SysEleven

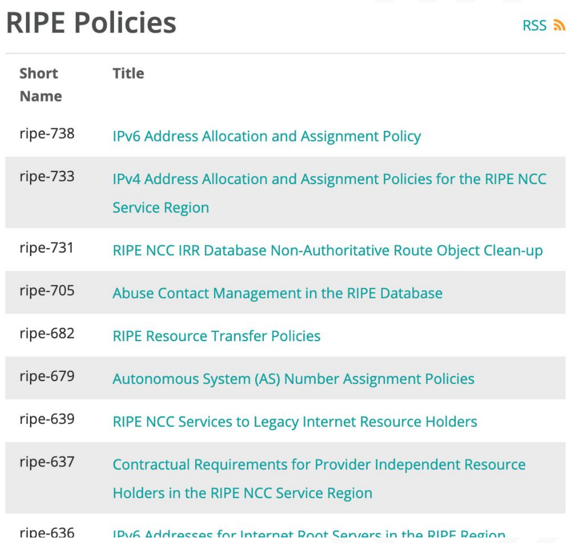
// DENOG13 2021

Automate RIPE DB handling with NetBox

08. November 2021 | Christian Harendt

Status quo

Many prefixes, you are responsible for

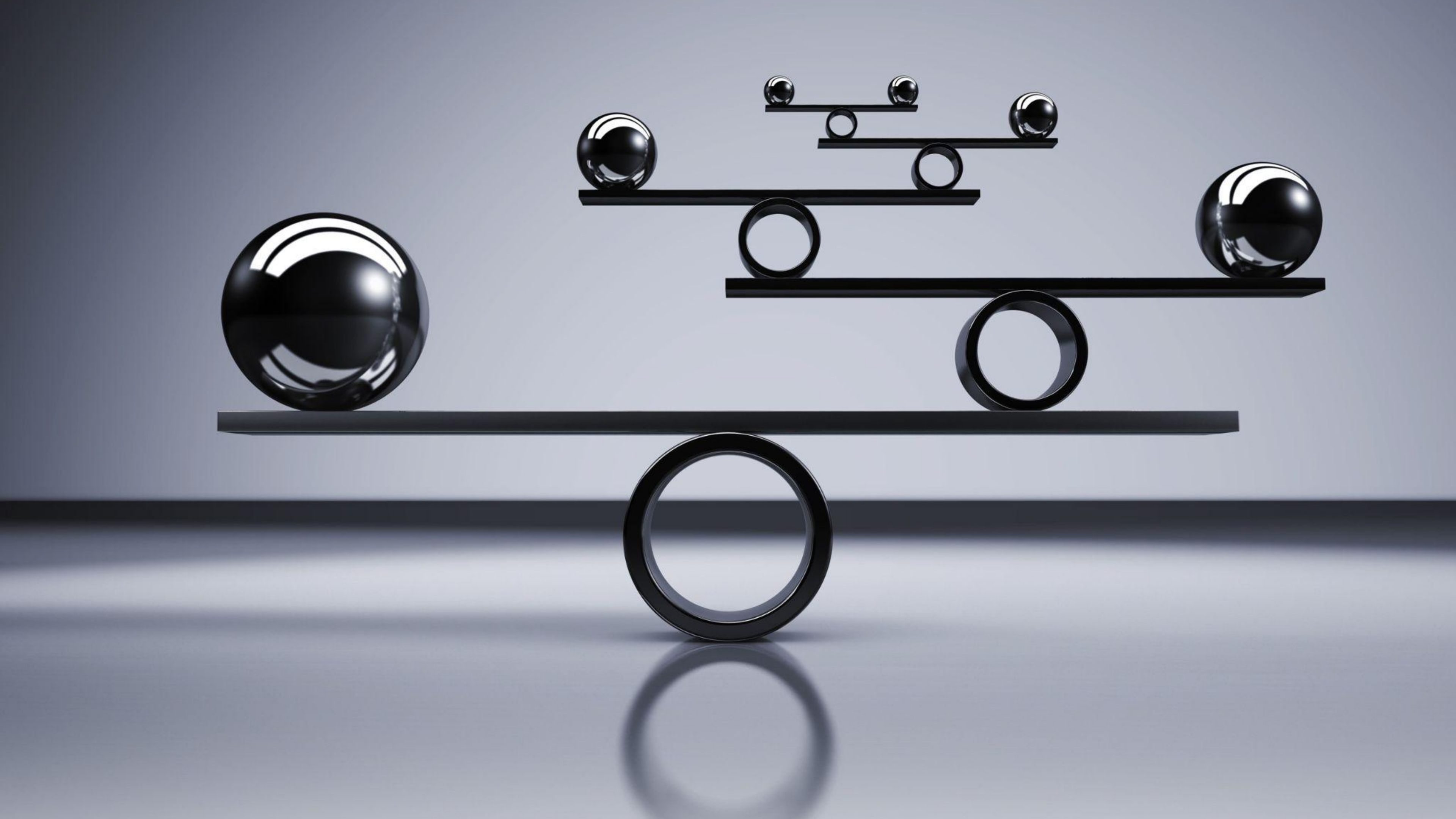


RIPE Policies

RSS

Short Name	Title
ripe-738	IPv6 Address Allocation and Assignment Policy
ripe-733	IPv4 Address Allocation and Assignment Policies for the RIPE NCC Service Region
ripe-731	RIPE NCC IRR Database Non-Authoritative Route Object Clean-up
