Opensource network monitoring with NetBox and Telegraf

Agenda

- Components of the solution
- End-to-end process
- Demo

Benefits of this solution

- License free
- Horizontal scalability and performance
- Customizability
- Analytics
- Break silos by using same tools as compute team
- Source of truth approach

Components of any network monitoring solution

- Inventory
- Configurations
- Data collector (SNMP,gMNI, scripts)
- Database (time series)
- Visualization

Components of this solution

- Inventory
  - NetBox
- Data collector and configurations
  - Telegraf (SNMP, gNMI, etc.)
  - toml files generated with Python based on NetBox data
- Time series database
  - any modern TSDB (InfluxDB, Prometheus, Mimir…)
- Visualization with Grafana

Telegraf

- Lightweight and powerful data collection agent by Influxdata
- Plugin architecture
  - 255 input plugins
    - SNMP/traps, gNMI, JTI, Netflow, sFlow, Cisco MDT
  - 59 output plugins
    - Influxdb, Prometheus, SQL, HTTP etc
  - 29 processor plugins

Telegraf config file

```
[inputs]
[[inputs.snmp]]
name = "device_uptime"
agents = [ "10.10.10.1",]
retries = 2
interval = "60s"
sec_name = "${SNMP_SEC_NAME}"
sec_level = "authPriv"
auth_password = "${SNMP_AUTH_PASSWORD}"
auth_protocol = "MD5"
priv_password = "${SNMP_PRIV_PASSWORD}"
priv_protocol = "AES"
max_repetitions = 127
[[inputs.snmp.field]]
oid = "1.3.6.1.6.3.10.2.1.3.0"
name = "uptime"

[[inputs.snmp.table]]
name = "device_system_metrics"
index_as_tag = true
inherit_tags = [ "hostname",]
[[inputs.snmp.table.field]]
oid = "1.3.6.1.4.1.2636.3.1.13.1.5"
name = "sensorName"
is_tag = true
```

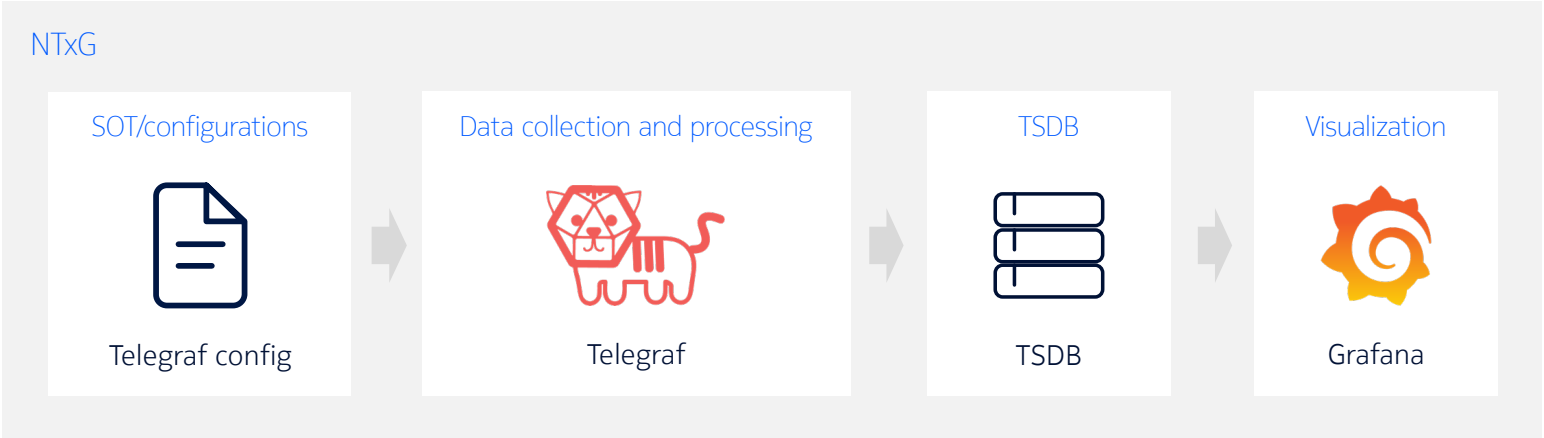
Telegraf

NTxG

SOT/configurations	Data collection and processing	TSDB	Visualization
Telegraf config	Telegraf	TSDB	Grafana

