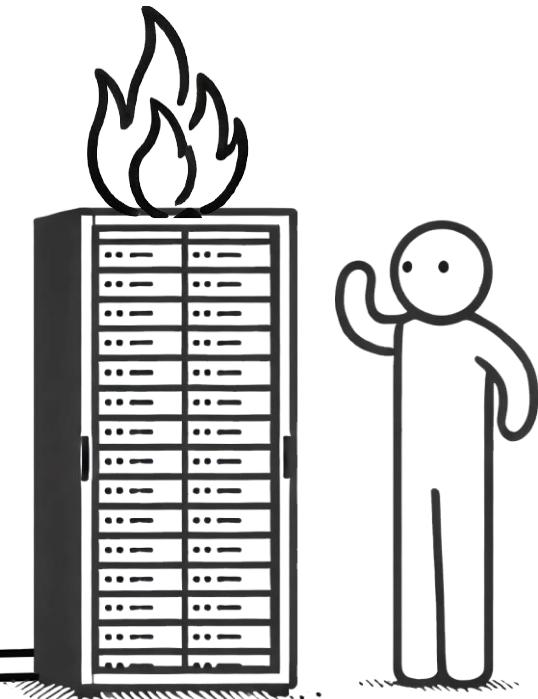


I got Netbox, now what?

10 minutes of fun madness



Steinn “Steinzi” Bjarnarson 11.2024



AGENDA

This talk is for the bottom 50% of the class

- You pick A NSoT
- What do we do After the installation?
 - Get data into it.
- Fun starter projects
- Public service announcements.



Pick an NSoT

You do you fam ❤️

- Lots of choices now!
- I'm not a cop.



>>> nautobot





Cool addon for my NSoT?

Device End-of-Life Report																									
Results		Compliance		Discovery		EoL Lifecycle		Forecast		Cost Analysis		Data Analysis		Filters											
EoL Announced: Yes		Configure Table																							
Quick search																									
Name	Type	Role	EoL Announced	Last Ship Date	EDL Service Requested Count	EDL Last Support Count	EDL Maintenance Count	EDL Vuln. Count	EDL Service Attach. Count	EDL Building Future Count	EDL SW Maintenance Count	EDL Last Sale	EDL Site Maintenance Migration Count	EDL Service Order Day Count											
abve-rest14	WS-C3750X-48P-L	Core Switch	Yes	—	163 weeks	154 weeks	—	239 weeks	333 weeks	333 weeks	313 weeks	166 weeks	—	—											
core-rest14	WS-C3750X-48P-L	Core Switch	Yes	—	163 weeks	154 weeks	—	228 weeks	333 weeks	333 weeks	313 weeks	206 weeks	—	—											
abve-rest14	WS-C3850-24T-S	Distribution Switch	Yes	—	45 weeks	84 weeks	—	64 weeks	124 weeks	124 weeks	114 weeks	116 weeks	—	—											
dist14	WS-C3850-24T-S	Distribution Switch	Yes	—	45 weeks	84 weeks	—	55 weeks	124 weeks	124 weeks	116 weeks	116 weeks	—	—											
abve-rest14	WS-C3850-24T-S	Access Switch	Yes	—	149 weeks	169 weeks	—	168 weeks	19 weeks	19 weeks	19 weeks	77 weeks	—	—											
dist14	WS-C3850-24T-S	Access Switch	Yes	—	149 weeks	169 weeks	—	168 weeks	19 weeks	19 weeks	19 weeks	77 weeks	—	—											
abve-rest14	WS-C3850-24T-S	Switch	Yes	—	163 weeks	154 weeks	—	228 weeks	333 weeks	333 weeks	313 weeks	365 weeks	—	—											
dist14	WS-C3850-24T-S	Switch	Yes	—	163 weeks	154 weeks	—	228 weeks	333 weeks	333 weeks	313 weeks	365 weeks	—	—											
abve-rest14	WS-C3850-24T-S	Switch	Yes	—	84 weeks	—	—	64 weeks	124 weeks	124 weeks	114 weeks	116 weeks	—	—											
dist14	WS-C3850-24T-S	Switch	Yes	—	84 weeks	—	—	64 weeks	124 weeks	124 weeks	114 weeks	116 weeks	—	—											
abve-rest14	WS-C3850-24T-S	Switch	Yes	—	84 weeks	—	—	64 weeks	124 weeks	124 weeks	114 weeks	116 weeks	—	—											
dist14	WS-C3850-24T-S	Switch	Yes	—	84 weeks	—	—	64 weeks	124 weeks	124 weeks	114 weeks	116 weeks	—	—											

Work

Hardware EoX Report | Netos

https://pod01.netos.io/plugins/netos-reporting/eox/devices

Search

demo

Global Device EoX Status Report

Organization

Reporting

MACRO REPORTS

Global Network Status

Financial Forecasts

TBM Infrastructure Analysis

TBM Telecom Analysis

LIFECYCLE

Device EoX

Inventory EoX

demo

Export

Device EoX by Region

EoX State by Region (Latin America and the Caribbean)

EoX Status (%) per Region over multiple years

The chart displays the percentage of devices in various EoX states from 2018 to 2026. The y-axis ranges from 0 to 50%. The x-axis shows years from 2018 to 2026. Multiple colored areas represent different EoX states, with a legend below detailing them.

Legend:

- End of Vulnerability
- End of Last Sale Date
- End of Routine Failure Analysis Date
- End of Service Contract Renewal Date
- Last Day for Service Contract
- Last Day of Support Date
- Maintenance Date
- End of SW Maintenance Release Date
- End of SW Maintenance Migration Date
- End of Service Attach Date

EoX State by Region (Latin America and the Caribbean)

EoX Status per Region

This chart shows the percentage of devices in various EoX states across different regions for each year from 2018 to 2027. The x-axis represents years from 2018 to 2027, and the y-axis represents percentages from 0% to 100%. Each bar is composed of multiple colored segments representing different EoX states.

Legend:

- End of Vulnerability
- End of Last Sale Date
- End of Routine Failure Analysis Date
- End of Service Contract Renewal Date
- Last Day for Service Contract
- Last Day of Support Date
- Maintenance Date
- End of SW Maintenance Release Date
- End of SW Maintenance Migration Date
- End of Service Attach Date

EoX State by Region (Southern Europe)

EoX Status (%) per Region over multiple years

The chart displays the percentage of devices in various EoX states from 2018 to 2026. The y-axis ranges from 0 to 28%. The x-axis shows years from 2018 to 2026. Multiple colored areas represent different EoX states, with a legend below detailing them.

Legend:

- End of Vulnerability
- End of Last Sale Date
- End of Routine Failure Analysis Date
- End of Service Contract Renewal Date
- Last Day for Service Contract
- Last Day of Support Date
- Maintenance Date
- End of SW Maintenance Release Date
- End of SW Maintenance Migration Date
- End of Service Attach Date

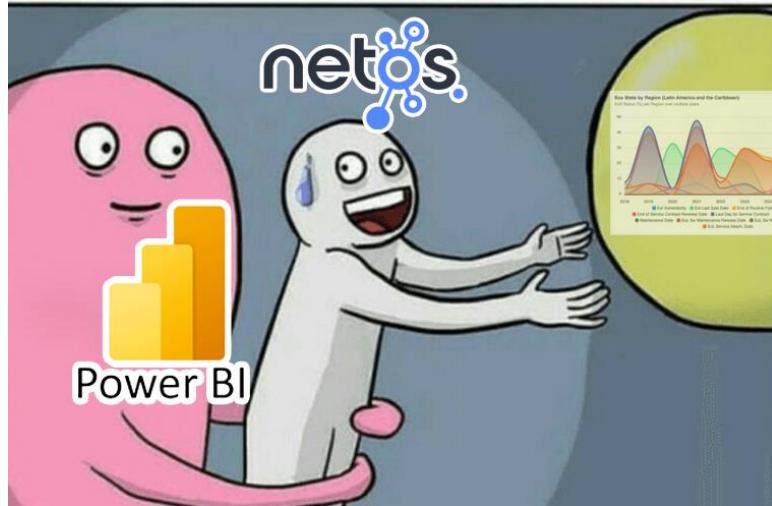
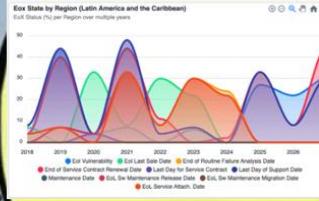
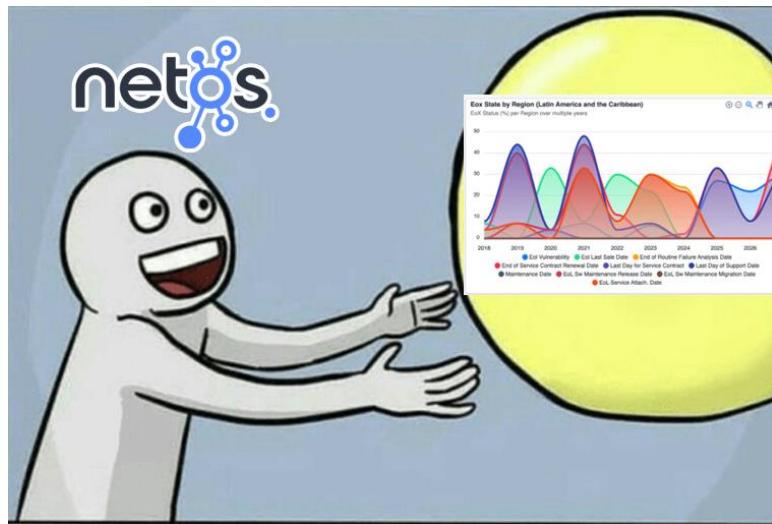
EoX State by Region (Southern Europe)

EoX Status per Region

This chart shows the percentage of devices in various EoX states across different regions for each year from 2018 to 2027. The x-axis represents years from 2018 to 2027, and the y-axis represents percentages from 0% to 80%. Each bar is composed of multiple colored segments representing different EoX states.