ripe-705	Abuse Contact Management in the RIPE Database
ripe-682	RIPE Resource Transfer Policies
ripe-679	Autonomous System (AS) Number Assignment Policies
ripe-639	RIPE NCC Services to Legacy Internet Resource Holders
ripe-637	Contractual Requirements for Provider Independent Resource Holders in the RIPE NCC Service Region
ripe-636	IPv6 Addresses for Internet Root Servers in the RIPE Region

RIPE policies to comply with



SysEleven

Current situation

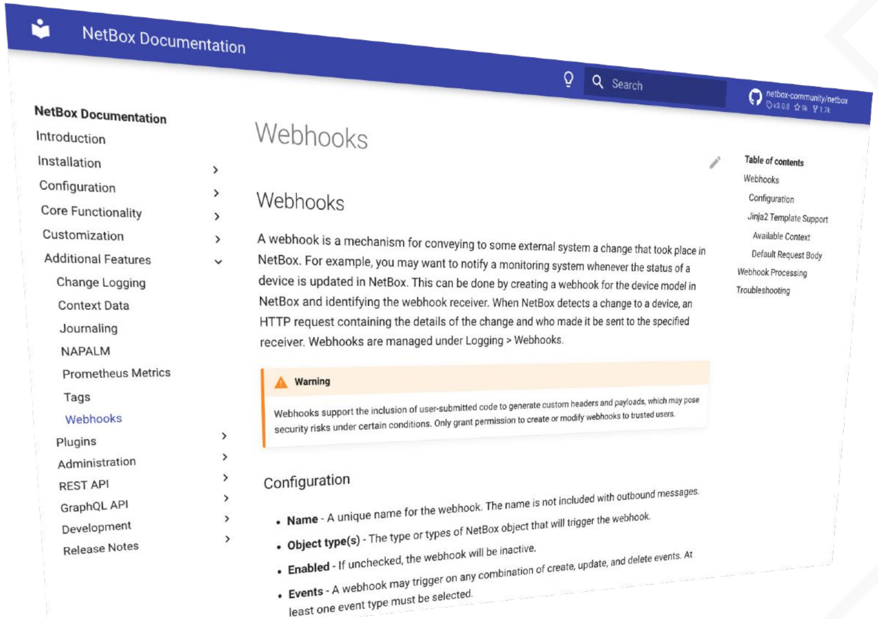
manually…..

updating…….

inetnum……

objects...