Telegraf

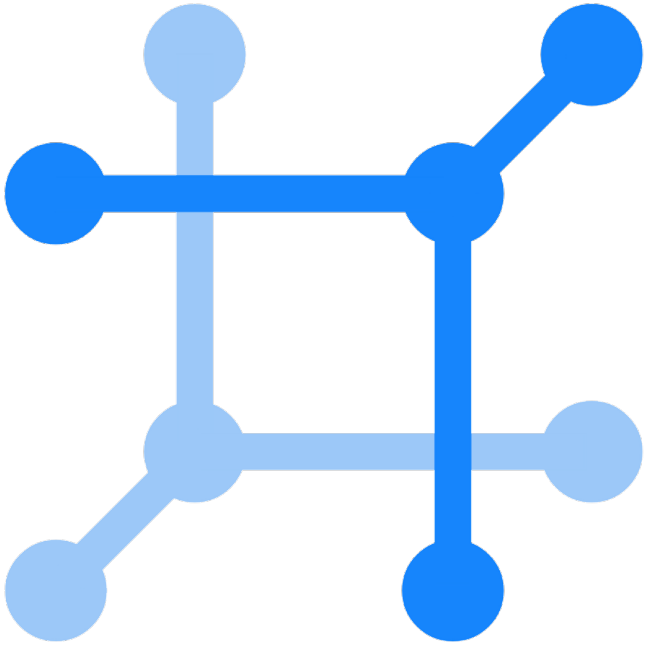
- Difficult to manage configuration files manually
- Different configs for
  - Vendors
  - Platforms
  - Device roles



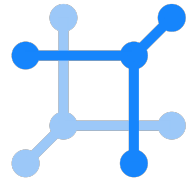
How to manage Telegraf configs

NetBox

- Open source DCIM/IPAM tool
- Documentation of devices which should be monitored
- Data models for
  - Sites
  - Racks
  - Devices (types, roles)
  - Device platforms/OS version
  - IPs
  - Contacts/tenancy

