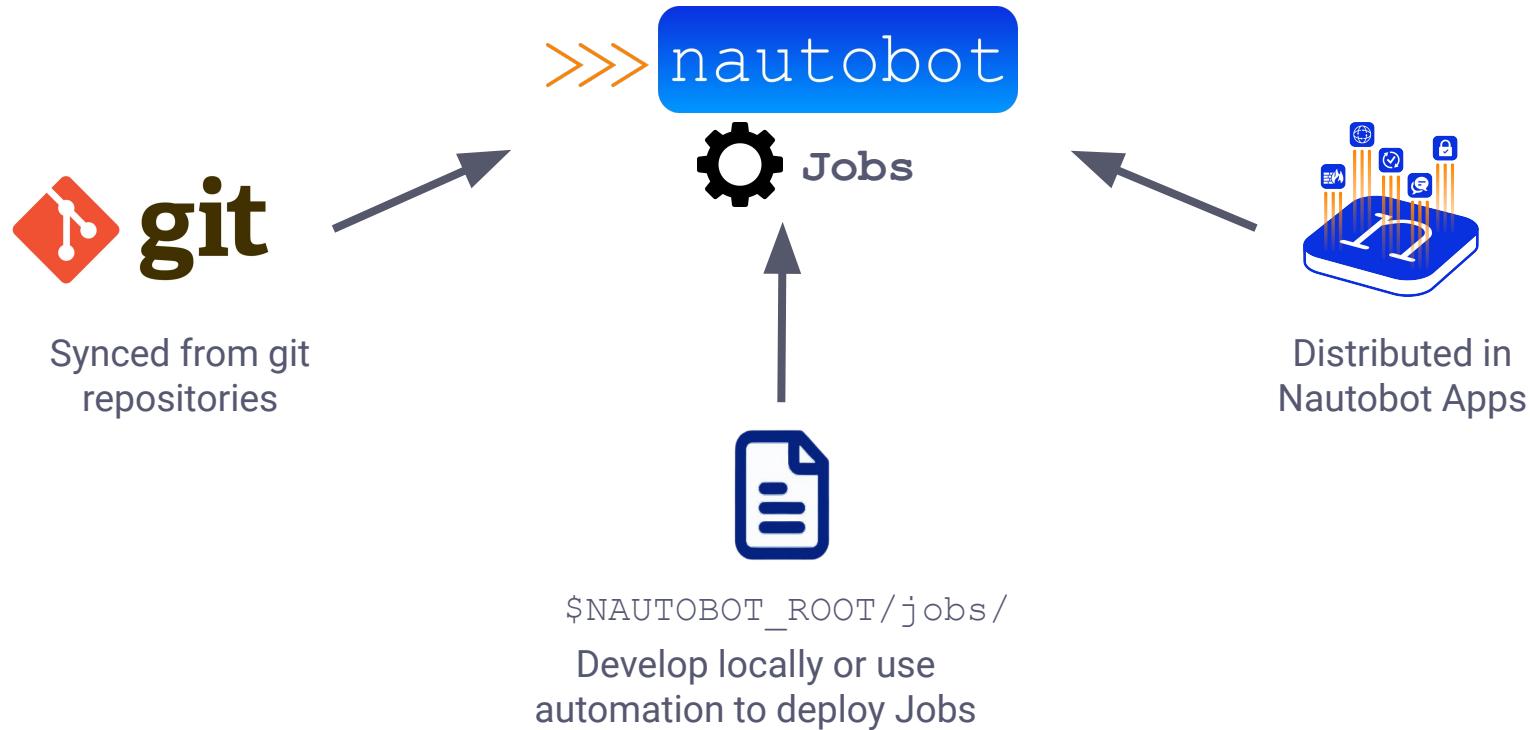
The screenshot shows the 'Job Data' configuration page. It includes fields for 'Region' (placeholder '-----'), 'Site Code' (placeholder 'Site Code'), and 'Leaf switches count' (placeholder 'Leaf switches count'). Below these fields is a checkbox labeled 'Commit changes' with the sub-instruction 'Commit changes to the database (uncheck for a dry-run)'. At the bottom right are 'Run Job' and 'Cancel' buttons.