But wait, there is an api for that...



NetBox Documentation

Search

netbox-community/netbox

NetBox Documentation

- Introduction
- Installation
- Configuration
- Core Functionality
- Customization
- Additional Features
  - Change Logging
  - Context Data
  - Journaling
  - NAPALM
  - Prometheus Metrics
  - Tags
  - Webhooks
- Plugins
- Administration
- REST API
- GraphQL API
- Development
- Release Notes

Webhooks

Table of contents

- Webhooks
  - Configuration
  - Jinja2 Template Support
  - Available Context
  - Default Request Body
- Webhook Processing
- Troubleshooting

Webhooks

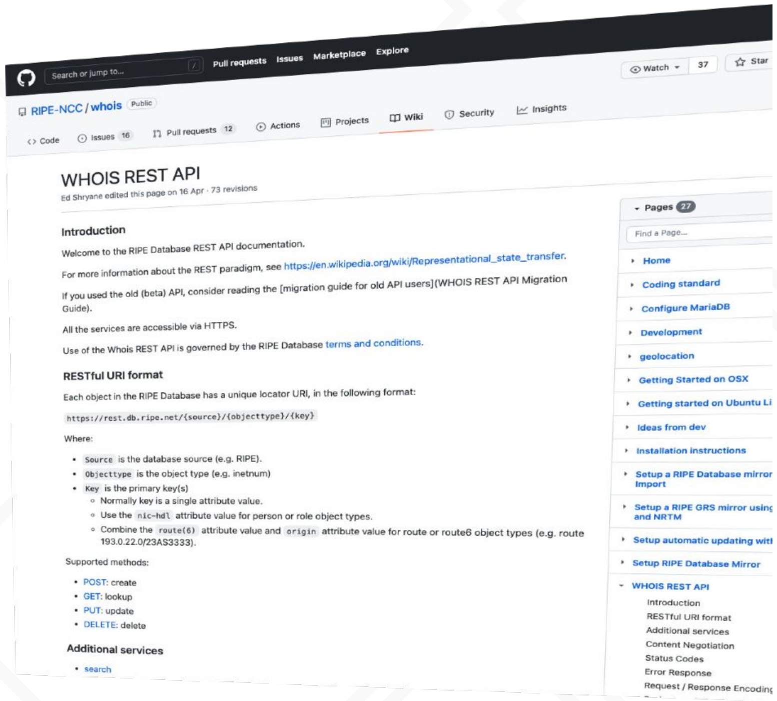
A webhook is a mechanism for conveying to some external system a change that took place in NetBox. For example, you may want to notify a monitoring system whenever the status of a device is updated in NetBox. This can be done by creating a webhook for the device model in NetBox and identifying the webhook receiver. When NetBox detects a change to a device, an HTTP request containing the details of the change and who made it be sent to the specified receiver. Webhooks are managed under Logging > Webhooks.

Warning

Webhooks support the inclusion of user-submitted code to generate custom headers and payloads, which may pose security risks under certain conditions. Only grant permission to create or modify webhooks to trusted users.

Configuration

- **Name** - A unique name for the webhook. The name is not included with outbound messages.
- **Object type(s)** - The type or types of NetBox object that will trigger the webhook.
- **Enabled** - If unchecked, the webhook will be inactive.
- **Events** - A webhook may trigger on any combination of create, update, and delete events. At least one event type must be selected.



Search or jump to... Pull requests Issues Marketplace Explore

Watch 37 Star

RIPE-NCC / whois Public

Code Issues 16 Pull requests 12 Actions Projects Wiki Security Insights

WHOIS REST API

Ed Shryane edited this page on 16 Apr · 73 revisions

Introduction

Welcome to the RIPE Database REST API documentation.

For more information about the REST paradigm, see https://en.wikipedia.org/wiki/Representational_state_transfer.