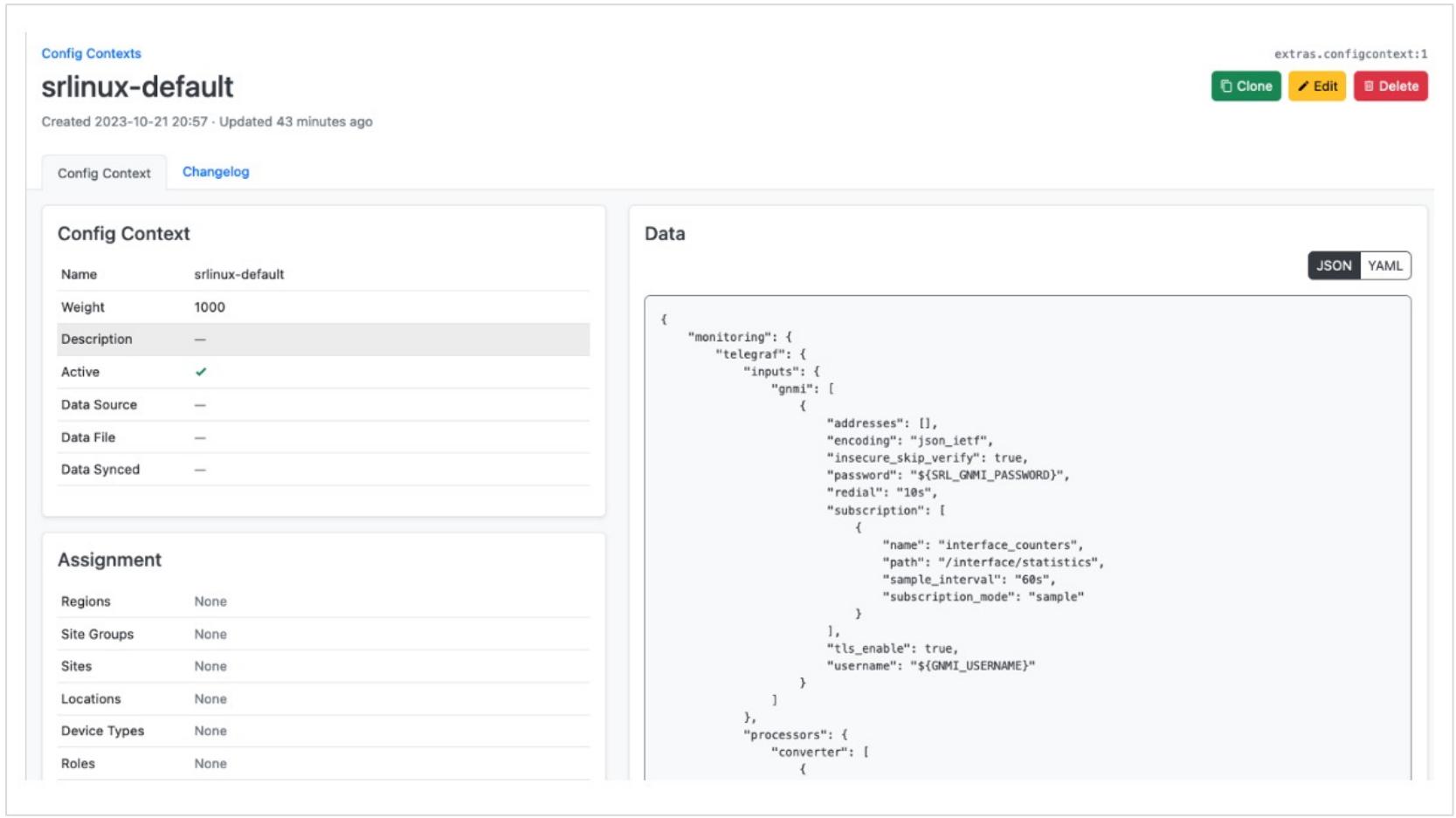


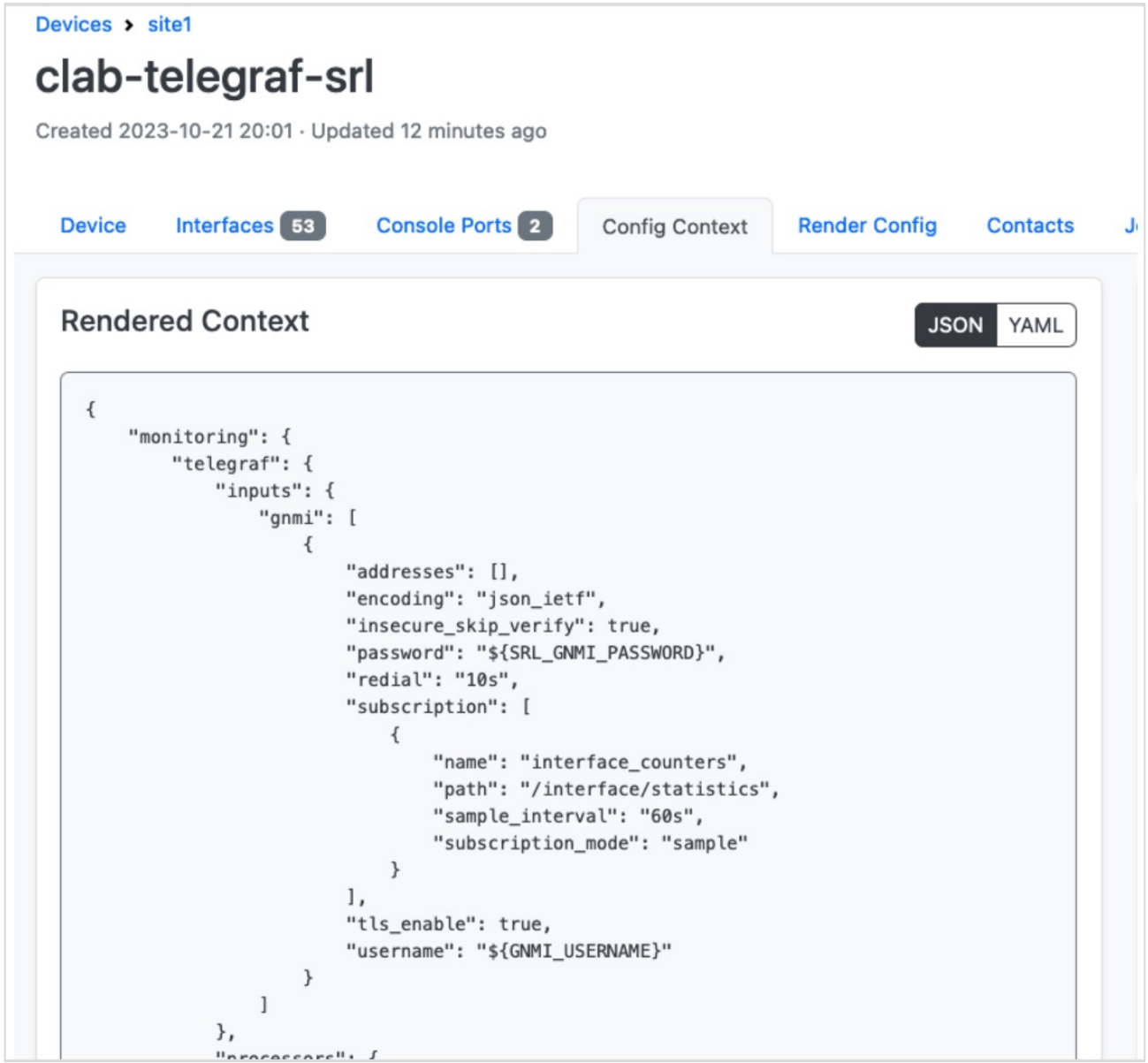
NetBox key features for this solution



- Data models
  - Devices, device models, device roles, sites, etc.
- Config contexts
  - Arbitrary JSON data which applies to device
  - Telegraf configuration template as ‘Config Context’
- Webhooks
  - Notifications about changes
- Custom links
  - Links to graphs
- Custom fields
  - For example maintenance breaks schedules