Legend:

- End of Vulnerability
- End of Last Sale Date
- End of Routine Failure Analysis Date
- End of Service Contract Renewal Date
- Last Day for Service Contract
- Last Day of Support Date
- Maintenance Date
- End of SW Maintenance Release Date
- End of SW Maintenance Migration Date
- End of Service Attach Date



I got Netbox, now what?

Get “good” data into netbox

Automagic...



By hand...



Do it by hand?



One day of manual data entry in

netbox

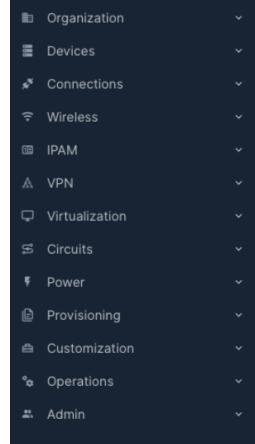


I GOT THE BLACK LUNG POP!

The hard part is keeping sync.

I got Netbox, now what?

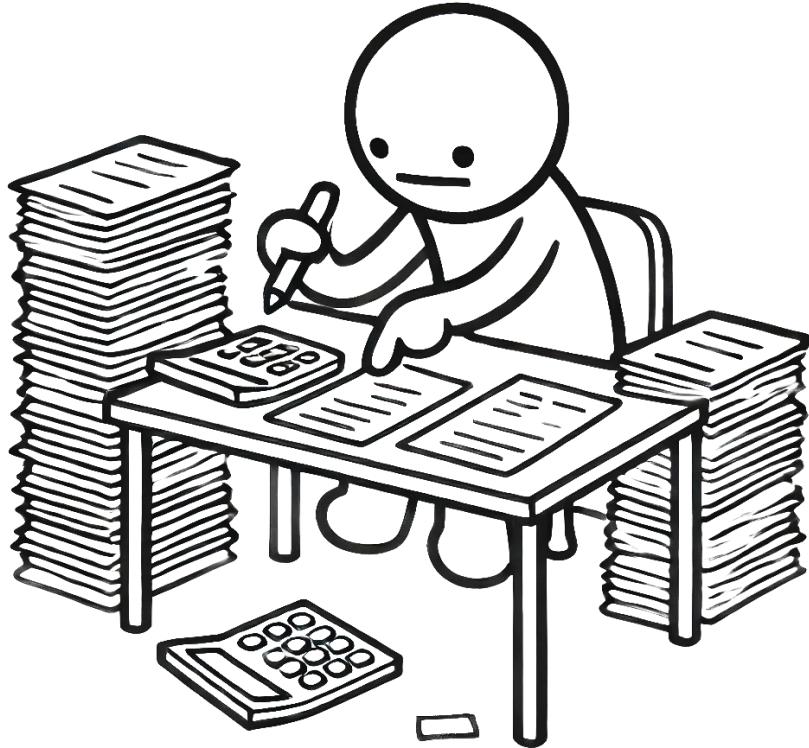
Figure out what Data you NEED!



“Data”

Lots of “It depends”

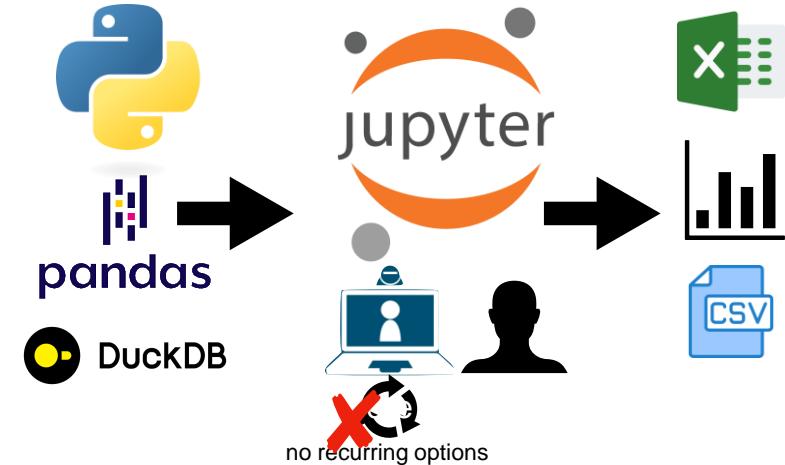
- Systems of Record
- Upstream / Downstream
- Leading / Trailing data
- Transformations
- Checks & Validation



LVL 1 - Scripts

Just run the script locally

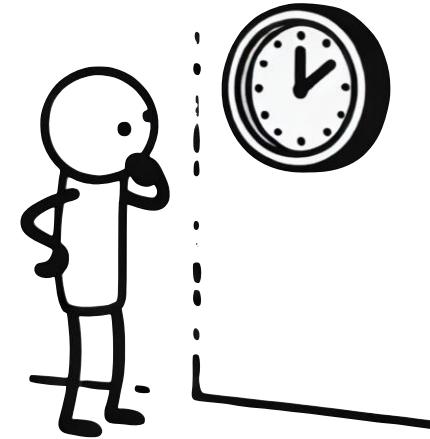
- Just Write the Script in a notebook and run it when needed
- It is great when the boss wants a quick report on something
- “not great” for sharing
 - Don’t share API keys...



A screenshot of a Jupyter Notebook interface. On the left, the code cell contains Python code for generating a scatter plot. The code imports `matplotlib.pyplot` and uses it to scatter training data (green), test data (red), and predicted data (blue) based on chirp counts. It also plots the regression line. The right panel shows the resulting scatter plot titled "Temperature based on chirp count" with axes for "Chirp/minute" and "Temperature".

```
96 #%% [markdown]
97 # Visualize the results
98 #
99 # The following code generates a plot: green dots are
100 # training data, red dots are test data, blue dots are
101 # predictions. Gray line is the regression itself. You see
102 # that all the blue dots are exactly on the line, as they
103 # should be, because the predictions exactly fit the model
104 # (the line).
105 #
106 Run Cell | Run Above | Run Below
107 #%% [code]
108 import matplotlib.pyplot as plt
109
110 plt.scatter(X_train, y_train, color = 'green')
111 plt.scatter(X_test, y_test, color = 'red')
112 plt.scatter(X_test, y_pred, color = 'blue') # The
113 # predicted temperatures of the same X test input.
114 plt.plot(X_train, regressor.predict(X_train), color =
115 'gray')
116 plt.title('Temperature based on chirp count')
117 plt.xlabel('Chirps/minute')
118 plt.ylabel('Temperature')
119 plt.show()
120
121 Run Cell | Run Above | Run Below
122 #%% [markdown]
123 # %% Closing comments
124 #
125 #
126 # At the end of the day, when you create a model, you use
127 # training data. Then you start feeding test data (real
128 # observations) to see how well the model actually works.
129 # You may find that the model is a little inaccurate over
```

