NetOpsKube Tutorial

NetOps Stack for Core/Aggr of Any Size

Mauricio (Mau) Rojas

What would you learn?

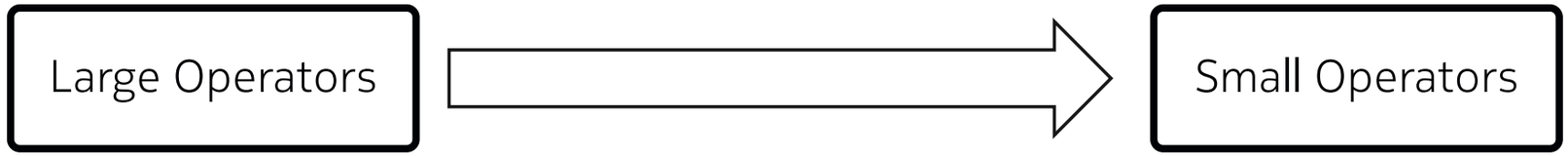
NetOpsKube

- Deploy **gNMIc**, **Prometheus**, and **Grafana** using **Kubernetes operators** for simplified management
- Implement **GitOps** workflows using **FluxCD** to manage all configurations from a Git repository, enabling **quick rollbacks**
- Discover how **SDCIO** helps maintain your desired state, identifying and correcting any **configuration drift** from your live network elements
- Discover how running these tools on **Kubernetes** provides **scalability**, **self-healing**, and high availability

What Operators would need from a NetOps stack?

Observability and Configuration Management Platform

NetOpsKube



Governance and Efficiency

Service Availability

Heavy/Distributed

Golden Config Compliance

AIOps (Noise Reduction)

Lightweight/SaaS

Config Drift Detection

Total Visibility

Our High-Performance Telemetry Stack (We Think)

As long as nobody deploys new hardware, updates a config, or looks at it too closely.

NetOpsKube

Where is Prometheus saving data, and is the disk full?

Why did the whole stack die because one VM rebooted?

One typo in a gNMIc YAML brings down the pipeline.

Spending more time fixing tools than fixing the network.

Config Management: “Please Don't Touch Anything”

NetOpsKube

We aren't managing a network; we are babysitting a delicate ecosystem that fears change.

Manually reverse-engineer what the commands used to be

Do all routers match their intended deployment configs?

"Someone" modified an ACL directly on the box!!

NetOps Stack

Adopt GitOps with ease

NetOpsKube

1

Deploy your NetOps stack in minutes

2

Version control network changes, enabling easy rollback.

3

Achieve resilient, scalable network operations, easily

OpenSource Projects for NetOps

A multitude of choices, hard to bring into focus.

NetOpsKube

OPENSOURCE PROJECTS LED BY NOKIA

gNMIc

> 200K downloads

Kubenet

CONTAINERlab

> 400K downloads

GitLab

GitOps

sdcio

python™

NetOpsKube

NetOpsKube

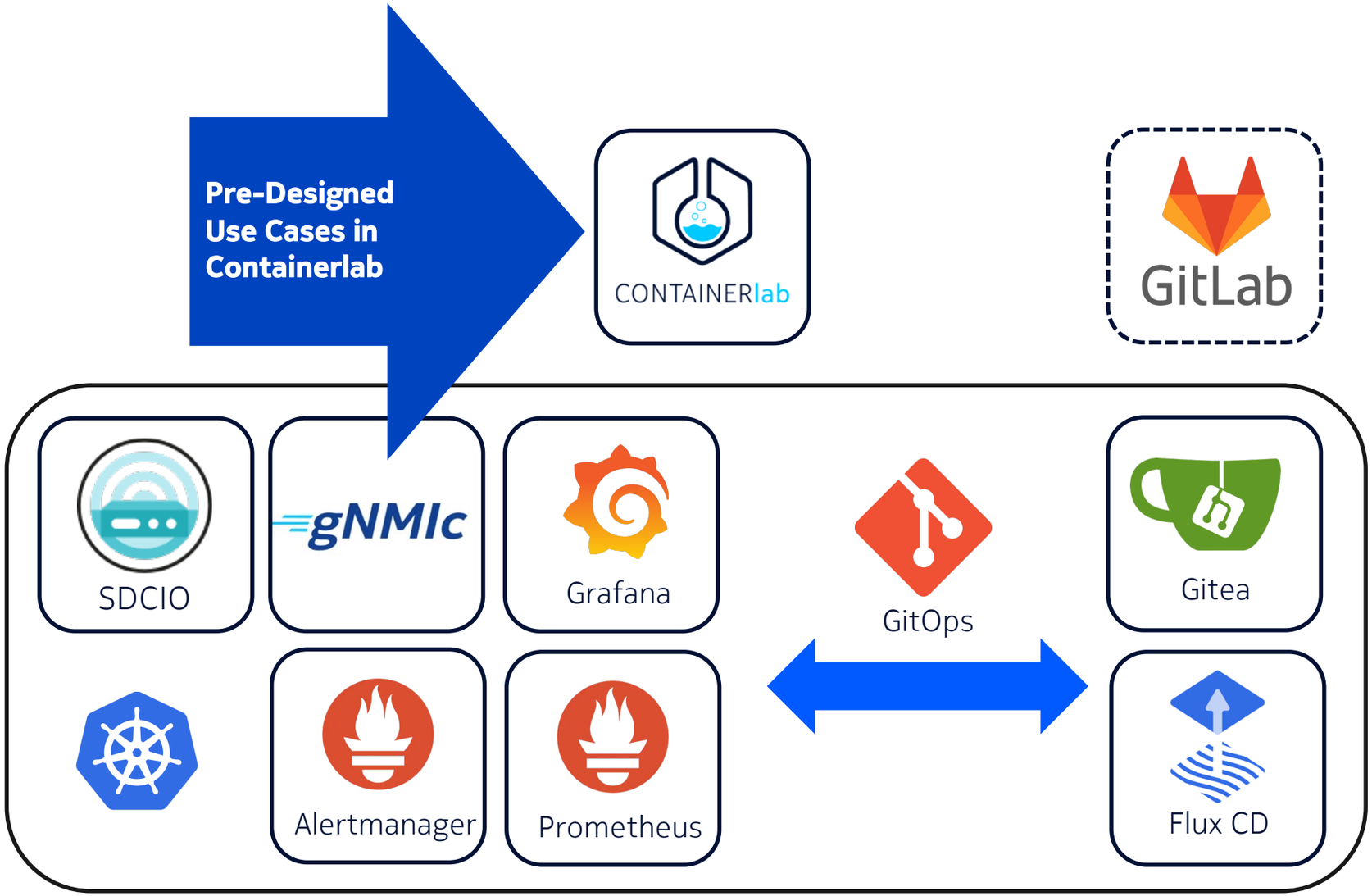
Config Management
Observability
Alarms/Notifications