NetBox config context

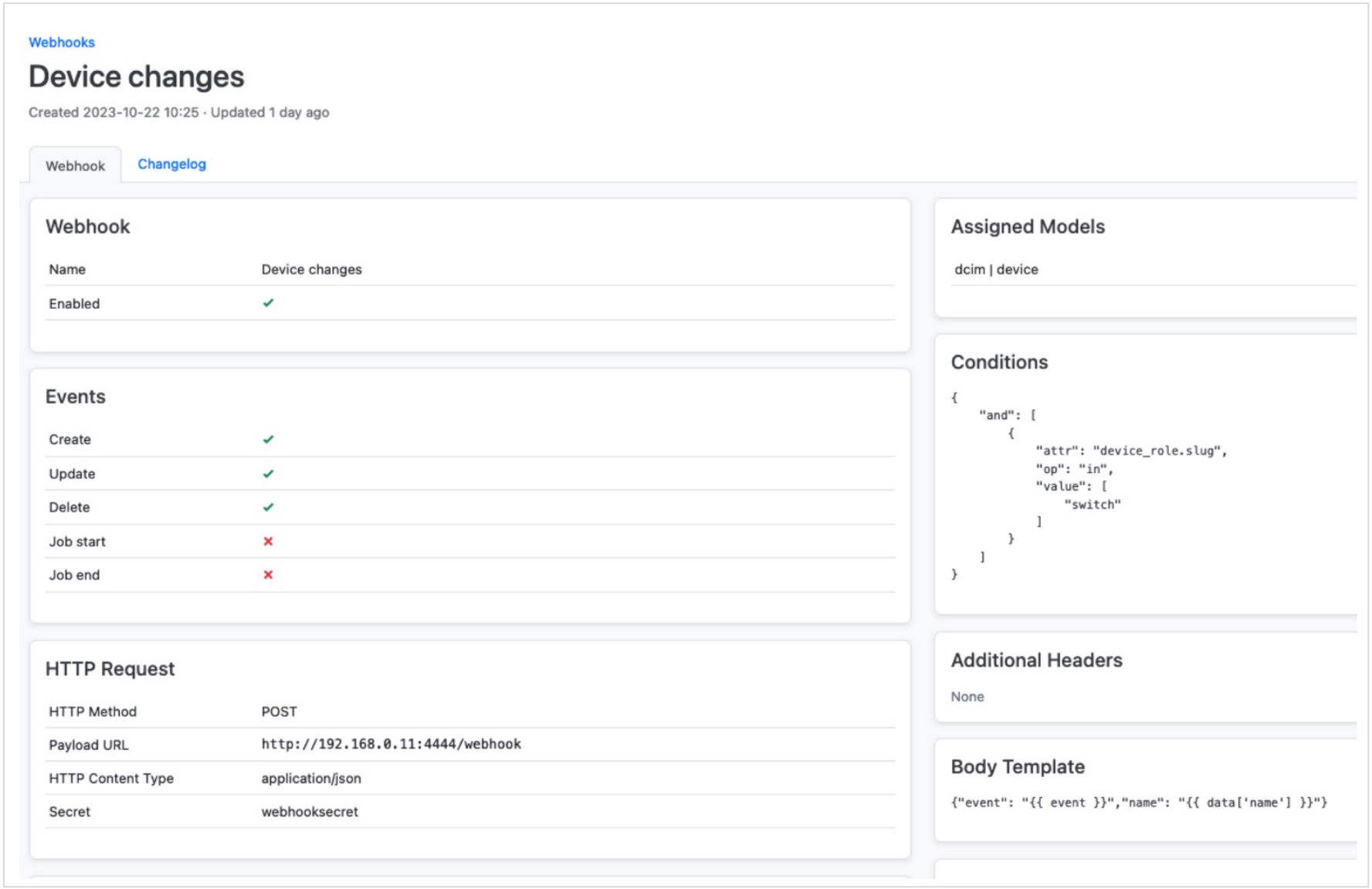



Devices › site1
clab-telegraf-srl
Created 2023-10-21 20:01 · Updated 12 minutes ago
Device
Interfaces 53
Console Ports 2
Config Context
Render Config
Contacts
Rendered Context
JSON
YAML
{
"monitoring": {
"telegraf": {
"inputs": {
"gnmi": [
{
"addresses": [],
"encoding": "json_ietf",
"insecure_skip_verify": true,
"password": "${SRL_GNMI_PASSWORD}",
"redial": "10s",
"subscription": [
{
"name": "interface_counters",
"path": "/interface/statistics",
"sample_interval": "60s",
"subscription_mode": "sample"
}
],
"tls_enable": true,
"username": "${GNMI_USERNAME}"
}
]
},

How to trigger config generation

NetBox Webhooks

- Webhooks are notifications which can be sent from NetBox when something happens
  - For example, device status/configuration change
- Customizable payload

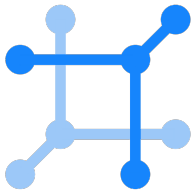

Webhooks
Device changes
Created 2023-10-22 10:25 · Updated 1 day ago
Webhook
Changelog
Webhook
Name
Device changes
Enabled
Events
Create
Update
Delete
Job start
Job end
HTTP Request
HTTP Method
POST
Payload URL
http://192.168.0.11:4444/webhook
HTTP Content Type
application/json
Secret
webhooksecret
Assigned Models
dcim | device
Conditions
{
"and": [
{
"attr": "device_role.slug",
"op": "in",
"value": [
"switch"
]
}
]
}
Additional Headers
None
Body Template
{"event": "{{ event }}","name": "{{ data['name'] }}"}

Config generation script with Webhook listener

- Python script listens for hooks from NetBox
- Webhook listener can be made, for example with Python, FastAPI or Flask
- Renders config
  - Dedicated configuration file for each device
- Optional queuing for the received hooks

NTxG

SOT/configurations



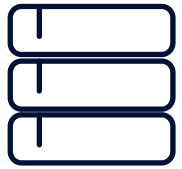
NetBox

Config generation script

Data collection and processing

Telegraf

TSDB



TSDB

Visualization

Grafana

Add GitOps

Why Git?