>>> Nautobot Jobs

- Enterprise access and governance controls
 - RBAC
 - Approvals
 - Scheduling
 - Administrative configuration
 - Flexible logging and auditing
 - Secret and sensitive data integrations & controls
- Distributed Python execution environment
 - Scalable deployment options
 - Single node to global workers
- Integrated access to data within Nautobot
 - Intelligent user input using existing data
 - Easy data manipulation within a Job

Properties		
Approval required	<input checked="" type="checkbox"/> <i>Override default value</i> If set, the configured value will remain even if the underlying Job source code changes	<input type="checkbox"/> Approval required Whether the job requires approval from another user before running
Commit default	<input type="checkbox"/> <i>Override default value</i> If set, the configured value will remain even if the underlying Job source code changes	<input type="checkbox"/> Commit default Whether the job defaults to committing changes when run, or defaults to a dry-run
Hidden	<input type="checkbox"/> <i>Override default value</i> If set, the configured value will remain even if the underlying Job source code changes	<input type="checkbox"/> Hidden Whether the job defaults to not being shown in the UI
Read only	<input type="checkbox"/> <i>Override default value</i> If set, the configured value will remain even if the underlying Job source code changes	<input type="checkbox"/> Read only Whether the job is prevented from making lasting changes to the database
Soft time limit	<input type="checkbox"/> <i>Override default value</i> If set, the configured value will remain even if the underlying Job source code changes	0 Maximum runtime in seconds before the job will receive a SoftTimeLimitExceeded exception. Set to 0 to use Nautobot system default
Time limit	<input type="checkbox"/> <i>Override default value</i> If set, the configured value will remain even if the underlying Job source code changes	0 Maximum runtime in seconds before the job will be forcibly terminated. Set to 0 to use Nautobot system default
Has sensitive variables	<input checked="" type="checkbox"/> <i>Override default value</i> If set, the configured value will remain even if the underlying Job source code changes	<input checked="" type="checkbox"/> Has sensitive variables Whether this job contains sensitive variables

>>> Job Delivery & Management



Three primary way for getting Jobs into Nautobot

>>> Python-Driven UI Forms

```
class Meta:  
    """Meta class for AddVlan"""\n\n    name = "Add VLAN to Selected Device"  
    description = "Executes the Ansible Tower Job to add a new VLAN to the switch."  
  
    device = ObjectVar(  
        model = Device,  
        required = True,  
        # We only want to show spines in JCY  
        query_params = {  
            "site": "jcy",  
            "name": ["jcy-spine-01.infra.ntc.com", "jcy-spine-02.infra.ntc.com"],  
        }  
    )  
  
    vlan = StringVar(  
        description = "The VLAN ID",  
        required = True  
    )  
  
    vlan_name = StringVar(  
        description = "The VLAN name",  
        required = True  
    )  
  
    def run(self, data=None, commit=None):  
        """Executes the job"""\n
```

Access any data already in Nautobot

```
# Specify a job input of a Dynamic Choice Field with all Sites  
site = ObjectVar(model=Site)
```



Jobs / NOC Tasks / Add VLAN to Selected Device / Submission and Scheduling

Add VLAN to Selected Device

Executes the Ansible Tower Job to add a new VLAN to the switch.

Run

Job Data

Device

- jcy-spine-01.infra.ntc.com
- jcy-spine-02.infra.ntc.com

Vlan name

None

The VLAN name

>>> Creating Dynamic Forms and Chaining Filters Together

```
class VerifyHostnameNoInput(Job):  
    """Job without inputs."""
```

```
    # Specify a job input of a Dyna  
    site = ObjectVar(model=Site)
```

Choose a SITE - in Nautobot

```
    # Specify a job input of a Dyna  
    device = ObjectVar(model=Device, query_params={"site_id": "$site"})
```

Choose a DEVICE - in that SITE

```
    # Specify a job input of a Dyna  
    iface = ObjectVar(model=Interface, query_params={"device_id": "$device"})
```

Choose an INTERFACE - on that DEVICE

```
    # Specify a job input VLAN to b  
    vlan = StringVar()
```

Configure - that INTERFACE

This example just accepts an arbitrary VLAN ID.

>>> Execute a Job via API

Give teams programmatic access to your Jobs

```
curl -X POST \
      -H "Authorization: Token b7tryandcopymytoken617786cbcd3" \
      -H "Content-Type: application/json" \
      -H "Accept: application/json; indent=4" \
      https://nautobot.demo.networktocode.com/api/extras/jobs/git.kitt-nautobot
-demos/02-netmiko-nautobot-job-verify-hostname/VerifyHostnameNoInput/run/ \
--data '{"commit": true}'
```

>>> Nautobot Jobs Advanced Features

Approvals, Scheduling & Job Hooks

>>> Job Approvals

Jobs can optionally require an approval before execution.