NetOpsKube

production ready

lightweight

scalable


Pre-Designed
Use Cases in
Containerlab
CONTAINERlab
GitLab
NetOpsKube
SDCIO
gNMIc
Grafana
GitOps
Gitea
Alertmanager
Prometheus
Flux CD

NetOps Stack for Multiple use Cases

Multiple Use Cases – Multiple Domains – Multiple **Clab** Topologies

NetOpsKube

Broadband Network Gateway (BNG)

Aggregates residential and small business subscribers, handling authentication (AAA), session management, and policy enforcement.

Direct Internet Access (DIA)

Provides dedicated, high-bandwidth, symmetrical internet connectivity directly to enterprise customers with guaranteed Service Level Agreements (SLAs).

Metro Aggregation

Converges traffic from various access nodes (mobile backhaul, business, residential) and carries it across the metropolitan area to the network core.

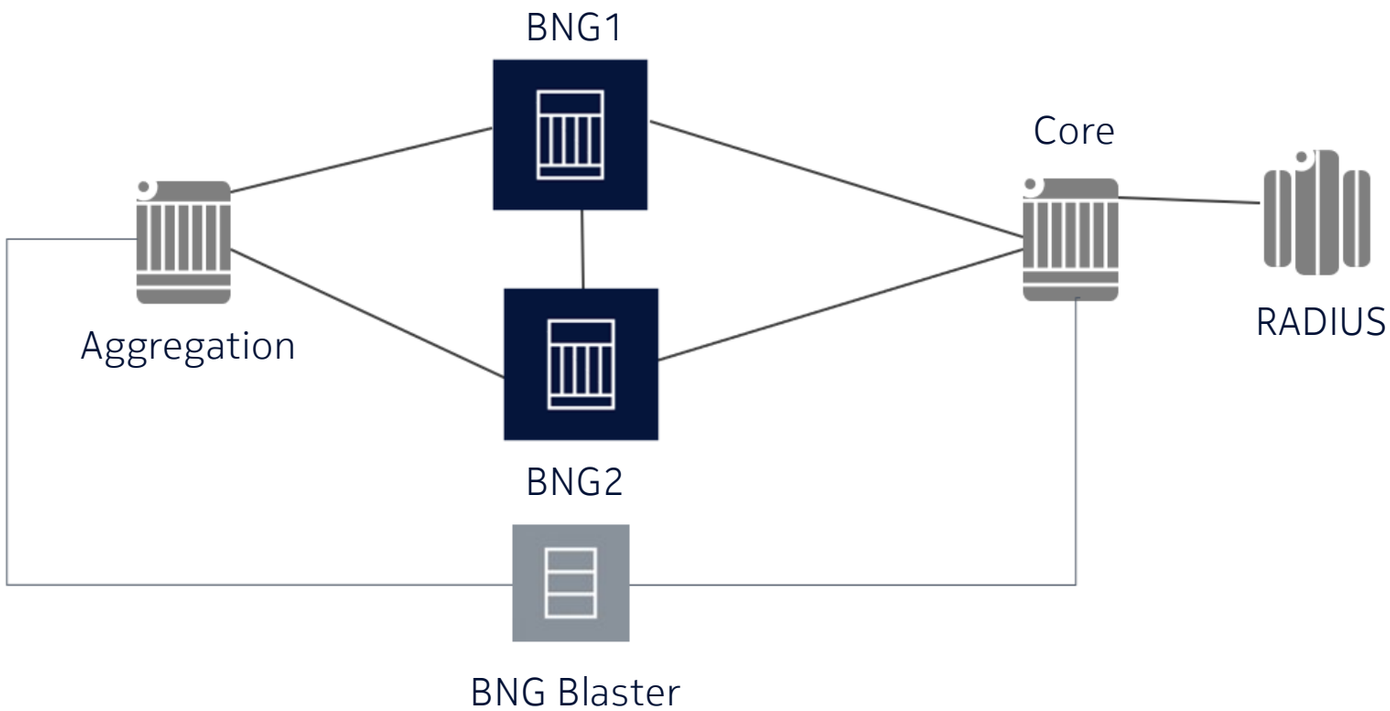
CONTAINERlab

Broadband Network Gateway (BNG) part 1

Multiple use cases (clab topologies)