- Version control
- Repeatability
- Scalability
- Disaster recovery
  - Unintended changes to NetBox
    - NetBox doesn’t have option to roll back changes
  - Issues with NetBox upgrades
  - NetBox unavailability
- CI and testing

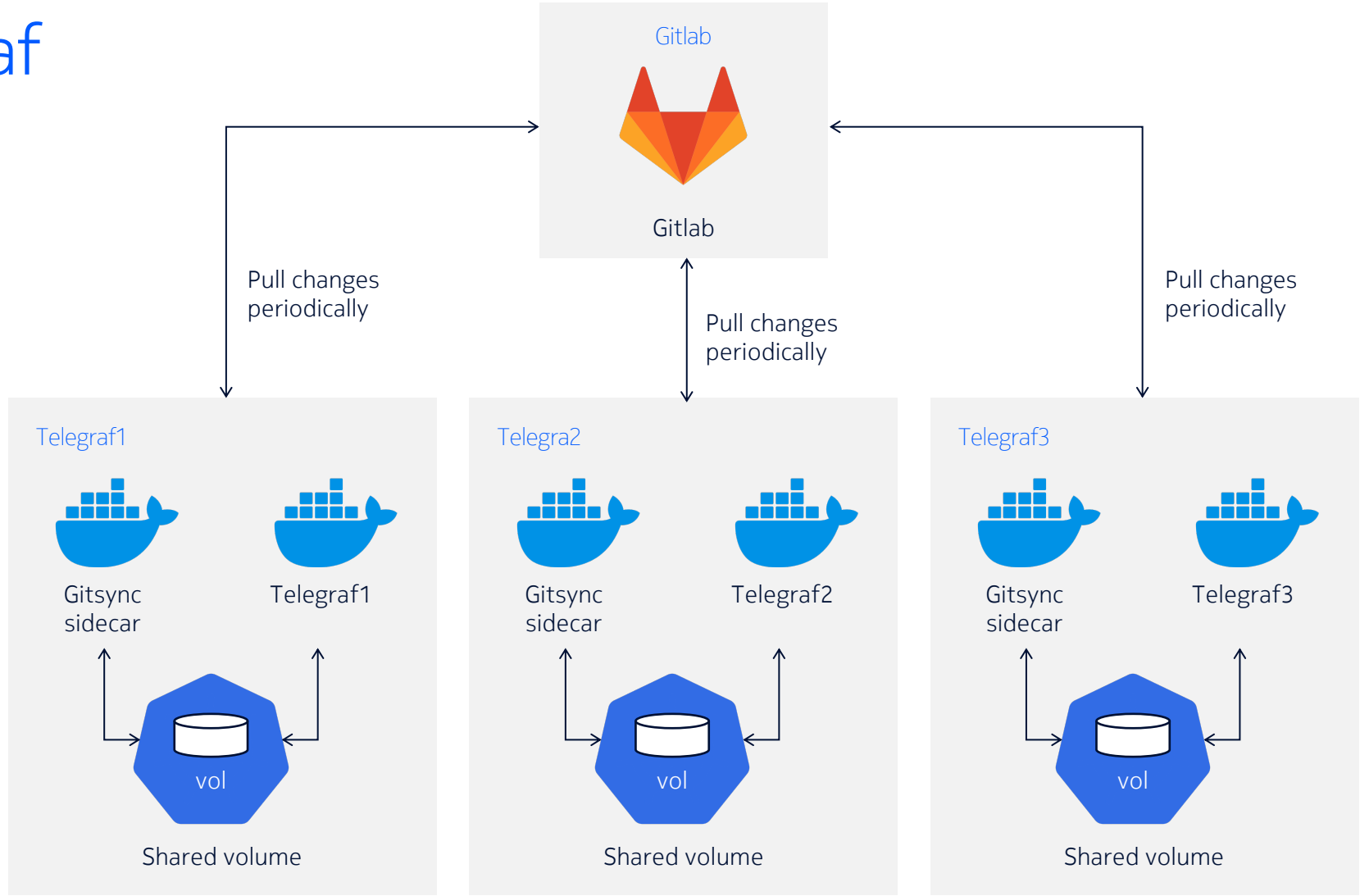
How to deploy Telegraf configs

Gitsync

- Multiple different ways to deploy the configs
  - https://github.com/kubernetes/git-sync
- A sidecar app which clones a git repo and keeps it in sync with the upstream
- Can pull one time, or on a regular interval
- Selection HEAD of a branch, git tag, a specific git hash
- It will only re-pull if the target of the run has changed in the upstream repository