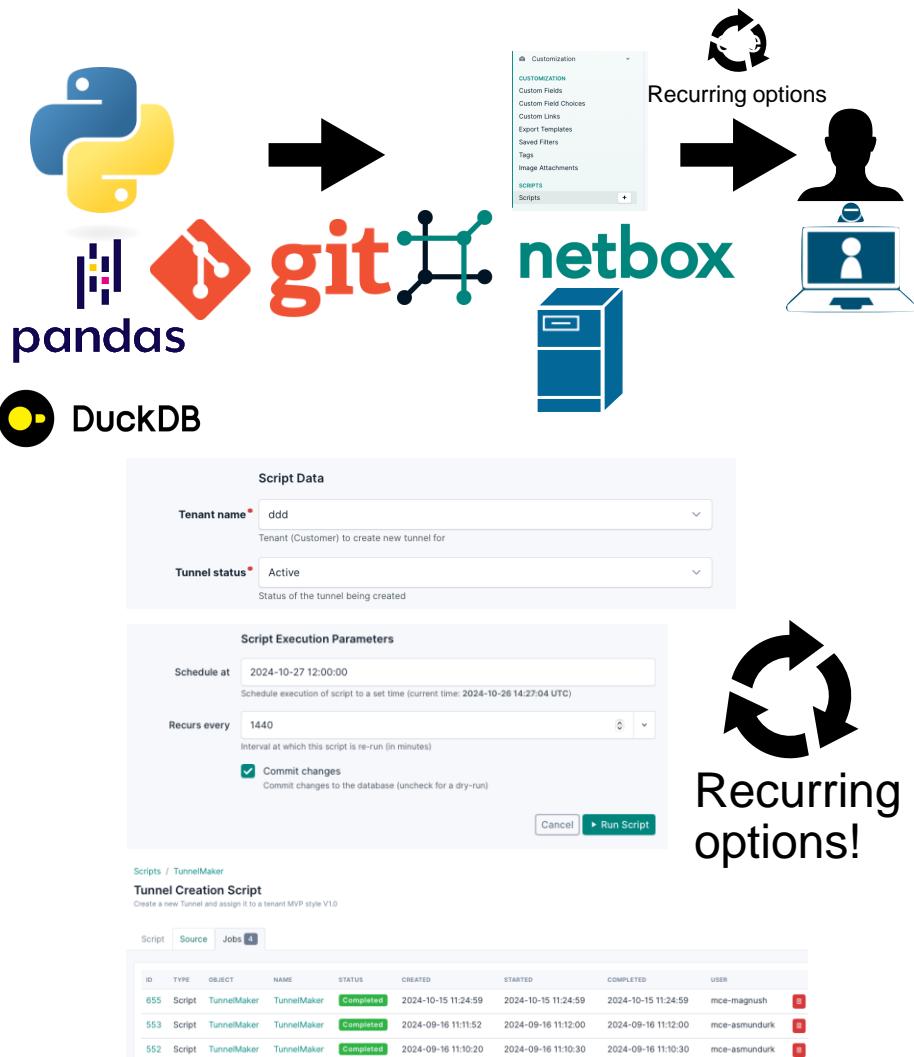
Finding/Documenting/Fixing Cron
Jobs **SUCKS**



LVL 2 - Script in netbox

Run the script from netbox

- Stick the script in netbox via Git repo.
 - **GREAT!** One-off Creation or simple reoccurring Sync!
 - Inputs from netbox via web UI!
 - Easy to share with co-workers
 - **Poor** for debugging
 - Securing API external keys is a pain
 - “mid” observability
 - **hard** to do Pre/post-data testing



Config Generation

Jinja2 inside netbox scripts

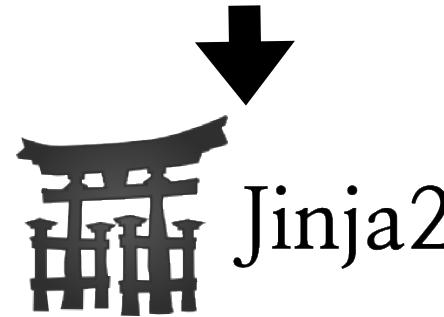
- All the data you want inside of netbox
- and spit out the config
- Maybe even send it to your Change management system?



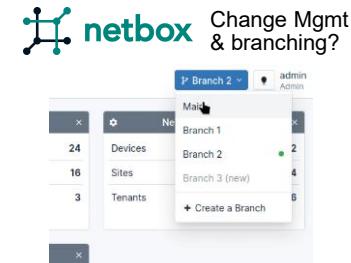
Script Data

Tenant name: Tenant (Customer) to create new tunnel for

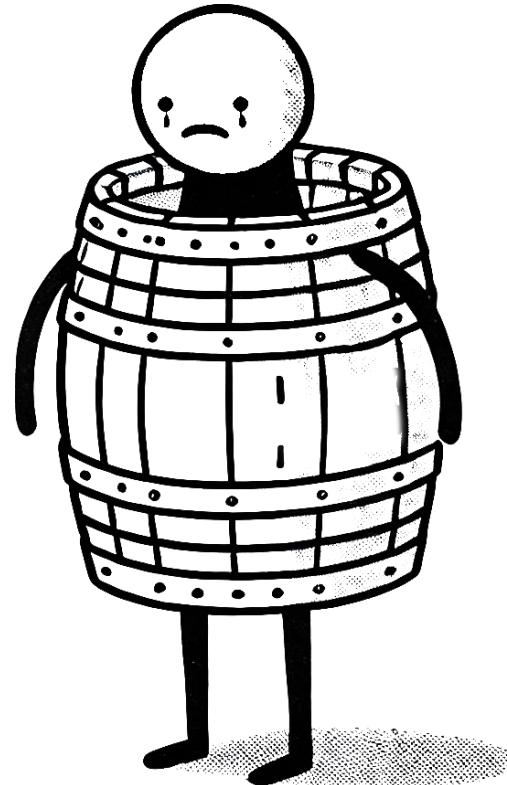
Tunnel status: Status of the tunnel being created



servicenow Jira

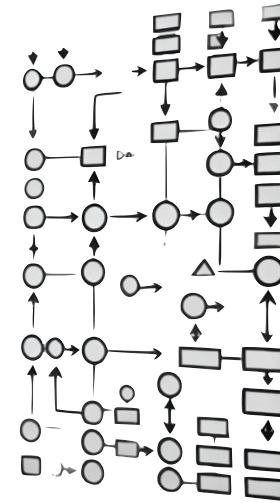
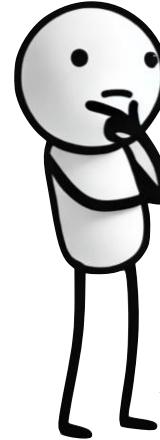


I got Netbox, now what?
**Dollar store netbox
native “DAG”?**

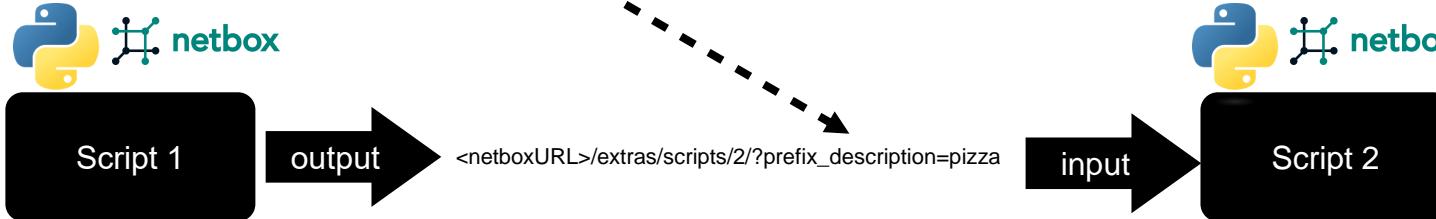


Netbox native “DAG”... Populated URLs feature

- Populate script 2 with data from script 1.
- This is what a crazy person would do...
- but it works...



