



RIPE Réseaux IP
Européens

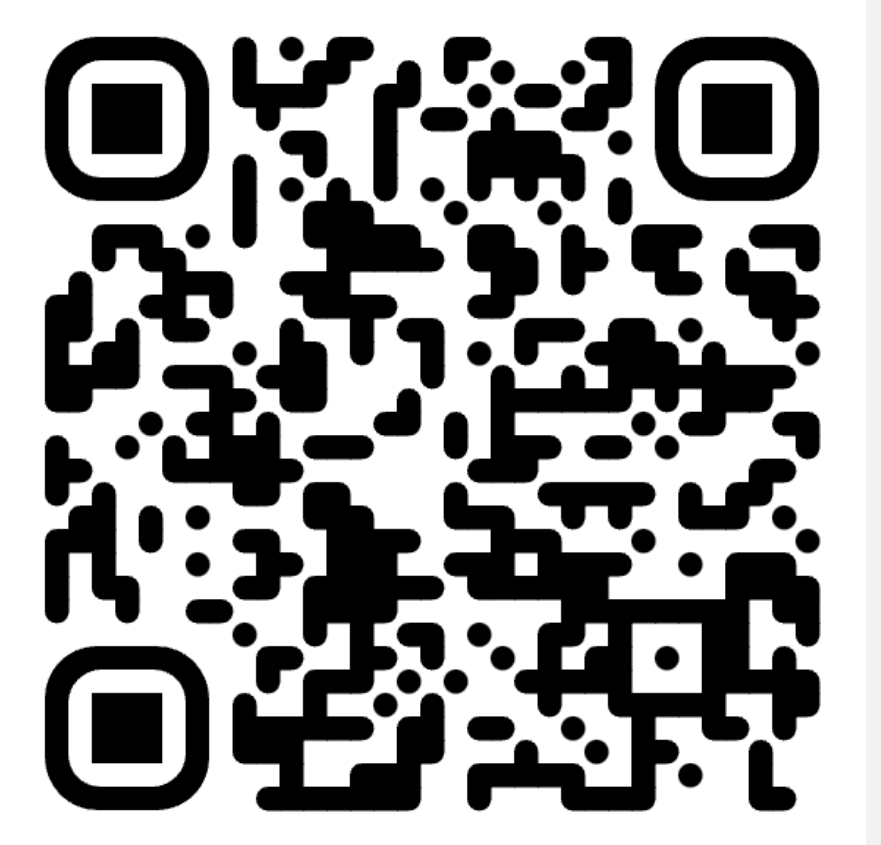
Annet takes control of the network

Grigorii Solovev, Yandex Infrastructure | RIPE 90 | 13.05.2025

Annet - network configuration automation tool



- Opensource project - <https://github.com/annetutil>
- Widely adopted in several large companies
- Supports vendors:
 - Cisco – Cisco IOS/-XE/-XR, NXOS
 - Juniper – JunOS
 - Huawei – VRP5/VRP8
 - Arista – EOS
 - Nokia – TiMOS
 - MikroTik – RouterOS
 - SONiC –
/etc/sonic/config_db.json
 - Whitebox – Linux (FRR, snmpd, ntpd and etc)



Why we decided to develop Annet?



- Multivendor networks
- Legacy devices
- A lot of devices in production (10k+)
- Complicated configurations
- Transition between configurations

Software defined configuration



- Full configuration templating
- Create reference configuration and deploy it
- Advantages:
 - Configuration checks
 - Mass changes
 - Actual documentation
 - Git, review and tests

Configuration methods



Requirements	CLI	NETCONF	Ansible
Machine-readable	✗	✓	✓
Multivendor	✗	+	+
Reliable delivery	✗	✓	+
Human-readable	✓	✗	+
Comfortable for writing	✓	✗	+
Versatility	✓	✗	+

Annet uses CLI method



- Every device has CLI
- CLI is convenient for engineers
- CLI is machine-readable in reality

Need to be solved:

- Multivendor support
- Reliable delivery

Annet: fundamental



- Declarative
- Modularity
- Templates on Python
- Can support every device (every device has CLI)
- There is no one single model of device configuration
- Integrated with inventory system
- Server less, only CLI client

Hands on examples



```
$ annet gen -g ntp sas-127d2
ntp server disable
ntp ipv6 server disable
ntp server source-interface all disable
ntp ipv6 server source-interface all disable
ntp unicast-server 95.108.129.181 vpn-instance MEth0/0/0
ntp unicast-server 95.108.129.183 vpn-instance MEth0/0/0
ntp unicast-server 37.140.149.198 vpn-instance MEth0/0/0
clock timezone UTC add 00:00:00
```