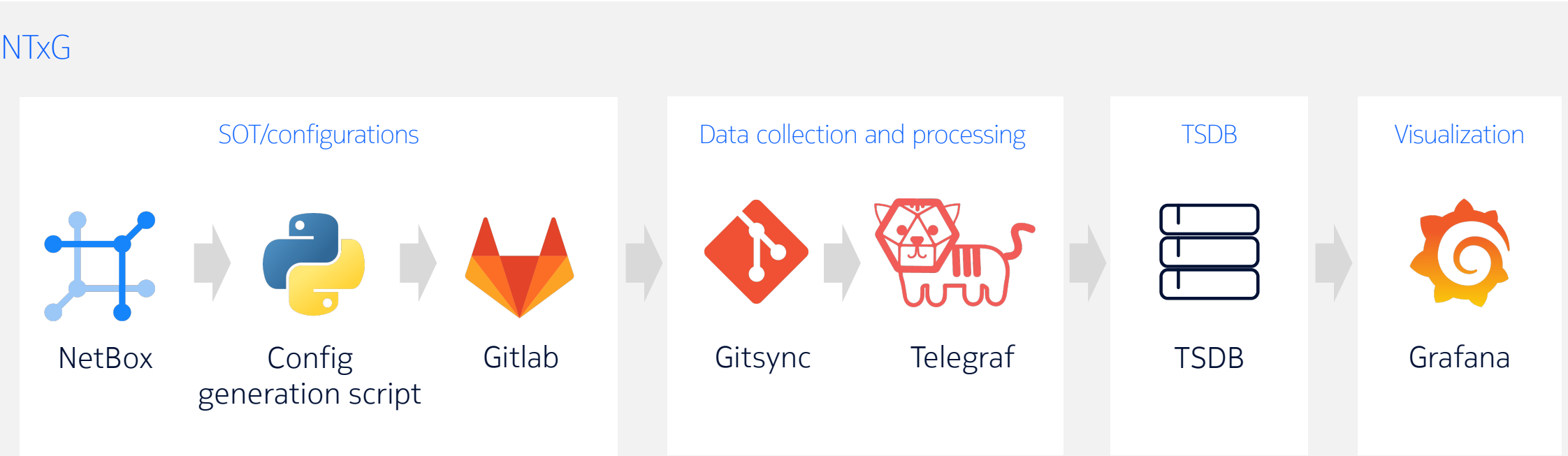
Gitsync and Telegraf

- Shared volume between Gitsync and Telegraf containers



Configs with gitsync

NTxG



Database

- What is the best time series database?
  - The one which is already being used in your organization
  - The one which is managed by someone else
- Telegraf supports currently 59 output plugins
- Open feature request if some DB is missing

Data processing

Data normalization

- Multivendor/platform environment, data is in multiple formats
  - Memory in kB, MB, utilization percentage
  - Uptime in s, ms, boot time epoch
- Normalize data on collection vs visualization/alerting
  - Telegraf vs. Grafana

Converter

- Converters values to strings, integers, etc.

```
[[processors.converter]]
   order = 1
 [processors.converter.fields]
   integer = ["in*", "out*"]
```

Rename

```
[[processors.rename]]
 [[processors.rename.replace]]
   field = "in-octets"
   dest = "in_octets"
```