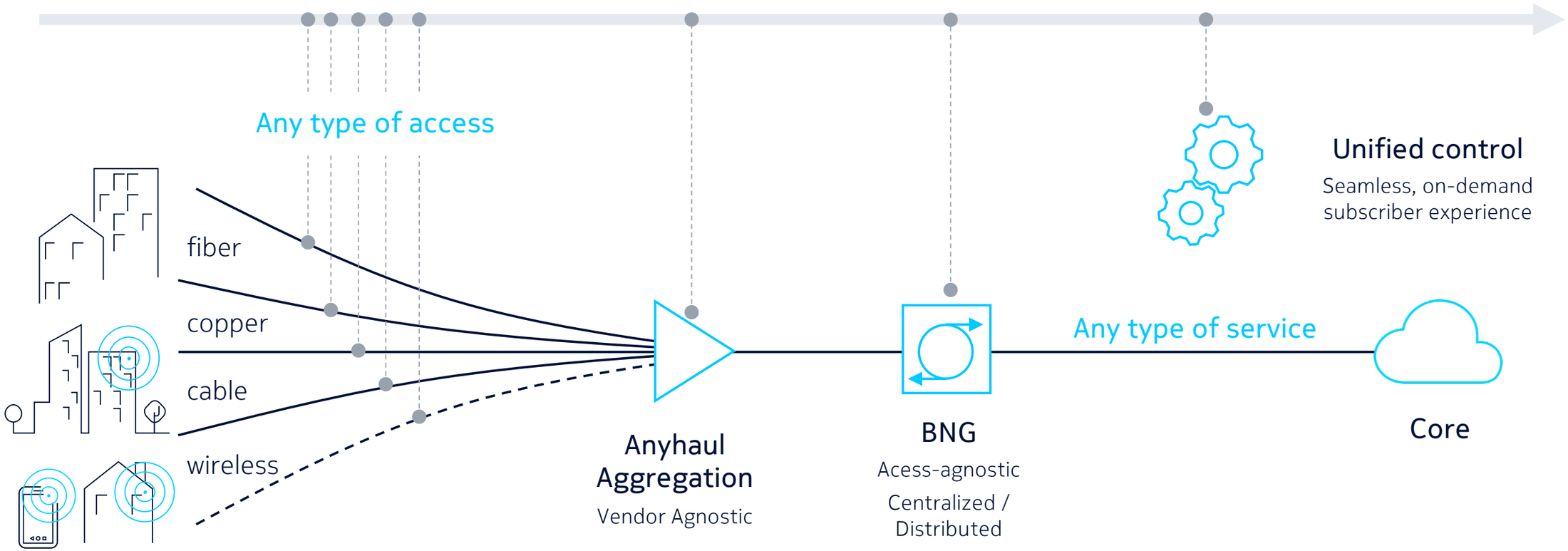
NetOpsKube

CONTAINERlab



Broadband Network Gateway (BNG) part 2

NetOpsKube