- Users can run jobs in “dry-run” mode
- Approvers have specific permissions
- Dedicated queue for approvals

Job Data

Device

Commit changes
Commit changes to the database (uncheck for a dry-run)

Job Execution

Type

The job can either run immediately, once in the future, or on a recurring schedule.

This job requires approval to run or schedule.

Jobs Approval Queue

Name	Job	Execution Type	Requestor	Requested	Actions
Generate FW Config via Capirca - 2022-10-06 08:00:49.101349+00:00	Generate FW Config via Capirca.	Once in the future	ntc	Oct. 6, 2022 8:00 a.m.	

Showing 1-1 of 1

Job Data

Device

Commit changes
Commit changes to the database

Approval Request

Requestor

Requested Start Time

>>> Job Scheduling

Jobs can be scheduled to run in the future or on a recurring basis.

- Scheduling during a maintenance window
- Recurring data quality and sync jobs
- Simplified schedule options
- Advanced crontab support (v1.3.10)

Job Execution

Type: Recurring custom
The job can either run immediately, once in the future, or on a recurring schedule.

Schedule name: T/W/F Data Sync
Name for the job schedule.

Starting date and time: 2022-10-06 12:00:00
The scheduled time is relative to the Nautobot configured timezone: UTC.

Crontab: 0 0 * * 1,3,5
Custom crontab syntax (* * * * *)

Schedule Job Cancel

Job Execution

Type: ✓ Once immediately
Once in the future
Recurring hourly
Recurring daily
Recurring weekly
Recurring custom

Run Job Now Cancel

>>> Job Hooks

Job Hooks allow for advanced event based automation (v1.4.0).

Execute jobs automatically when events occur in Nautobot.

Data change events:

- Create
- Update
- Delete

Add a new job hook

Job hook

Name	Deploy VLAN Assignments
Content Type(s)	x dcim interface
Job	VLAN Deployment Hook

The job that this job hook will initiate

Enabled

Type create
Call this job hook when a matching object is created.

Type update
Call this job hook when a matching object is updated.

Type delete
Call this job hook when a matching object is deleted.

[Create](#) [Create and Add Another](#) [Cancel](#)



>>> Demos

>>> Nautobot Jobs Demos

Meetup

- Review Netmiko Scripts
- Jobs UI Overview
- Review & Execute Netmiko Job 1
- Review & Execute Netmiko Job 2
- Review & Execute Netmiko Job 3
- Update Job Name
- Generate cable plan Job - NOC Tasks
- Perform device update and trigger JobHook

Due to shotty wireless, the demo will be done in slides.



>>> Job: Assign VLAN to Interface



- Jobs
- Jobs
- Job Approval Queue
- Scheduled Jobs
- Job Results
- Job Hooks



Jobs

Name	Enabled	Description	Last run	status	
<h3>▼ 02-netmiko-nautobot-job-verify-hostname</h3>					
Verify Hostname Pattern For NYC	✓	Checks all devices at NYC for configured hostname pattern.	Oct. 12, 2022 7:56 a.m.	Completed	
<h3>▼ 03-netmiko-nautobot-job-verify-hostname-site-selection</h3>					
Verify Hostname Pattern For Any Site	✓	Checks all devices at any Site for configured hostname pattern.	Oct. 11, 2022 9:22 a.m.	Completed	
<h3>▼ 04-netmiko-nautobot-job-assign-vlan-to-interface</h3>					
Add VLAN to an Interface	✓	Configures the specified VLAN on an Interface.	Oct. 12, 2022 7:58 a.m.	Completed	
<h3>▼ Capirca Jobs</h3>					
Generate FW Config via Capirca.	✓	Generate FW Config via Capirca and update the models.	Sept. 21, 2022 8:01 p.m.	Completed	
<h3>▼ Circuit Maintenance</h3>					
Update Circuit Maintenances	✓	Fetch Circuit Maintenance Notifications from Sources and create or update Circuit Maintenances accordingly.	Never	N/A	



Job Result: Verify Hostname Pattern For NYC

Re-RunJob Result Advanced Additional Data

Summary of Results

Job Description Checks all devices at NYC for configured hostname pattern.

Status Failed

Started at Oct. 12, 2022 7:56 a.m.