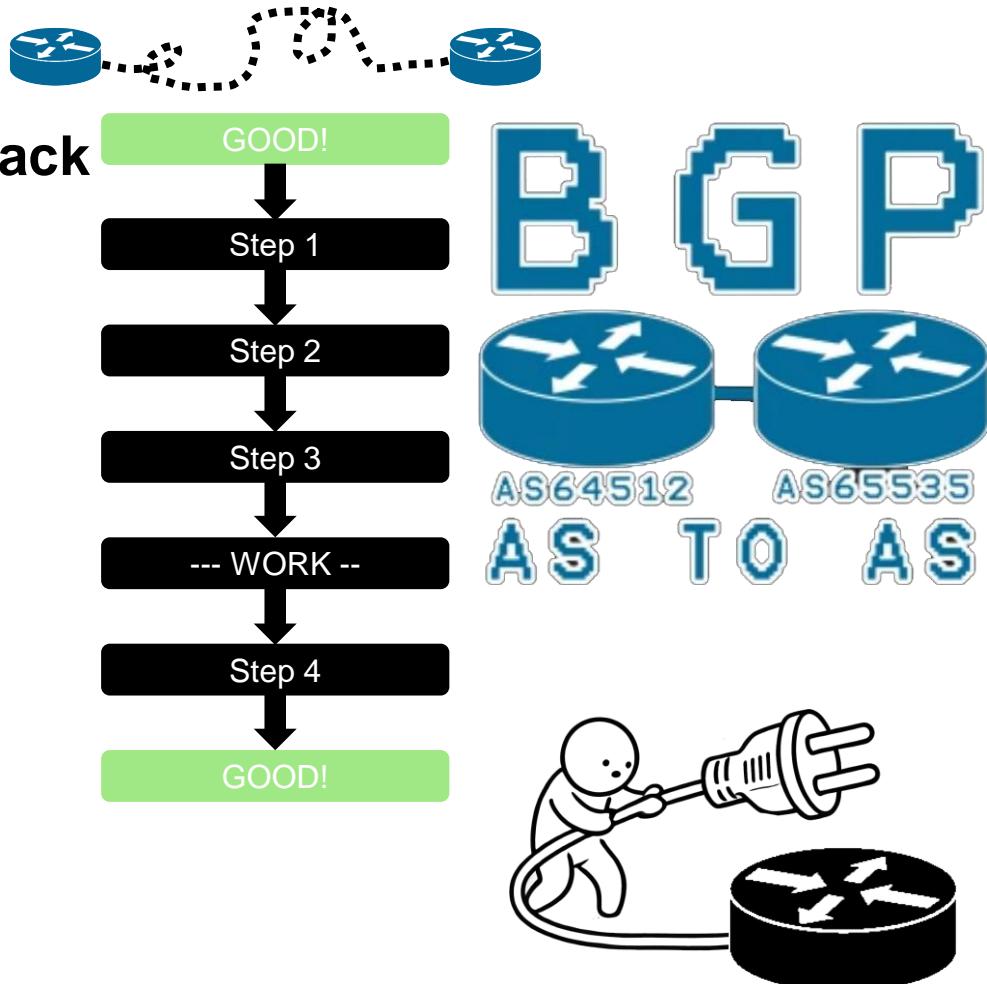
The screenshot shows the Netbox interface with the 'SCRIPTS' section selected in the sidebar. Other visible sections include 'Customization', 'CUSTOMIZATION', 'Custom Fields', 'Custom Field Choices', 'Custom Links', 'Export Templates', 'Saved Filters', 'Tags', and 'Image Attachments'.



“DAG” Usecase

Drain the BGP link & bring it back

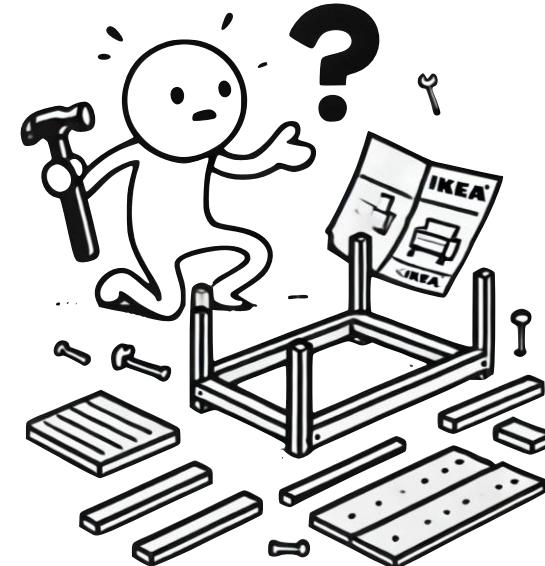
- **Script 1:** Select the link to be drained in script 2 ->
- **Script 2:** Perform the BGP link drain ->
- **Script 3:** Shut down the link
- -- Perform the maintenance --
- **Script 4:** Bring the link back to working state.



LVL 3 - 3rd party

Netbox integrated tools

- Pay some lovely people and get data into netbox.
- Probably the fastest way to go
- If you need some custom business logic magic then there will be some assembly required.



no AUTH or RBAC*

LVL 4 - DAGs



dagster



PREFECT



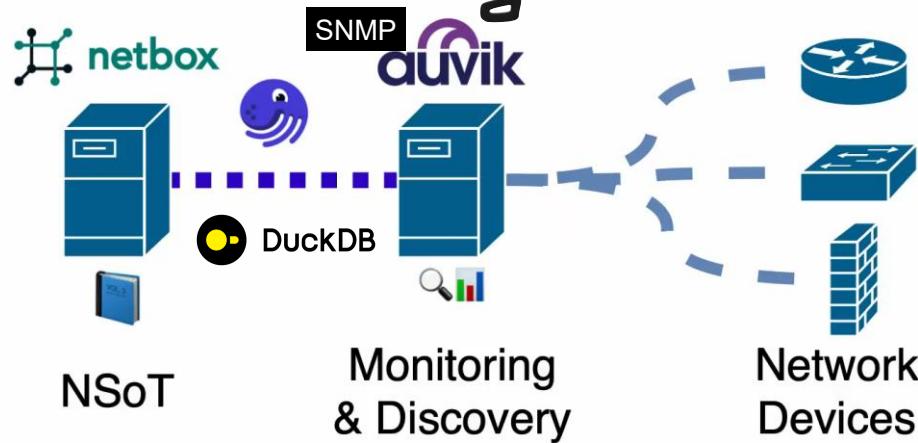
Apache
Airflow



Temporal

Push data into netbox.

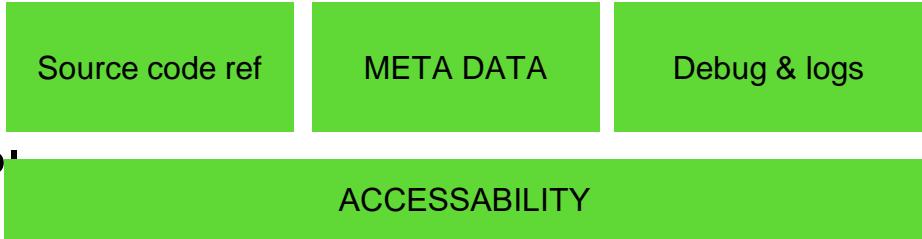
- Sync Devices into netbox
- Data we care about
 - Name “**router01.example.com**”
 - Serial “**ABCD1234**”
 - software versions “**LALA.69**”
 - Hardware type “**ISR-IOU123**”
 - Vendor “**DISCO**”



Daster dagster

DAG workflow

- Download JSON from Auvik API
 - Apply business logic processing.
 - **CHECKS** on every asset!
 - Push changes into netbox.



ACCESSABILITY



We do basic SQL

And Data CHECKS!

Example:

No sites without tenants!



DuckDB

JSON -> SQL

```
def netbox_sites(database: DuckDBResource):
    """
    The raw netbox sites dataset, loaded into a DuckDB database
    """
    sql_query = """--sql
CREATE OR REPLACE TABLE netbox_sites AS
SELECT
    site.id,
    slug AS site_name,
    -- why did I have "site" as a type?
    "site" AS type,
    site.tenant->.slug AS tenant,
    site.tenant->.url AS tenant_url,
    site.region->.slug' AS region,
    site.status->.value' AS status,
    site.description AS description,
    site.created AS created,
    site.last_updated AS last_updated,
FROM read_json('data/raw/netbox_sites.json', format='array') site;
    """
```

Read a JSON file into a table:

```
CREATE TABLE todos (userId UBIGINT, id UBIGINT, title VARCHAR, completed BOOLEAN);
COPY todos FROM 'todos.json';
```

netbox_sites
View in Asset Catalog

Description
The raw netbox sites dataset, loaded into a DuckDB database

Latest materialization

Run	Run 4239ba9c
dagster/job	auvlik_to_netbox_job @ a55fc222
dagster/asset	netbox_sites
Timestamp	26 Oct at 15:30
dagster/row_count	640
dagster/column_schema	[Show Table Schema]
tenant_count	138
region_count	82
status_count	2
status_count_by_tenant	status count

DuckDB



SQL -> Dataframe

```
# Get sites with missing tenant
missing_tenant_query = """--sql
SELECT site_name
FROM netbox_sites
WHERE tenant IS NULL
"""
missing_tenant = conn.execute(missing_tenant_query).df()
```