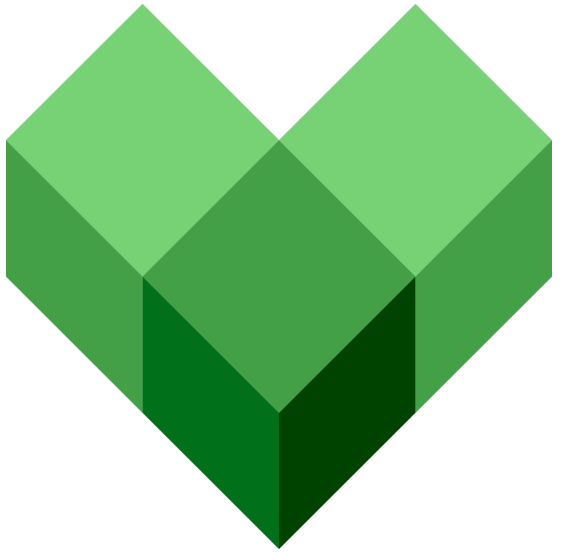
Deduplication

- Remove the metrics which don’t change
  - Reduce the amount of data
  - Especially handy for data collected with SNMP

```
[[processors.dedup]]
  namepass = ["software_version","system_boot_time","platform_temperature","interface_counters","system_hostname"]
  dedup_interval = "600s"
```

Starlark processor

- Dialect of Python
- Capabilities
  - Math operations
  - String operations
  - Renaming tags
  - Logic operations
- Minimal impact on performance
  - In local tests, it takes about 1µs to run a modest script to process one metric



Starlark processor

```
[[processors.starlark]]
 namepass = ["system_memory",]
 source = '''
def apply(metric):
    if "reserved" in metric.fields.keys():
        mem_available = float(metric.fields['reserved'])
        mem_total = float(metric.fields['physical'])
        metric.fields['system_memory_utilization'] = int(mem_available / mem_total) * 100
        for k, v in metric.fields.items():
            if k != 'system_memory_utilization':
                metric.fields.pop(k)
        return metric
    else:
        return None
'''
```

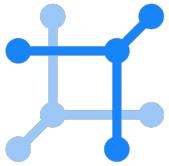
Starlark

```
[[processors.starlark]]
 namepass = ["system_cpus",]
 source = '''
def apply(metric):
    if "cpu/state/idle/instant" in metric.fields.keys():
        cpu_idle = metric.fields["cpu/state/idle/instant"]
        metric.fields["system_cpu_utilization"] = 100 - cpu_idle
        for k, v in metric.fields.items():
            if k != 'system_cpu_utilization':
                metric.fields.pop(k)
        return metric
    else:
        return None
'''
```


NOKIA

NTxG

SOT/configurations



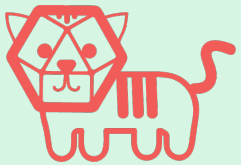
NetBox

Config generation script

Gitlab

Data collection and processing

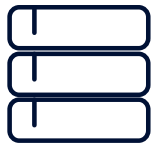
Gitsync



Telegraf

Starlark

TSDB



TSDB

Visualization

Grafana

Links

- NetBox
  - https://docs.netbox.dev/
  - https://github.com/netbox-community/netbox
- Telegraf
  - https://www.influxdata.com/time-series-platform/telegraf/
- Telegraf
  - https://www.influxdata.com/intergration/snmp
  - https://www.influxdata.com/blog/telegraf-best-practices-snmp-plugin/

Links

- Telegraf gNMI plugin
  - https://github.com/influxdata/telegraf/blob/master/plugins/inputs/gnmi/README.md
- Telegraf and Starlark
  - https://www.influxdata.com/blog/how-use-starlark-telegraf/
- Gitsync
  - https://github.com/kubernetes/git-sync

Links

- SR Linux container
  - https://network.developer.nokia.com/dc-fabrics/try-sr-linux/
- Grafana
  - https://garafana.com/oss/grafana
- ContainerLab
  - https://containerlab.dev
- gNMIc
  - https://github.com/openconfig/gnmic

Demo

Demo setup

- Docker containers
- NetBox running on docker-compose
  - https://github.com/netbox-community/netbox-docker
- Containerlab - https://containerlab.dev
  - Switches
  - Webhook listener
  - Telegraf
  - Gitsync
  - Influxdb
  - Grafana