BNG - TR-101 Aggregation Architecture

NetOpsKube

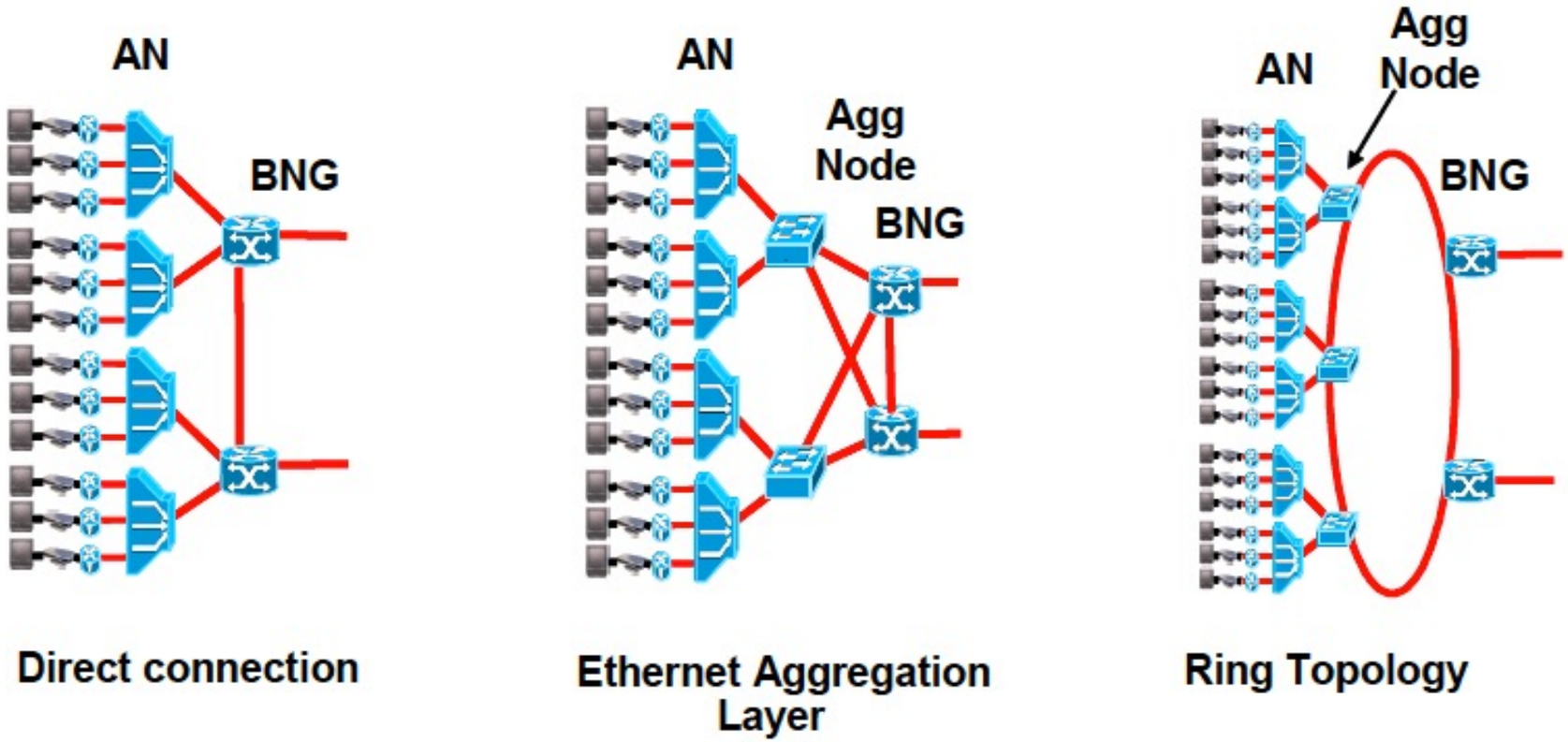
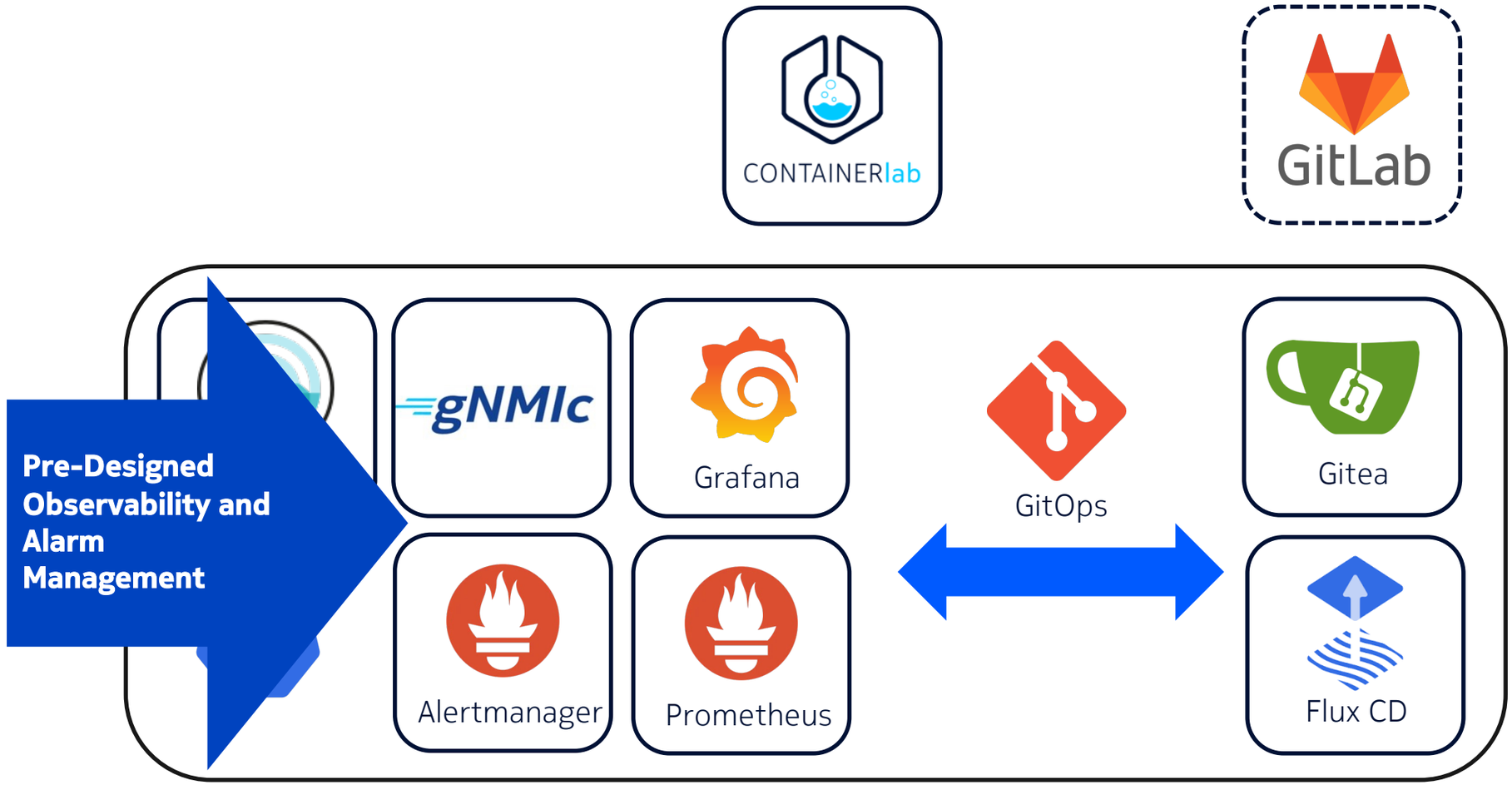