User ntc

Duration 0 minutes, 15.35 seconds

Logs

 Filter log level or message

Time	Grouping	Level	Object	Message
2022-10-12 07:56:24.297319	run	Success	nyc-bb-01.infra.ntc.com	Configured hostname is correct.
2022-10-12 07:56:26.454037	run	Failure	nyc-leaf-01.infra.ntc.com	Does Not Match Hostname Pattern.
2022-10-12 07:56:28.728762	run	Success	nyc-leaf-02.infra.ntc.com	Configured hostname is correct.
2022-10-12 07:56:30.829038	run	Success	nyc-rtr-01.infra.ntc.com	Configured hostname is correct.
2022-10-12 07:56:32.853534	run	Success	nyc-rtr-02.infra.ntc.com	Configured hostname is correct.

>>> Creating Dynamic Forms and Chaining Filters Together

```
class VerifyHostnameNoInput(Job):  
    """Job without inputs."""
```

```
    # Specify a job input of a Dyna  
    site = ObjectVar(model=Site)
```

Choose a SITE - in Nautobot

```
    # Specify a job input of a Dyna  
    device = ObjectVar(model=Device, query_params={"site_id": "$site"})
```

Choose a DEVICE - in that SITE

```
    # Specify a job input of a Dyna  
    iface = ObjectVar(model=Interface, query_params={"device_id": "$device"})
```

Choose an INTERFACE - on that DEVICE

```
    # Specify a job input VLAN to b  
    vlan = StringVar()
```

Configure - that INTERFACE

This example just accepts an arbitrary VLAN ID.

The background of the slide is a photograph of a city skyline at night, likely New York City, featuring numerous skyscrapers with their lights on.

>>> Job: Emailing Cable Plans

▼ NOC Tasks

Add VLAN to Selected Device	✓	Executes the Ansible Tower Job to add a new VLAN to the switch.	Aug. 17, 2022 6:41 p.m. by ntc	Completed			
Email Cable Plan and Rack Elevation Diagrams	✓	This Job automatically sends a cable plan as HTML and CSV along with SVG for each rack.	Oct. 12, 2022 7:46 a.m. by ntc	Completed			

>>> nautobot

Organization ▾ Devices ▾ IPAM ▾ Virtualization ▾ Circuits ▾ Power ▾ Device Lifecycle ▾ Secrets ▾ Jobs ▾ Extensibility ▾ Security ▾ Golden Config ▾

Jobs / NOC Tasks / Email Cable Plan and Rack Elevation Diagrams / Submission and Scheduling

Email Cable Plan and Rack Elevation Diagrams

Read-only

This Job automatically sends a cable plan as HTML and CSV along with SVG for each rack.

Job Data

Site: -----

Email: None
The e-mail address to send the cable plan

ⓘ This job may have sensitive variables and can only be executed immediately. This can be changed in the Job edit view to set 'has_sensitive_variables' to 'False' if appropriate.



1 of 6,823

1

Cable Plan and Rack Elevations for ATL ➔ Inbox x



netdevops@networktocode.com

to me ▾

 8:47 AM (6 hours ago)

☆

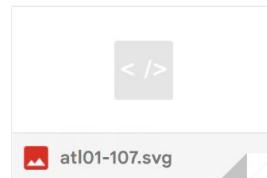
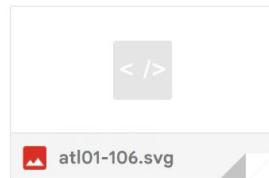
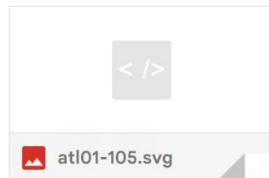
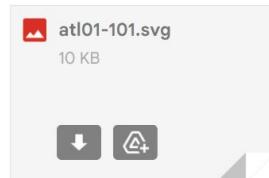
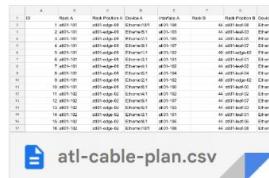
←

•

Please see the attached cable plan and rack elevations for ATL.