Hands on examples

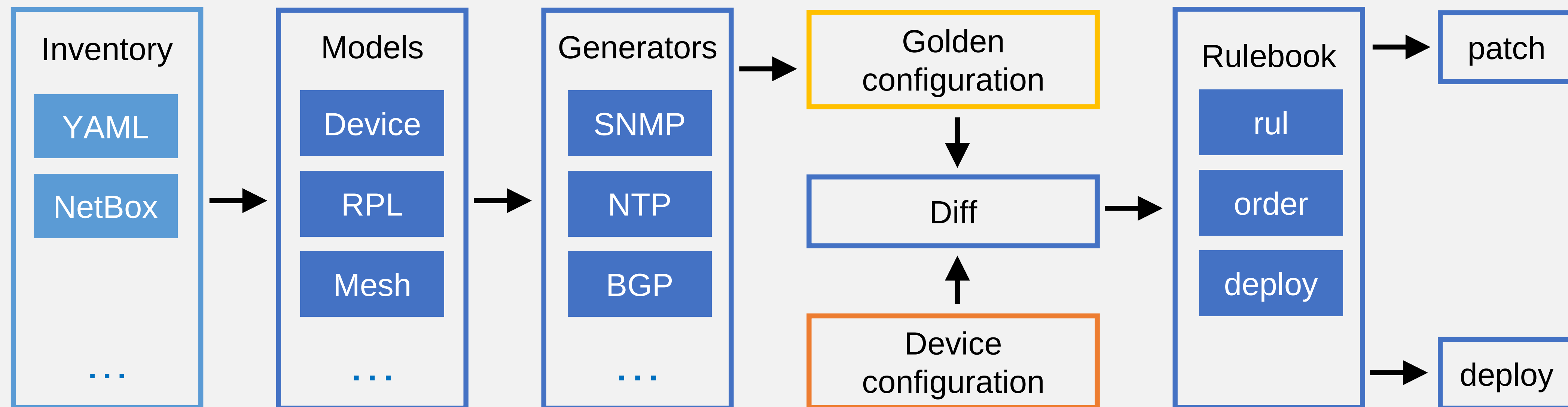


```
$ annet diff -g ntp sas-127d2
- clock timezone FRA add 02:00:00
+ clock timezone UTC add 00:00:00
+ ntp unicast-server 95.108.129.181 vpn-instance MEth0/0/0
+ ntp unicast-server 95.108.129.183 vpn-instance MEth0/0/0
```

```
$ annet patch -g ntp sas-127d2
ntp unicast-server 37.140.149.198 vpn-instance MEth0/0/0
ntp unicast-server 37.140.149.199 vpn-instance MEth0/0/0
clock timezone UTC add 00:00:00
```

```
$ annet deploy -g ntp sas-127d2
$
```

Architecture



Models: Device



- Based on inventory data
- Create python class object

```
>>> device.fqdn
'sas-127d2.yndx.net'
>>> bool(device.hw.Arista)
False
>>> [iface.name for iface in device.interfaces]
['10GE1/0/1', '10GE1/0/2', ..., 'Eth-Trunk217', 'Eth-Trunk218', ..., 'LoopBack0', 'MEth0/0/0']
```

Models: RPL



- Route policies are described in Python pseudo code
- Create python class objects

```
@routemap
def IMPORT_CONNECTED(device, route):
    with route(
        number=10,
        R.protocol == "connected",
        R.match_v4("LOCAL_NETS", or_longer=(24, 32))
    ) as rule:
        rule.community.set("ADVERTISE")
        rule.allow()
    with route(number=20) as rule:
        rule.deny()
```



- Route policies elements are defined by Python objects

```
PREFIX_LISTS = [  
    IpPrefixList(name="LOCAL_NETS", members=["192.168.0.0/16"])  
]  
COMMUNITIES = [  
    CommunityList(name="ADVERTISE", members=["65000:1"])  
]
```

Models: Mesh



- Describe device neighbors based on templates
- Templates applies to connection in the inventory
- Create Python class objects for remote peers

```
@registry.direct("tor-{pod}-{num}", "spine-{pod}-{plane}")
def tor_to_spine(tor, spine, session):
    tor.asnum = BASE_ASNUM + 100 + tor.match.pod * 10 + tor.match.num
    tor.addr = f"10.{spine.match.plane}.{tor.match.num}.12/24"
    tor.import_policy = "TOR_IMPORT_SPINE"
    tor.export_policy = "TOR_EXPORT_SPINE"
    spine.asnum = BASE_ASNUM + 200 + spine.match.pod
    spine.addr = f"10.{spine.match.plane}.{tor.match.num}.11/24"
    spine.import_policy = "SPINE_IMPORT_TOR"
    spine.export_policy = "SPINE_EXPORT_TOR"
```