If you used the old (beta) API, consider reading the [migration guide for old API users](WHOIS REST API Migration Guide).

All the services are accessible via HTTPS.

Use of the Whois REST API is governed by the RIPE Database terms and conditions.

RESTful URI format

Each object in the RIPE Database has a unique locator URI, in the following format:

```
https://rest.db.ripe.net/{source}/{objecttype}/{key}
```

Where:

- `Source` is the database source (e.g. RIPE).
- `Objecttype` is the object type (e.g. inetnum)
- `Key` is the primary key(s)
  - Normally key is a single attribute value.
  - Use the `nic-hdl` attribute value for person or role object types.
  - Combine the `route(6)` attribute value and `origin` attribute value for route or route6 object types (e.g. route 193.0.22.0/23AS3333).

Supported methods:

- POST: create
- GET: lookup
- PUT: update
- DELETE: delete

Additional services

- search

Pages 27

Find a Page...

- Home
- Coding standard
- Configure MariaDB
- Development
- geolocation
- Getting Started on OSX
- Getting started on Ubuntu Li
- Ideas from dev
- Installation instructions
- Setup a RIPE Database mirror Import
- Setup a RIPE GRS mirror using and NRTM
- Setup automatic updating wit
- Setup RIPE Database Mirror
- WHOIS REST API
  - Introduction
  - RESTful URI format
  - Additional services
  - Content Negotiation
  - Status Codes
  - Error Response
  - Request / Response Encoding

...that is insufficient.

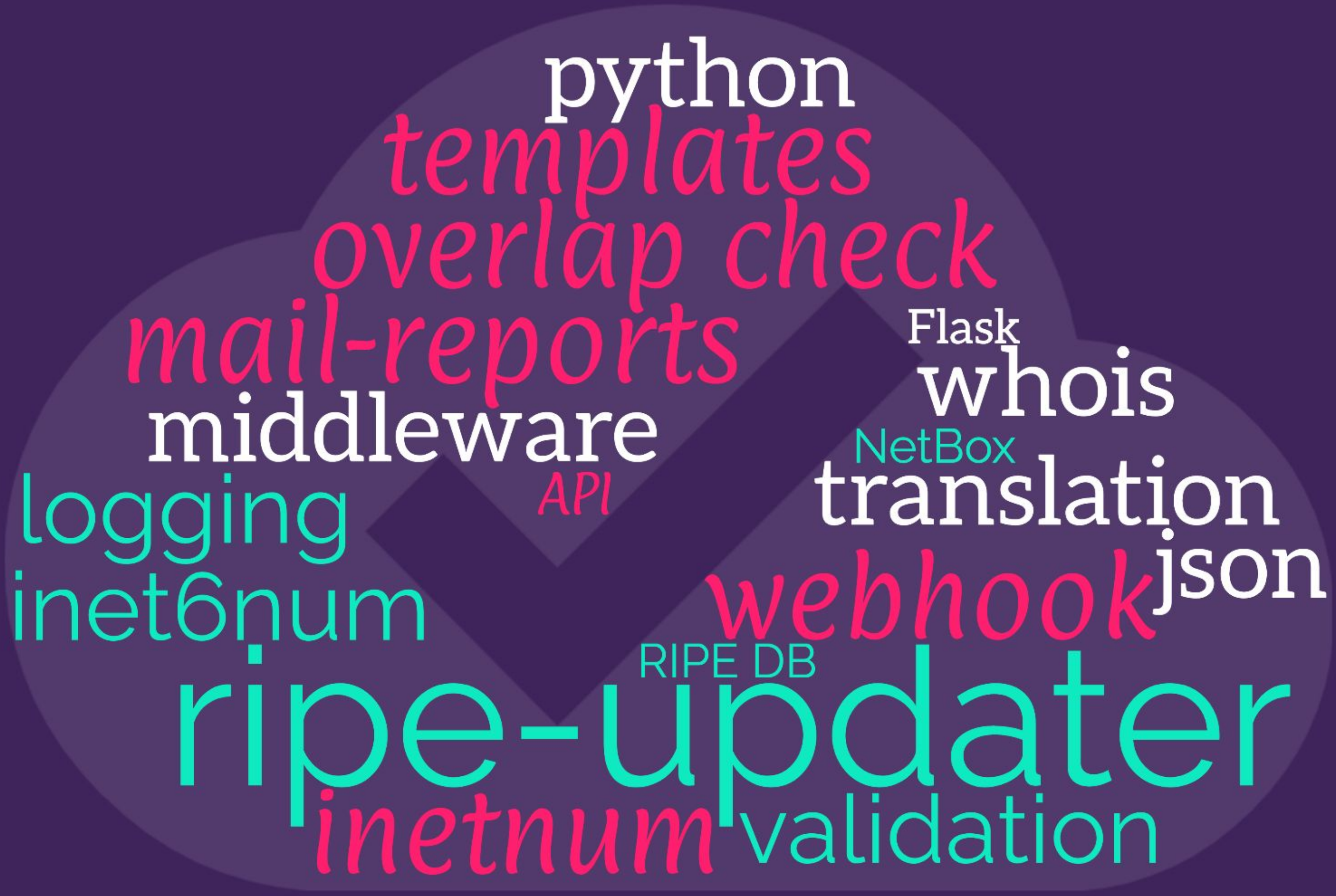
NetBox

```
prefix: 198.51.100.0/25
tenant: cust1
aggregate: 198.51.100.0/24
...
```