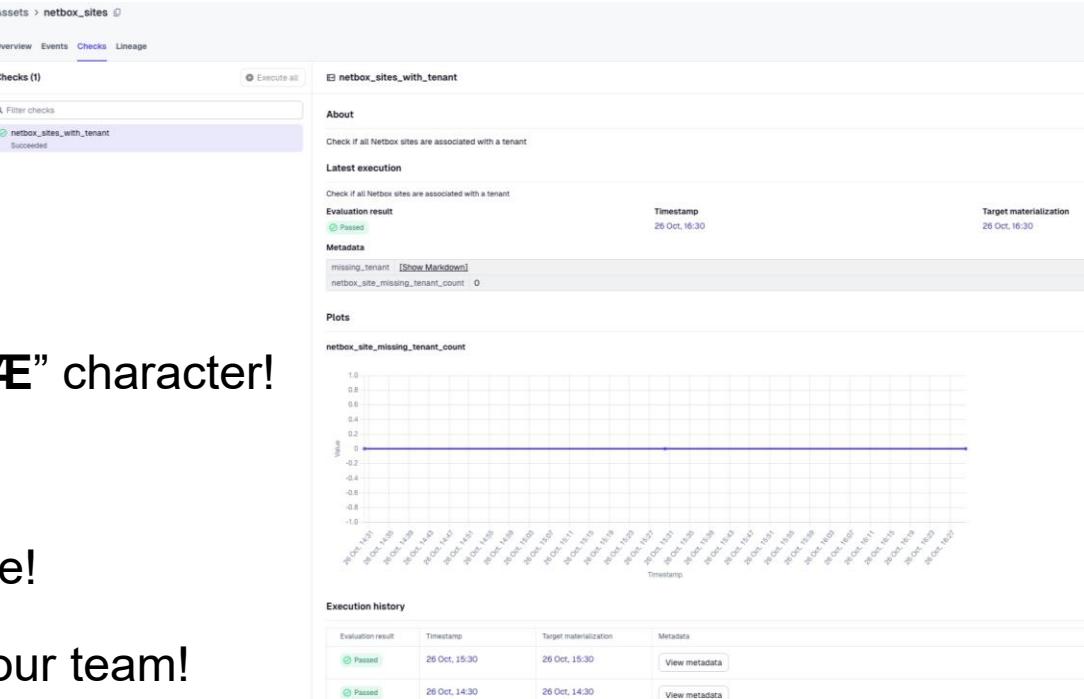
“Data hygiene” Checks!

```
yield AssetCheckResult(
    description="Check if all Netbox sites are associated with a tenant",
    passed=bool(len(missing_tenant) == 0),
    severity=AssetCheckSeverity.WARN,
    metadata={
        "missing_tenant": MetadataValue.md(missing_tenant.to_markdown()),
        "netbox_site_missing_tenant_count": MetadataValue.int(len(missing_tenant))
    }
)
```

Checks!

Lots of them please!

- Encode your business logic
 - Example:
No Device hostnames with “Æ” character!
 - No “null” Serial numbers!
- Ensure quality and data hygiene!
- Accessible to all members of your team!



```
yield AssetCheckResult(  
    description="Check if all Netbox sites are associated with a tenant",  
    passed=bool(len(missing_tenant) == 0),  
    severity=AssetCheckSeverity.WARN,  
    metadata={  
        "missing_tenant": MetadataValue.md(missing_tenant.to_markdown()),  
        "netbox_site_missing_tenant_count": MetadataValue.int(len(missing_tenant))  
    }  
)
```



DuckDB

I love duckDB

-
- Bye Bye **SQLite**

DuckDB: In-Process Analytical DBMS

- What is it for?
 - Data Science
 - Within Analytical Tools - Python/R
 - Data Visualizations and Reporting Tools
 - IoT and Sensor Data
 - BI and Analytics Tools
- Advantages:
 - Resource Sharing
 - Fast Data Transfer

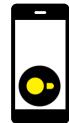
Pipeline Component



Interactive Analysis



“Creative” Architecture



DuckDB combines the best of databases and dataframes

	Dataframes	Client-Server Databases	SQLite	DuckDB
Fast analytical queries	Yes	Yes	No	Yes
Fast data transfer	Yes	No	No	Yes
Easy to use	Yes	No	Yes	Yes
Data Science Integrations	Yes	No	No	Yes
Query Optimization	Some	Yes	Yes	Yes
Built-in Storage	No	Yes	Yes	Yes
Larger than RAM Execution	No	Yes	Yes	Yes
Relational API	Yes	No	No	Yes
SQL Support	No	Yes	Yes	Yes

Go watch this talk later →



PostgreSQL

It's not a replacement for PostgreSQL

Learn more @ FREE!

- Fun course! Very short!
- Play with 
- Dagster  dagster
- DuckDB  DuckDB
- Pandas 
- Some taxi data!



The screenshot shows a course page for 'Dagster Essentials' on 'dagster university'. The page has a dark green header with the dagster logo and the text 'dagster university'. Below the header, the title 'Dagster Essentials' is displayed in large white text. Underneath the title, there's a section for 'Dagster Essentials' with a star rating of '4.8 (64 reviews)'. A brief description follows: 'In this course, learn the basics of Dagster, including how to represent a data pipeline as the data assets it produces and orchestrate a...'. At the bottom of the screenshot, there's a QR code.

Lesson 1: Introduction

Lesson 2: Prerequisites & setup

Lesson 3: Software-defined Assets

Lesson 4: Asset dependencies

Lesson 5: Definitions and code locations

Lesson 6: Resources



<https://courses.dagster.io/courses/dagster-essentials>

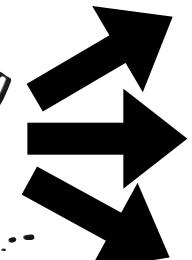
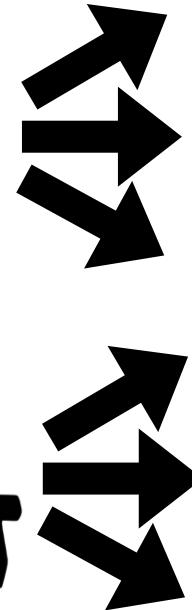
NSoT Constraints



There are always tradeoffs



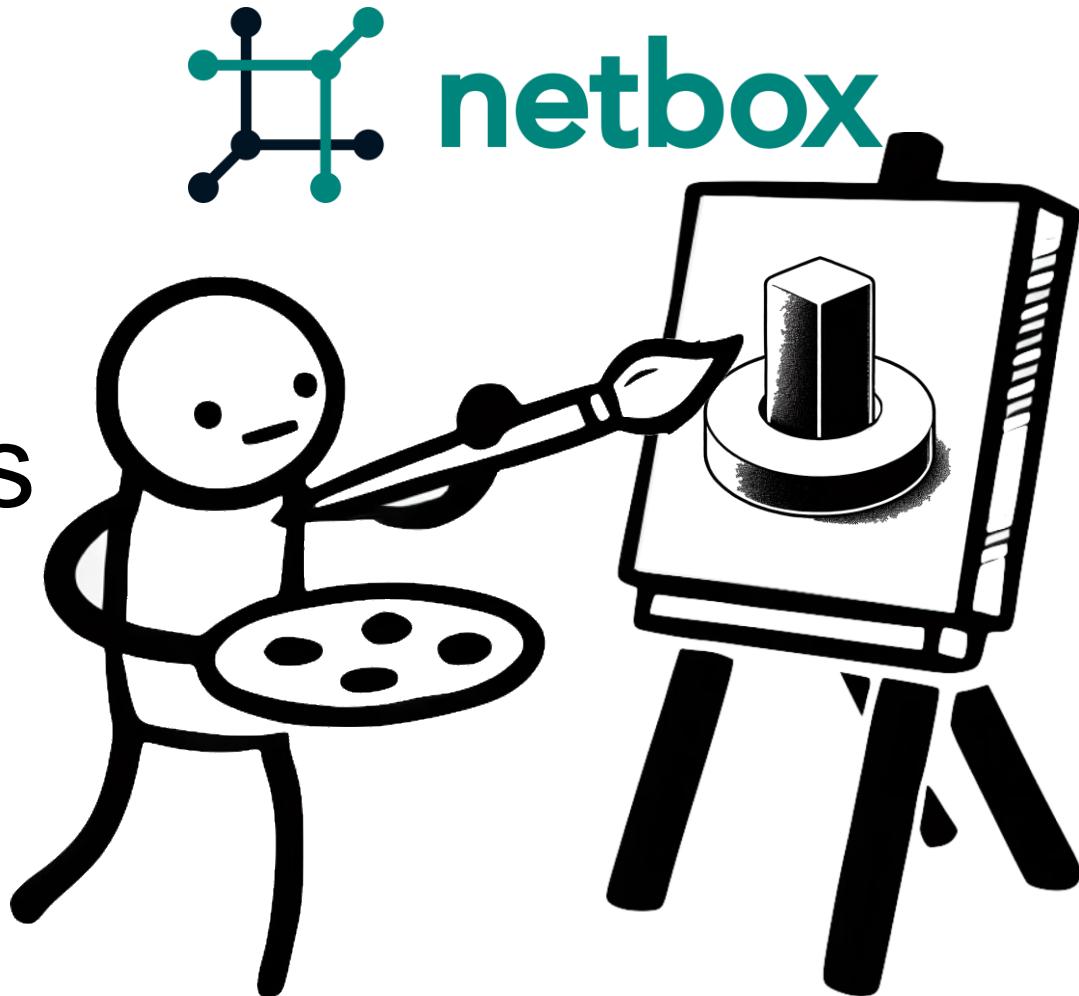
- Modeling your environment
- Level of detail
- Level of complexity
- Level of customization
- There's always a trade off



```
"custom_fields": {  
    "auvik_url": null,  
    "blobs_of_data": {  
        "yo-dawg": [  
            "I heard you like JSON, so I put some JSON  
            in your JSON so you can JSON while you JSON."  
        ],  
        "JSON everywhere": {  
            "yo-dawg": "Yo dawg!",  
            "JSON everywhere": "Yo dawg!",  
            "that's a lot of JSON": "Yo dawg!"  
        },  
        "that's a lot of JSON": "Yo dawg!"  
    }  
}
```

There are limits

```
{  
    "custom_fields": {  
        "auvik_url": null,  
        "blobs_of_data": {  
            "yo-dawg": [  
                "I heard you like JSON, so I put some  
                JSON in your JSON so you can JSON while you JSON."  
            ],  
            "JSON everywhere": {  
                "yo-dawg": "Yo dawg!",  
                "JSON everywhere": "Yo dawg!",  
                "that's a lot of JSON": "Yo dawg!"  
            },  
            "that's a lot of JSON": "Yo dawg!"  
        }  
    }  
}
```



The limitations

Solved by buckets of Custom fields of JSON...