-NTC

10 Attachments • Scanned by Gmail ⓘ



The cable plan:

ID	Rack	Rack Position	Device	Interface		Rack	Rack Position	Device	Interface	Connected
1	atl01-101	40	atl01-edge-01	Ethernet10/1	<--->	atl01-108	44	atl01-leaf-08	Ethernet1	
2	atl01-101	40	atl01-edge-01	Ethernet5/1	<--->	atl01-103	44	atl01-leaf-03	Ethernet1	
3	atl01-101	40	atl01-edge-01	Ethernet7/1	<--->	atl01-105	44	atl01-leaf-05	Ethernet1	
4	atl01-101	40	atl01-edge-01	Ethernet9/1	<--->	atl01-107	44	atl01-leaf-07	Ethernet1	
5	atl01-101	40	atl01-edge-01	Ethernet1/1	<--->	atl01-102	40	atl01-edge-02	Ethernet1/1	
6	atl01-101	40	atl01-edge-01	Ethernet3/1	<--->	atl01-101	44	atl01-leaf-01	Ethernet1	
7	atl01-101	40	atl01-edge-01	Ethernet4/1	<--->	atl01-102	44	atl01-leaf-02	Ethernet1	
8	atl01-101	40	atl01-edge-01	Ethernet6/1	<--->	atl01-104	44	atl01-leaf-04	Ethernet1	
9	atl01-101	40	atl01-edge-01	Ethernet2/1	<--->	atl01-102	40	atl01-edge-02	Ethernet2/1	
10	atl01-101	40	atl01-edge-01	Ethernet8/1	<--->	atl01-106	44	atl01-leaf-06	Ethernet1	
11	atl01-102	40	atl01-edge-02	Ethernet4/1	<--->	atl01-102	44	atl01-leaf-02	Ethernet2	
12	atl01-102	40	atl01-edge-02	Ethernet9/1	<--->	atl01-107	44	atl01-leaf-07	Ethernet2	
13	atl01-102	40	atl01-edge-02	Ethernet5/1	<--->	atl01-103	44	atl01-leaf-03	Ethernet2	

```

48
47
46
45
44 atl01-leaf-01
43
42
41
40 atl01-edge-01
39
38
37
36
35
34
33
32
31
30
29
28
27
26
25
24
23
22
21
20
19

```

cableplanandrackel....zip ^



>>> JobHook



Update Device

Created Oct. 11, 2022 · Updated 19 hours, 1 minute ago

Job Hook Advanced Change Log

Job Hook

Object Type(s) • dcim | device

Job DemoJobHookReceiver

Call On: ✘ Create ✓ Update ✘ Delete

Enabled ✓

12 lines (10 sloc) | 517 Bytes

```
1 from nautobot.extras.choices import ObjectChangeActionChoices
2 from nautobot.extras.jobs import JobHookReceiver
3
4
5 class DemoJobHookReceiver(JobHookReceiver):
6     def receive_job_hook(self, change, action, changed_object):
7         if action == ObjectChangeActionChoices.ACTION_UPDATE:
8             # log diff output
9             snapshots = change.get_snapshots()
10            self.log_info(f"CHANGE: {change}")
11            self.log_info(f"ACTION: {action}")
12            self.log_info(f"DIFF: {snapshots['differences']}")
```



>>> nautobot

Organization ▾ Devices ▾ IPAM ▾ Virtualization ▾ Circuits ▾ Power ▾ Device Lifecycle ▾ Secrets ▾ Jobs ▾ Extensibility ▾ Security ▾ Golden Config ▾ Plugins ▾

Job Results / demo-job-hook-receiver / DemoJobHookReceiver / Oct. 12, 2022 2:43 p.m. Search job results

Job Result: DemoJobHookReceiver

Job Result Advanced Additional Data

Summary of Results

Job Description

Status Completed

Started at Oct. 12, 2022 2:43 p.m.

User ntc

Duration 0 minutes, 27.03 seconds

Logs

Time	Grouping	Level	Object	Message
2022-10-12 14:43:44.764605	run	Info	—	CHANGE: dcim device yyz01-leaf-08 updated by ntc
2022-10-12 14:43:44.796165	run	Info	—	ACTION: update
2022-10-12 14:43:44.801752	run	Info	—	DIFF: {'removed': {'notes_url': None}, 'added': {'notes_url': '/api/dcim/devices/6f2a3b79-d01f-4224-910c-58832ccf2cc4/notes/'}}

>>> Resources

Code from today:

<https://github.com/networktocode/2022-ntc-content> -> webinars/nautobot-jobs/jobs

Community Webinar: Simplify Network Automation with Nautobot Jobs

<https://www.youtube.com/watch?v=YzS9AEnR4R0>

```
>>> network .toCode()
```

Thank you.