RIPE DB

```
inetnum: 198.51.100.0 -
198.51.100.127
netname: X?
desc: X?
admin-c: X?
tech-c: X?
...
```

Inspektor
Konsola
Debuger
Edytor stylów
Wydajność
Sieć
visual design


python
templates
overlap check
mail-reports
Flask
whois
middleware
NetBox
API
logging
translation
json
webhook
inet6num
RIPE DB
ripe-updater
inetnum
validation

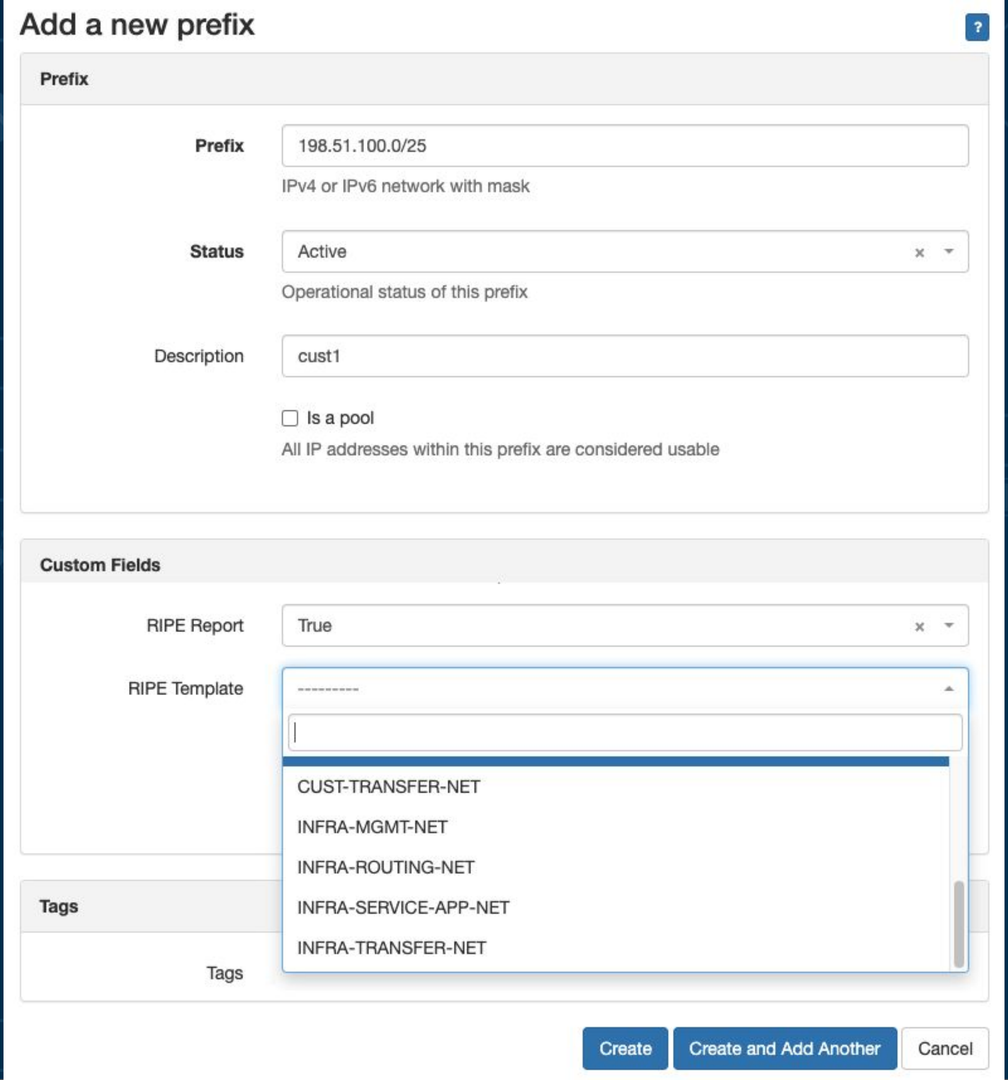
Requirements

SysEleven

NetBox
Version 2.4.0 or later
Add some custom fields

RIPE Database
Access with maintainer password
Wishlist: @RIPE api token access would be nice :)