Figure 11 – Example aggregation architecture options

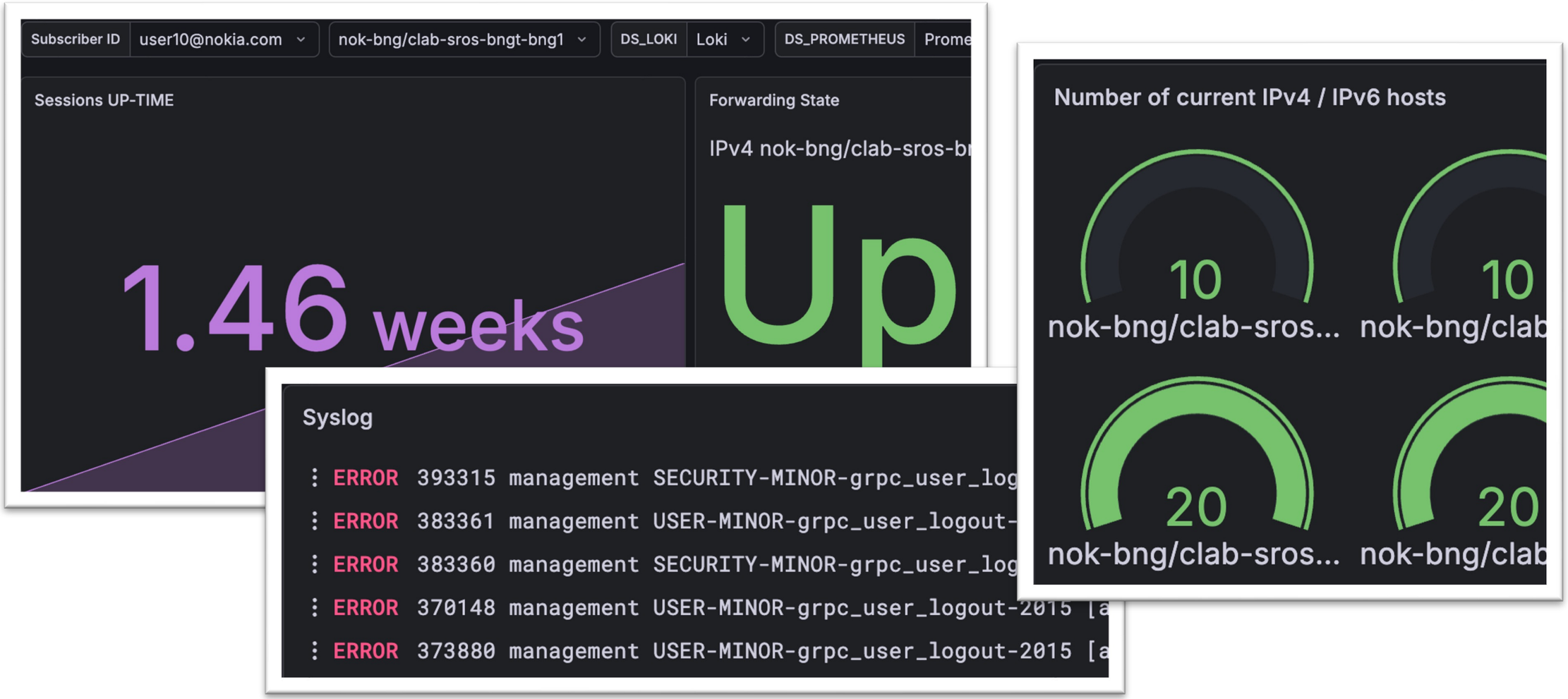
NetOpsKube

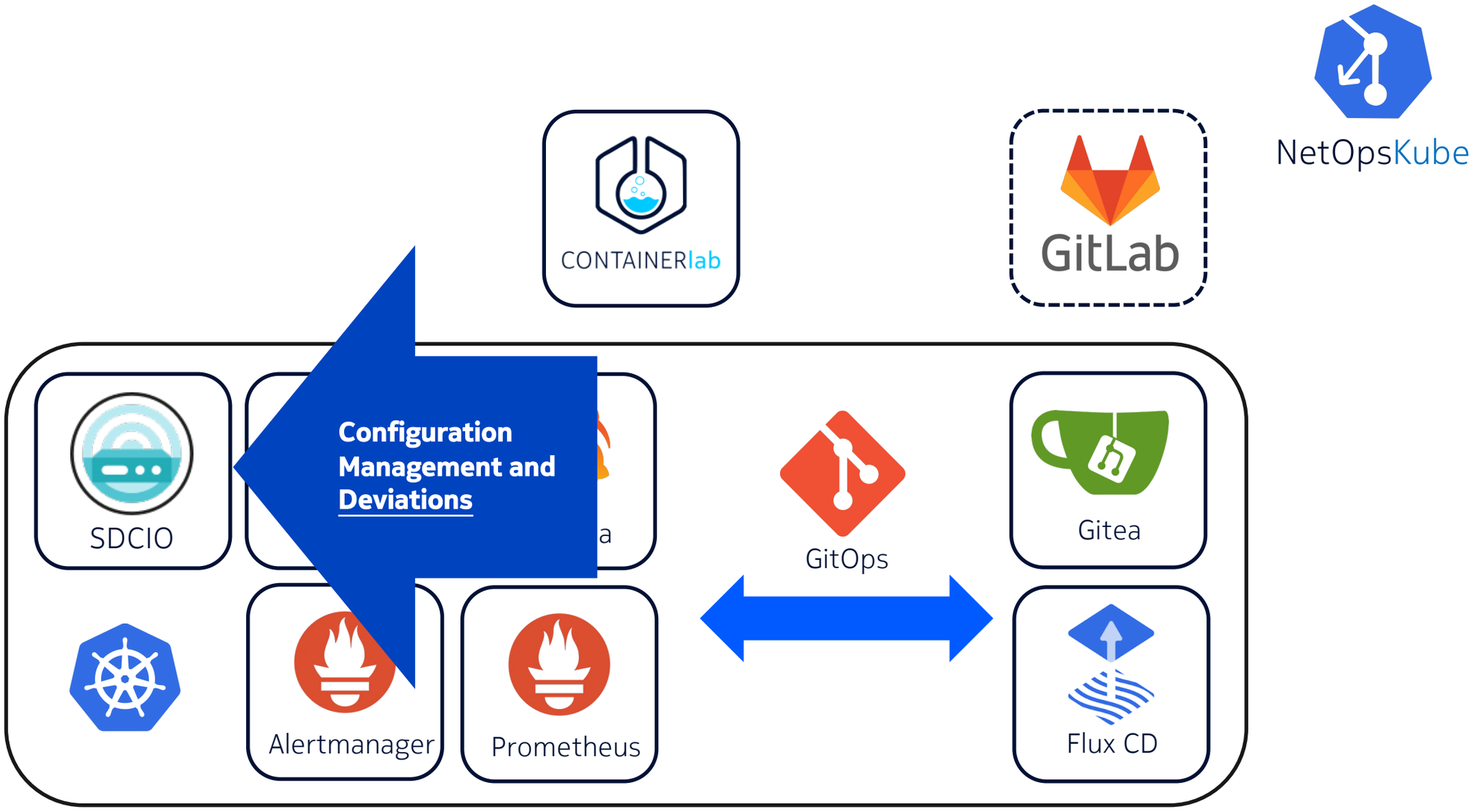

CONTAINERlab
GitLab
Pre-Designed Observability and Alarm Management
gNMIc
Grafana
GitOps
Gitea
Alertmanager
Prometheus
Flux CD

Broadband Network Gateway (BNG)