Models: Mesh



tor-1-1.nh.com

+ Add Components

Created 2024-11-07 11:19 · Updated 1 month, 3 weeks ago

Device

Interfaces 5

Config Context

Render Config

Contacts

Join

Quick search

☐	Name	Enabled	Type	Connection
☐	eth0	✓	100BASE-TX (10/100ME)	—
☐	eth1	✓	1000BASE-T (1GE)	spine-1-1.nh.com > eth1
☐	eth2	✓	1000BASE-T (1GE)	spine-1-2.nh.com > eth1

```
>>> [(peer.addr, peer.remote_as) for peer in mesh_data.peers]
[('10.1.1.11', ASN(65201)), ('10.2.1.11', ASN(65201))]
```



- Python generator is class with per vendor methods
- Generators has access to inventory data and models
- It yields configuration lines

```
def run_juniper(self, device):  
    if device.tags["Frankfurt"]:  
        servers = NTP_SERVERS_FRA  
    else:  
        servers = NTP_SERVERS_GLOBAL  
  
    for server in servers:  
        with self.multiblock("system", "ntp"):  
            yield "server", server, "version 3"
```

```
def run_huawei(self, device):  
    if device.tags["Frankfurt"]:  
        servers = NTP_SERVERS_FRA  
    else:  
        servers = NTP_SERVERS_GLOBAL  
  
    for server in servers:  
        yield "ntp unicast-server", server
```



- Limit scope of the generator
- Limit configurations lines for diff
- Uses for modularity

```
snmp-agent
ifindex constant
interface
    mac-address trap notification
    mac-address notification learning
set if-mib discard-statistics enable
acl number 410
acl ipv6 number 610
```



- Redefine no/undo/delete commands to expected format

```
interface 100GE0/1/39
-  port-queue ef wfq weight 3 outbound
+  port-queue ef wfq weight 5 outbound
```

```
interface 100GE0/1/39
  undo port-queue ef wfq outbound
  port-queue ef wfq weight 5 outbound
quit
```

- Very flexible logic
- Order of configurations lines in patch
- Response on questions



- Opensource library - gnetcli
- Written on Golang
- Annet uses gnetcli for connection to devices



- Pagination
- Interactive interaction
- NETCONF
- Various network operating systems
- CLI tool
- GRPC server
- Python SDK

Real world example - drain spine in DC fabric



1. Assign tag to the device

spine-1-1
Created 2025-02-26 11:25 · Updated 0 minutes ago

Device Config Context Render Config Cont...

Device	
Region	—
Site	lab
Location	—
Rack	—
Position	—
GPS Coordinates	—
Tenant	—
Device Type	Arista cEOS (1U)
Description	—
Airflow	—
Serial Number	—
Asset Tag	—
Config Template	—

Tags

maintenance

2. Update route policy

```
$ annet diff -g rpl spine-1-1
route-map SPINE_EXPORT_TOR permit 10
+ set community 65535:0 additive
```

```
$ annet deploy -g rpl spine-1-1
$
```

3. The task is done

Real world example - drain link between devices



1. Assign tag to the cable

Cables dcim.cable:1

#1 Bookmark Clone Edit Delete

Created 2025-01-26 14:00 · Updated 0 minutes ago

Cable Journal Changelog

Cable	
Type	—
Status	Connected
Tenant	—
Label	—
Description	—
Color	—
Length	—

Tags	
maintenance	

Termination A	
Site	lab
Location	—
Rack	—
Interface	r1.lab > Ethernet1

Termination B	
Site	lab
Location	—
Rack	—
Interface	r2.lab > Ethernet1

2. Update bgp configuration

```
$ annet diff -g bgp r1
router bgp 65001
+ neighbor 10.1.2.12 shutdown
```

```
$ annet deploy -g bgp r1
$
```

3. The task is done

Annet – feel free to add new ones



- Supports vendors:
 - Cisco – Cisco IOS/-XE/-XR, NXOS
 - Juniper – JunOS
 - Huawei – VRP5/VRP8
 - Arista – EOS
 - Nokia – TiMOS
 - MikroTik – RouterOS
 - SONiC – /etc/sonic/config_db.json
 - Whitebox – Linux (FRR, snmpd, ntpd and etc)

Annet – how to try



- Project repository - <https://github.com/annetutil>



- Labs



- Tutorial



- Telegram chat





Questions, feedback & dialogue