Platform
Able to run docker-container



Add a new prefix

Prefix

Prefix 198.51.100.0/25
IPv4 or IPv6 network with mask

Status Active
Operational status of this prefix

Description cust1

☐ Is a pool
All IP addresses within this prefix are considered usable

Custom Fields

RIPE Report True

RIPE Template ---------

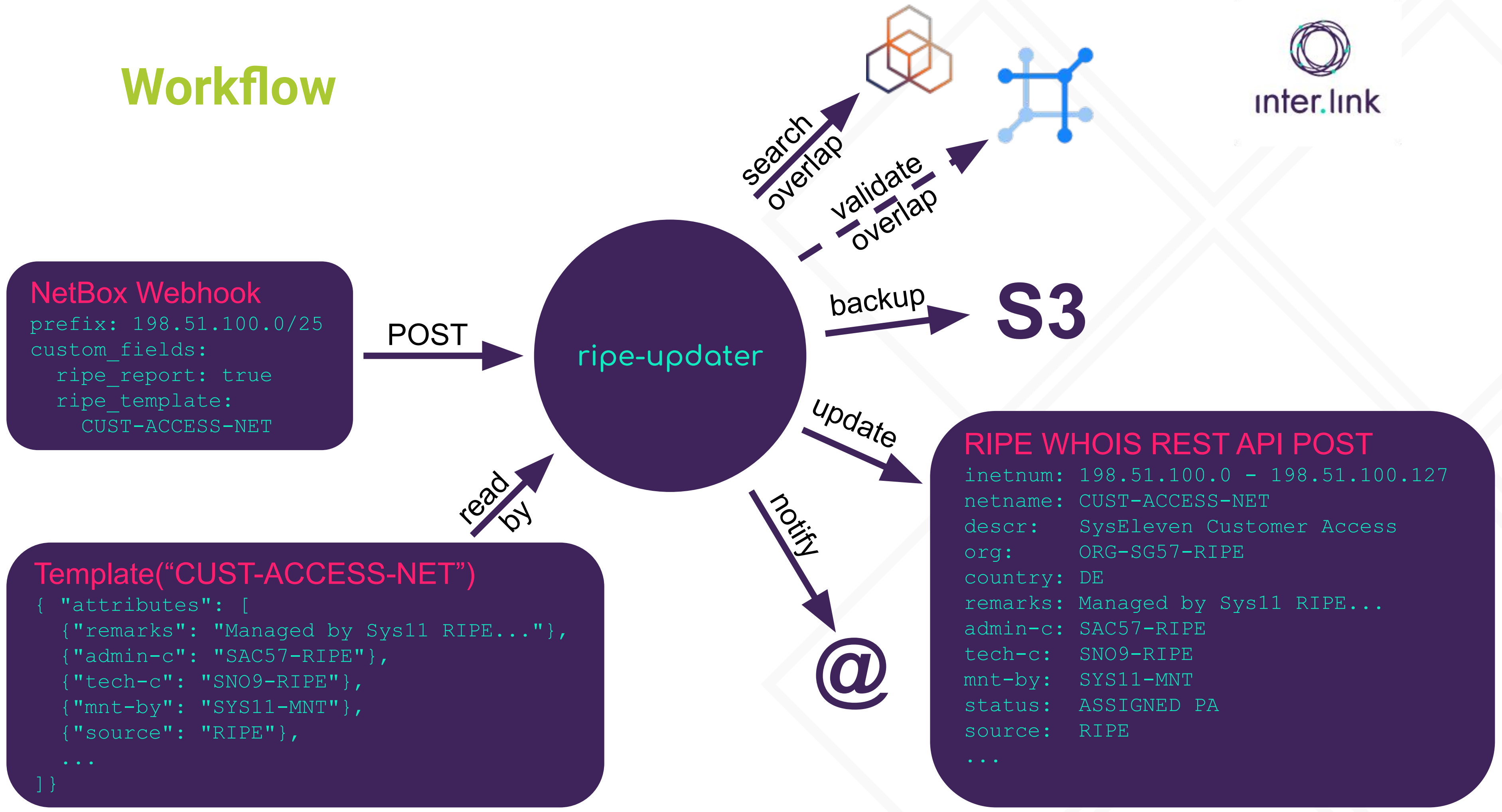
CUST-TRANSFER-NET
INFRA-MGMT-NET
INFRA-ROUTING-NET
INFRA-SERVICE-APP-NET
INFRA-TRANSFER-NET

Tags

Tags

SysEleven

Workflow



NetBox Webhook

```
prefix: 198.51.100.0/25
custom_fields:
  ripe_report: true
  ripe_template:
    CUST-ACCESS-NET
```

Template(“CUST-ACCESS-NET”)

```
{ "attributes": [
  {"remarks": "Managed by Sys11 RIPE..."},
  {"admin-c": "SAC57-RIPE"},
  {"tech-c": "SNO9-RIPE"},
  {"mnt-by": "SYS11-MNT"},
  {"source": "RIPE"},
  ...
]}
```

RIPE WHOIS REST API POST

```
inetnum: 198.51.100.0 - 198.51.100.127
netname: CUST-ACCESS-NET
descr:   SysEleven Customer Access
org:     ORG-SG57-RIPE
country: DE
remarks: Managed by Sys11 RIPE...
admin-c: SAC57-RIPE
tech-c:  SNO9-RIPE
mnt-by:  SYS11-MNT
status:  ASSIGNED PA
source:  RIPE
...
```

SysEleven

With ripe-updater

- ✓ Database is always updated
- ✓ Consistent objects
- ✓ Batch processing of all your prefixes is no problem

Release

ripe-updater on GitHub in Q1 2022

Stay tuned and watch
https://github.com/interdotlink/ripe-updater

SysEleven

Thank you for your attention!

and keep your RIPE DB objects updated

Any questions?

Main/initial code:	Mohamad Mouselli
Maintainer:	Christian Harendt
Architect:	Vincentz Petzholtz