Pre-Designed Panels

NetOpsKube

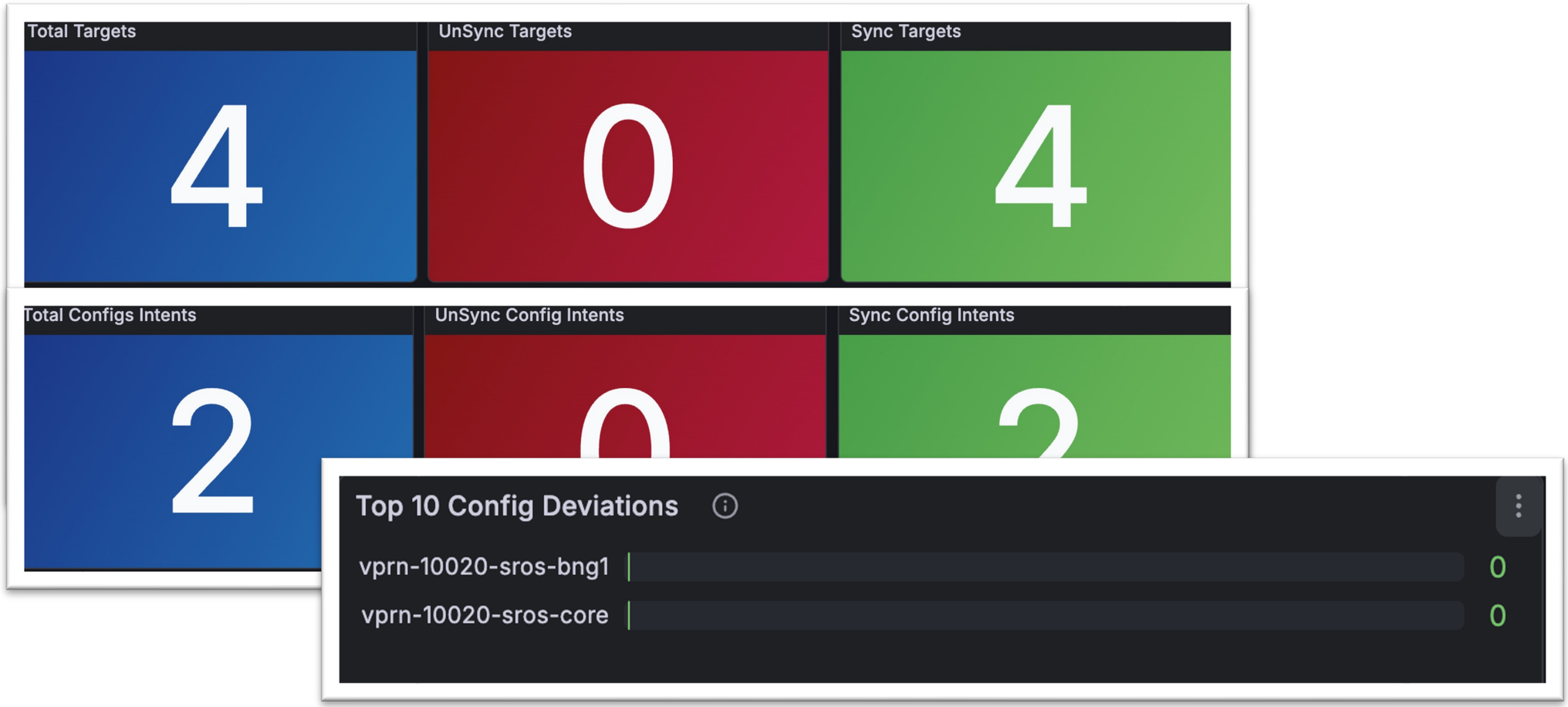


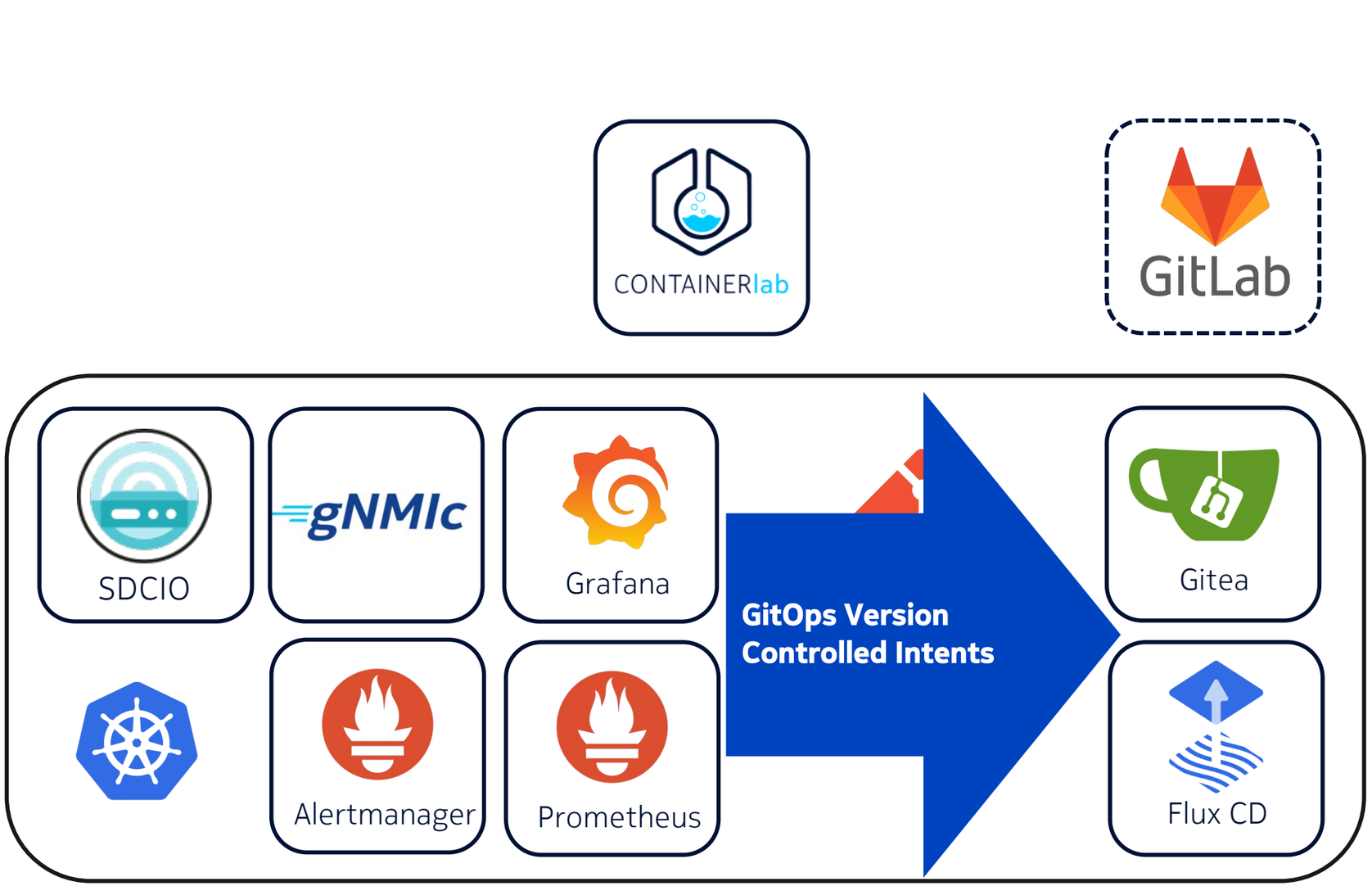

NetOpsKube
CONTAINERlab
GitLab
SDCIO
Configuration Management and Deviations
GitOps
Gitea
Alertmanager
Prometheus
Flux CD

Broadband Network Gateway (BNG)

Detect Deviations and Test Configurations in Advance