- Blobs of Custom fields JSON everywhere...
- Service Catalogue*
- VRFs for MPLS VPNv4/VPNv6 & EVPN-VXLAN
- L4-L7 firewall*
- Device OS/firmware & Version = Platform adventure*

Add a new custom field

Create

Custom Field

Object types* DCIM > Device

Name* just_a_little_json

Label oh-god

Description do-not-worry

Type* JSON

This will be displayed as help text for the form field. Markdown is supported.

Required This field is required when creating new objects or editing an existing object.

Must be unique The value of this field must be unique for the assigned object

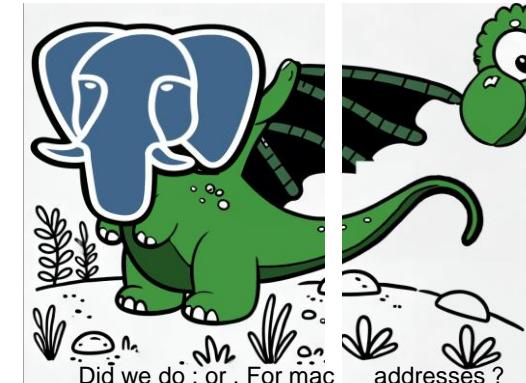
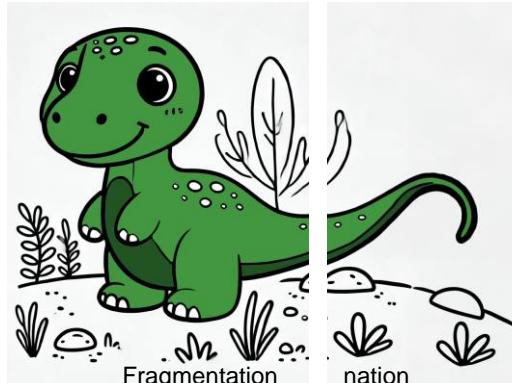


“and so everyone does their own thing to deal with”

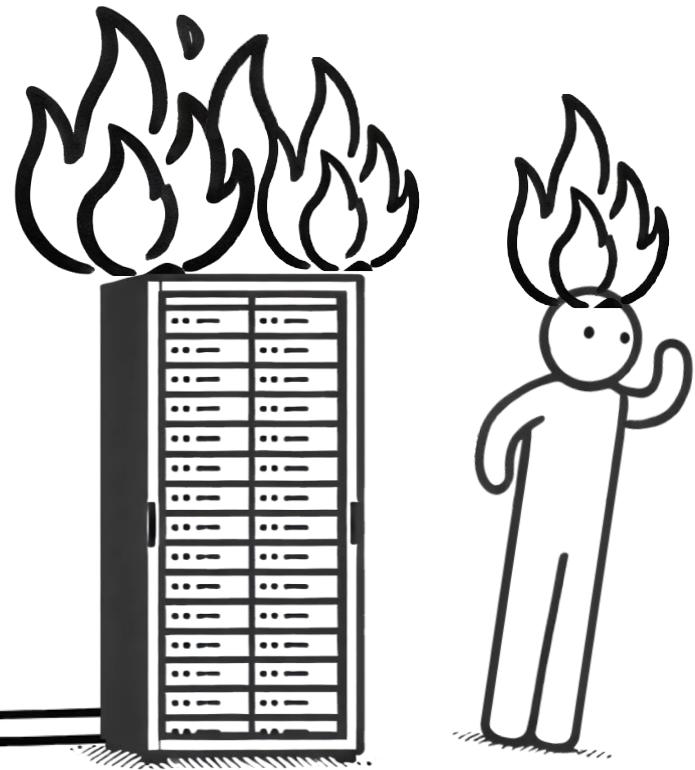
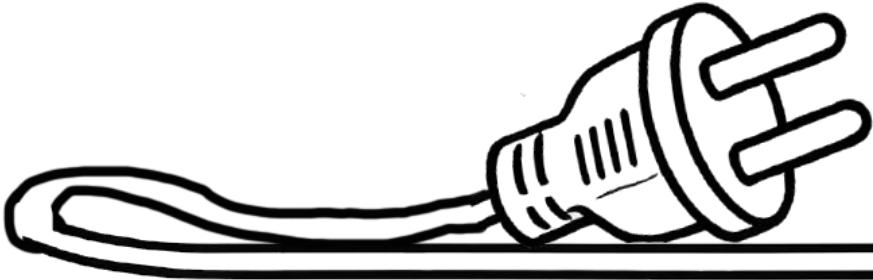


Extending Core Model limitation

You can do all kinds of stuff!

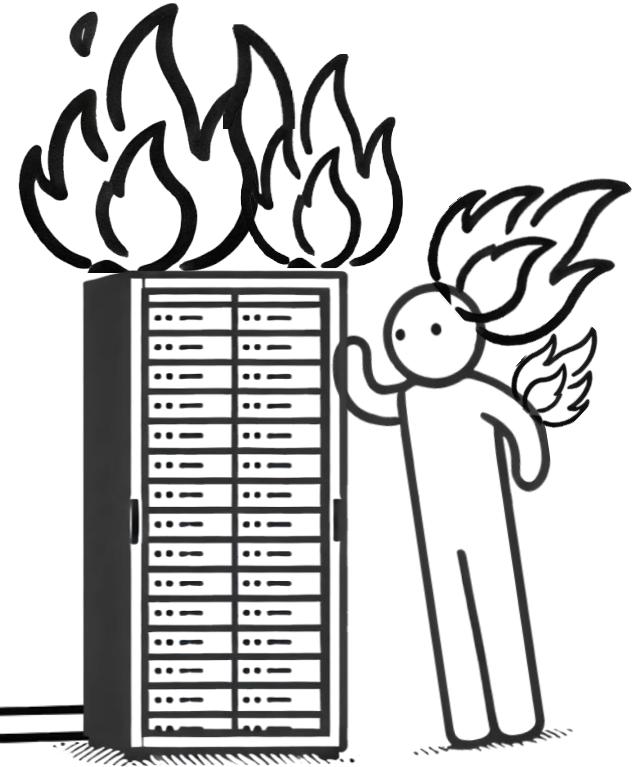


The real question is



How can I get emoji support

⁉️ 🍔🦅 US



Emoji support?



Is Emoji input



Change Log

ID	TIME	USERNAME	FULL NAME	ACTION	TYPE	OBJECT	REQUEST ID
49264	2024-10-16 15:32	mce-steinnb	Steinn Bjarnarson	Updated	Tag	🇳🇴	503faaa6-62dd-4b12-bcb1-f541f7822fc9
49263	2024-10-16 15:31	mce-steinnb	Steinn Bjarnarson	Created	Tag	🇳🇴	a974ffd2-b50a-452f-a829-16ad74db37f7

Emoji Search?



netbox Community

Search...

mce-steinnb Admin

Search

Results

Search Object type(s) All Objects

Lookup Partial match

Search

TYPE	OBJECT	FIELD	VALUE	ATTRIBUTES
— No results found —				

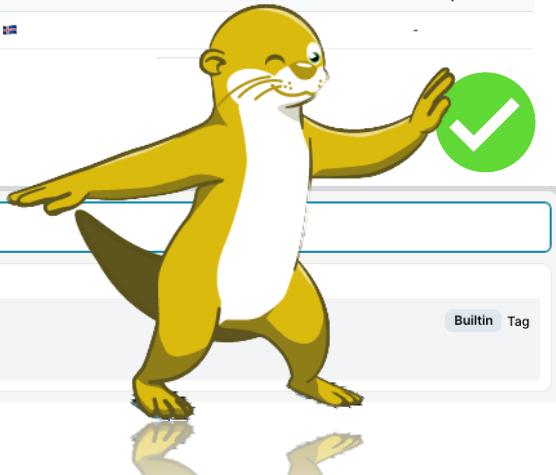
Tag 1

Standard Tag object to attached to other objects to provide some context.

Search an object Filters: 0

Name Description

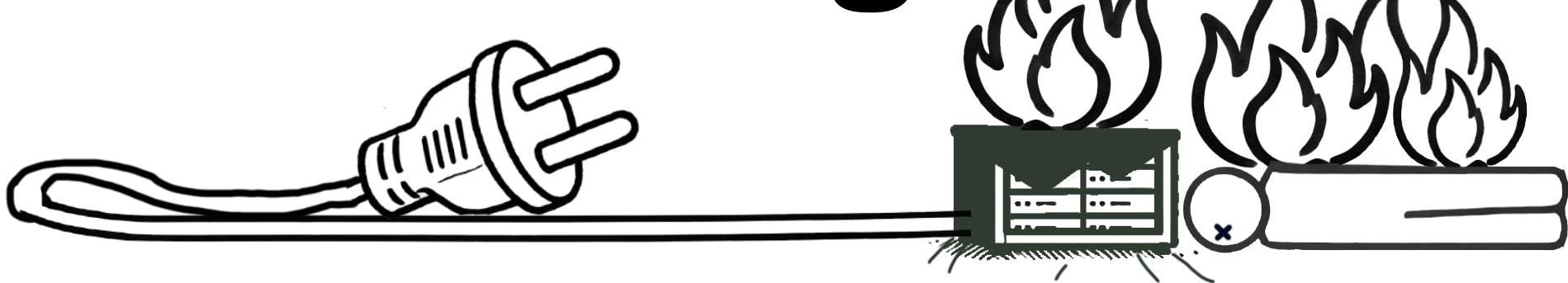
test



Search results for "🇳🇴"

test Description Tag

Proactive monitoring



Resource Monitoring

Grafana adventure!



- Keep an eye on resources being used in your network
- Get an alert before your resource pool is empty
- Keep everyone in your org happy!

Netbox Community

VLAN Groups

Results 52 Filters

NAME	SCOPE TYPE	SCOPE	VLANS	UTILIZATION
Vlan group n+1	—	—	10	10.0%
Vlan group n+1	—	—	10	10.0%
Vlan group n+1	—	—	15	15.0%
Vlan group n+1	—	—	6	3.0%
Vlan group n+1	—	—	194	77.6%
Vlan group n+1	—	—	91	91.0%
Vlan group n+1	—	—	36	72.0%
Vlan group n+1	—	—	32	42.7%
Vlan group n+1	—	—	30	60.0%

no Sort

Netbox Community

Prefixes

Results 33 Filters 1

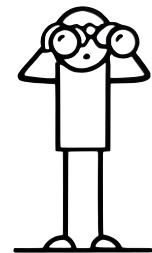
PREFIX	STATUS	CHILDREN	VRF	UTILIZATION
1.1.1.0/24	Container	3	Global	10.2%
1.1.1.0/24	Container	84	Global	32.4%
1.1.1.0/24	Container	16	Global	6.3%
1.1.1.0/24	Container	123	Global	26.0%
1.1.1.0/24	Container	168	Global	76.8%
1.1.1.0/24	Container	796	Global	41.7%
1.1.1.0/24	Container	14	Global	87.5%

no Sort

*Netbox version v4.1.4

VLAN usage monitoring

Quick and dirty Grafana



A [checkbox.com/api](#)

Parser Try backend Default Source URL Format Table Help GitHub

Type JSON Method GET URL <https://checkbox.xxxx.ai/api/vlan-groups/> Headers, Request params

- Parsing options & result fields Field types, alias and selectors

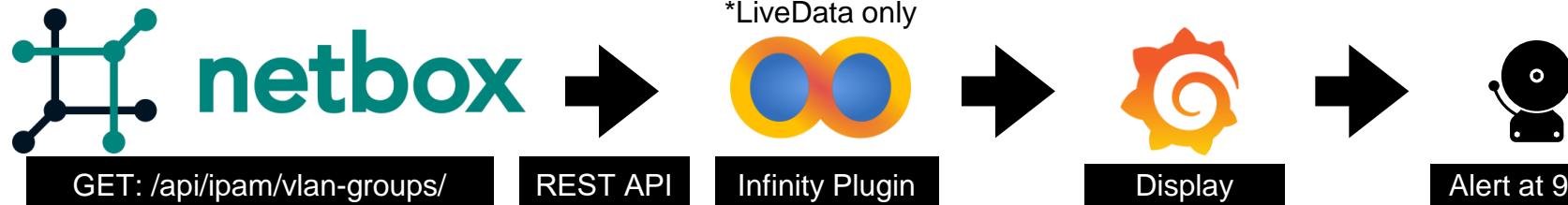
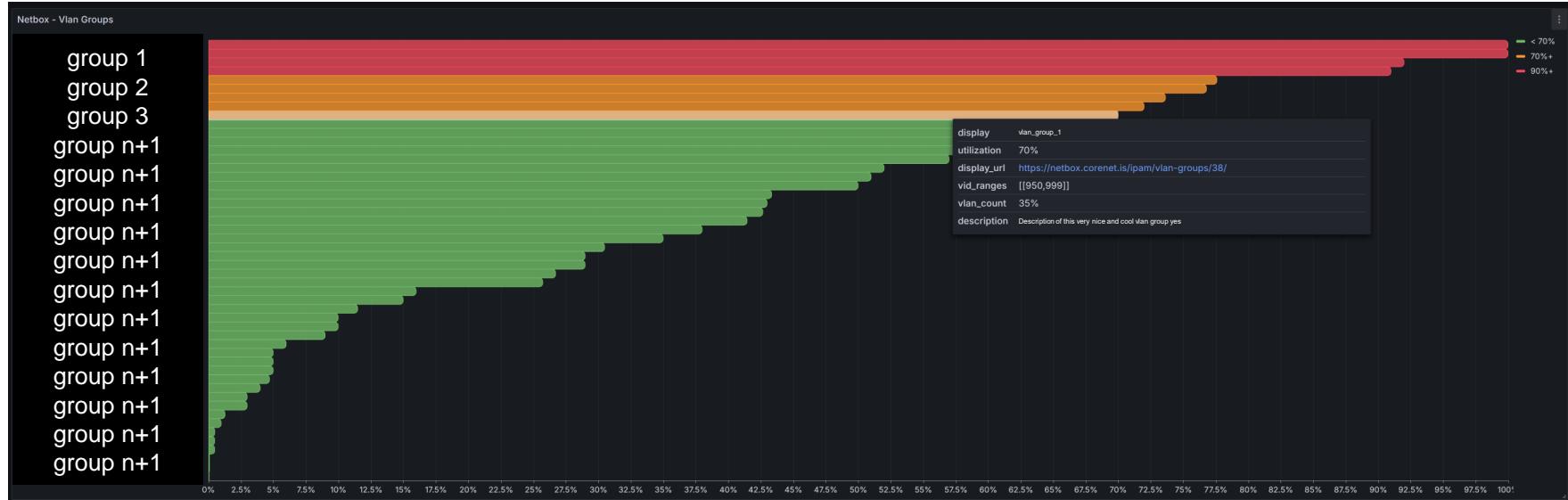
Rows/Root - optional
Rows/Root selector (optional)

Advanced Options - optional
Root returns object instead array? Is date in columnar format?

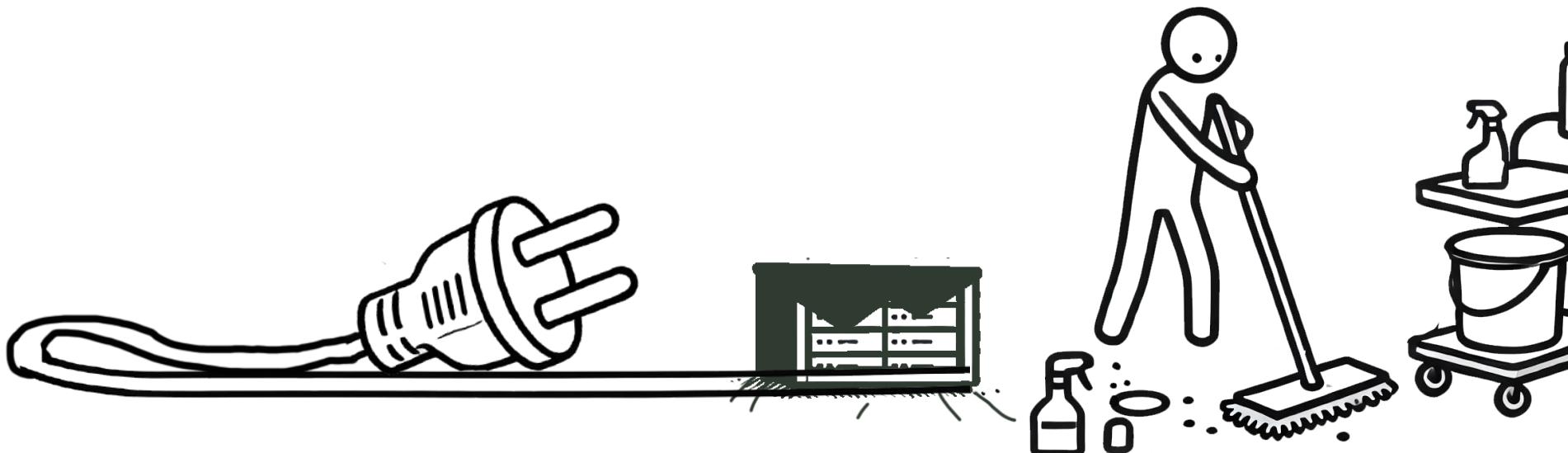
Columns - optional

Selector	display	as	Title	format as	String
Selector	utilization	as	Title	format as	Number
Selector	display_url	as	Title	format as	String
Selector	vlan_ranges	as	Title	format as	String
Selector	vlan_count	as	Title	format as	String
Selector	description	as	Title	format as	String

Add Column



Public service announcements.



Regex

Brief summary of typical disasters:

- Parsing email or IP addresses
- Replacing build time tokens
- Rolling your own string interpolations
- Validating a person's name
- Parsing a phone number
- Validating a street number
- Enforcing a zip code globally
- Searching in a database for something in a live query
- Parsing a date or a time
- Basically almost always outside of command line or ad-hoc query fu



Matt Fuller @matthewdfuller

...

We finally have a price tag to the breach. \$80 million because of a single extra **#AWS IAM permission**. Probably one of the most expensive "*" wildcards in history. **#cloud #security**

 The Washington Post @washingtonpost · 6 Aug 2020

Capital One to pay \$80 million fine for massive hack of customers' credit card applications and other sensitive data wapo.st/3aOegIS

License

No regex licenses have been issued at this time.



<https://regexlicensing.org/about/>

Book recommendation

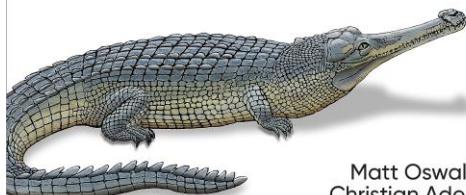


O'REILLY®

Network Programmability & Automation

Skills for the Next-Generation
Network Engineer

Second
Edition

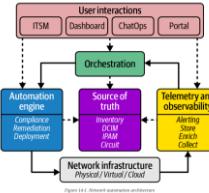


Matt Oswalt,
Christian Adell,
Scott S. Lowe &
Jason Edelman

Chapter 14. Network Automation Architecture

TIP

Because so many tools are available, choosing one over another could be overwhelming. But don't get blocked by the *premature optimization* trap, spending too much time now on solving a future issue (that may not even happen). Using the proposed architecture, you will be able to replace any tool with a new one when the necessity arises.



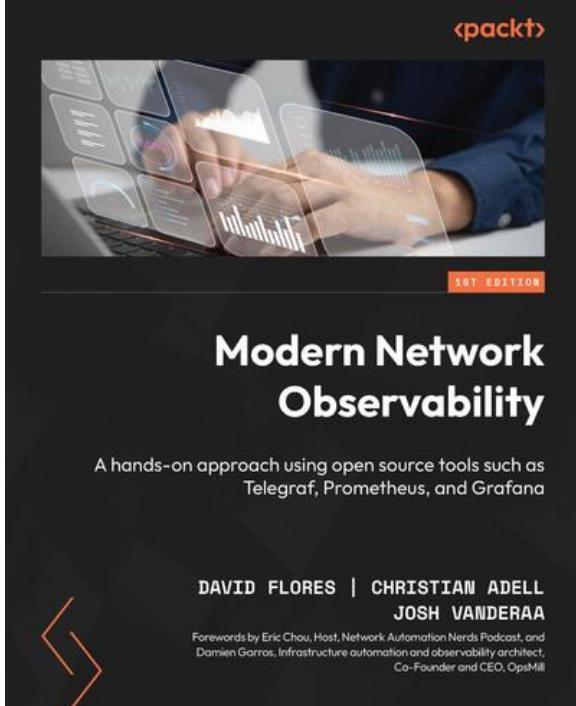
O'REILLY®

Safari

Read it on Safari online* Ch 14 and thumb
the rest! DO NOT READ COVER 2
COVER



Book recommendation



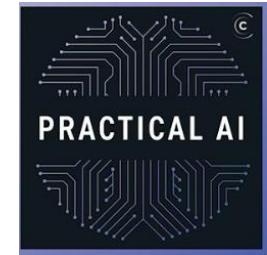
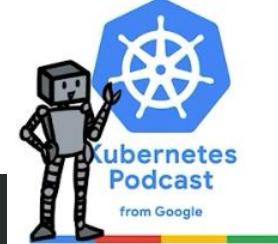
Modern Network Observability

I'm reading this on the plane.

O'REILLY®
Safari

Read it on Safari online

Podcasts



<https://rule11.tech/>

Network Automation Landscape

- Overview of everything*
- Codes on GIT, Pull requests welcome!
- Handing it over to NAF soon*



<https://steinzi.com>

SEARCH

EXPLORE GUIDE STATS

Filters GROUP Projects and products VIEW MODE: Grid Card ZOOM: - +

Network State Testing, Compliance & Querying Network Source of Truth

Network Discovery/Accuracy

Monitoring & Alerting Observability

Telemetry and observability

Telemetry & Data Collection

Automation Platforms and Management Continuous Configuration Automation (CCA)

Network Virtualization Platform

Virtualization

Software Libraries Network Programmability Frameworks

Python Ecosystem

Network Automation Landscape

The Network Automation Landscape is a comprehensive guide to the tools and platforms used in network automation. It is organized into several categories:

- Network State:** Testing, Compliance & Querying, Network Source of Truth.
- Network Discovery/Accuracy:** Network Discovery/Accuracy.
- Monitoring & Alerting:** Monitoring & Alerting, Observability.
- Telemetry and observability:** Telemetry and observability.
- Telemetry & Data Collection:** Telemetry & Data Collection.
- Automation Platforms and Management:** Continuous Configuration Automation (CCA), Automation Platforms and Management.
- Network Virtualization Platform:** Network Virtualization Platform.
- Virtualization:** Virtualization.
- Software Libraries:** Software Libraries, Network Programmability Frameworks.
- Python Ecosystem:** Python Ecosystem.

Each category contains a grid of icons representing different tools, along with their names and licensing information (e.g., Full Open Source, Premium, Limited). The tools listed include:

- Testing, Compliance & Querying: pyATS, netbox, INFRAHUB, Bluebeam, RANCID.
- Network Source of Truth: netbox, INFRAHUB, Bluebeam, RANCID.
- Network Discovery/Accuracy: Stulpit, FORWARD, netBrain, IP FABRIC, netOS fabric.
- Monitoring & Alerting: Icinga, LibreNMS, ZABBIX, netOS.netops, Airflow, Datadog, kentik.
- Observability: Telegraf, gNMIc, Temporal, BACKBOX.
- Telemetry and observability: RB, Icinga, LibreNMS, ZABBIX, netOS.netops, Airflow, Datadog, kentik.
- Telemetry & Data Collection: Stack, Splunk, Grafana.
- Automation Platforms and Management: AWX, iConfig, NSO, gluware, zpe, intential, BACKBOX, Ansible, SALT, Terraform, Pulumi.
- Continuous Configuration Automation (CCA): Ansible, SALT, Terraform, Pulumi.
- Network Virtualization Platform: GNS3, CONTAINERLAB, NETBOX, EVE.
- Virtualization: puppet, Pliant, spacelift, Temporal, RUNDECK, torero.
- Software Libraries: netmiko, pyeapi, ttp, DiffSync, Netconf Diff, PyangBind.
- Network Programmability Frameworks: firecracker, Open vSwitch.



DECEMBER 20, 1985 – JUNE 10, 2024

In memory of Nicholas Joseph Russo

Respected Member of the Network Industry & Network Automation Community

He was the smartest, hardest-working man I've ever known; An inspiration.



<http://web.archive.org/web/20240807144237/https://njrusmc.net/>



<https://www.dignitymemorial.com/obituaries/bel-air-md/nicholas-russo-11854721>



<https://www.pluralsight.com/courses/generative-ai-network-automation-enhancing>