NetOpsKube

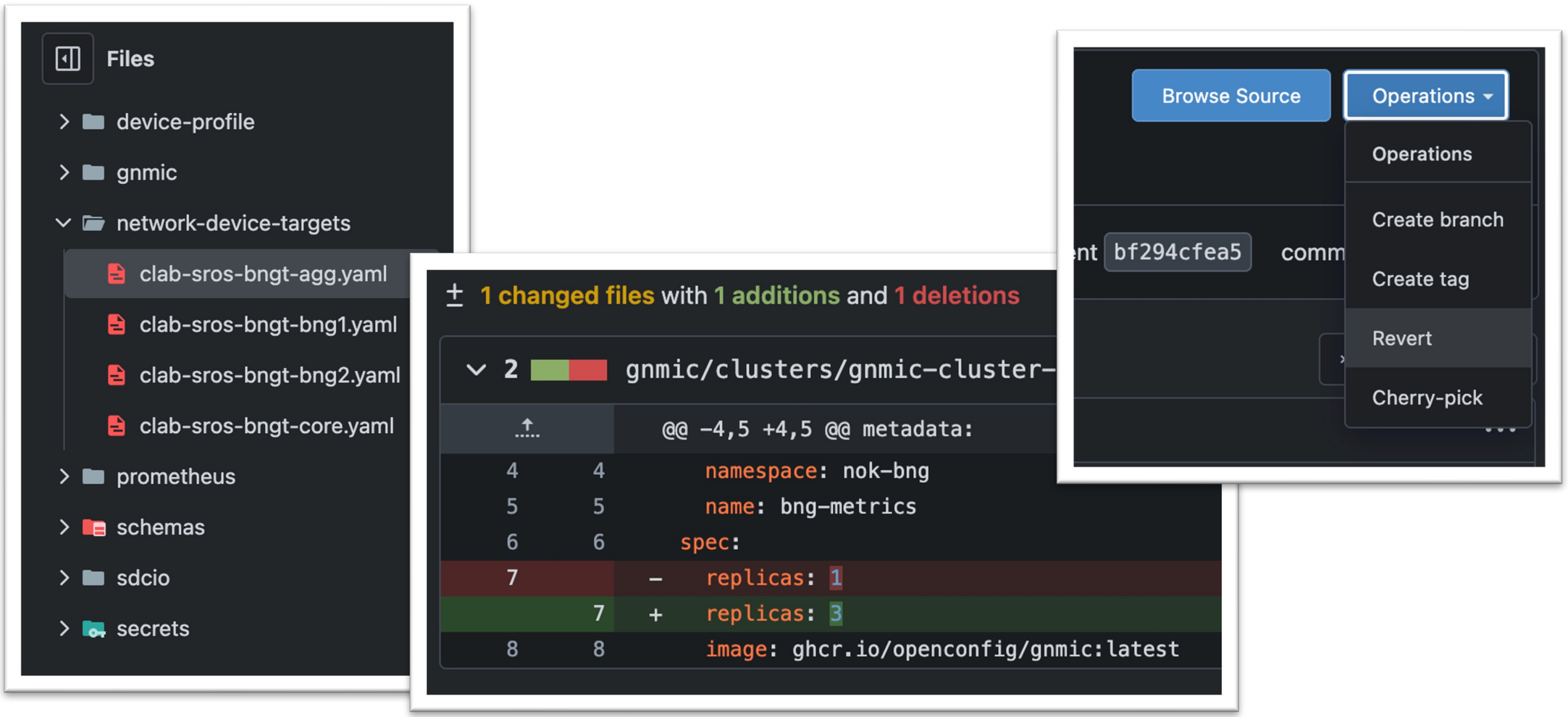



NetOpsKube
CONTAINERlab
GitLab
SDCIO
gNMIc
Grafana
GitOps Version
Controlled Intents
Gitea
Alertmanager
Prometheus
Flux CD

Broadband Network Gateway (BNG)

GitOps: Version Control and Easy Rollback

NetOpsKube



router container images might require vendor approval in advanced

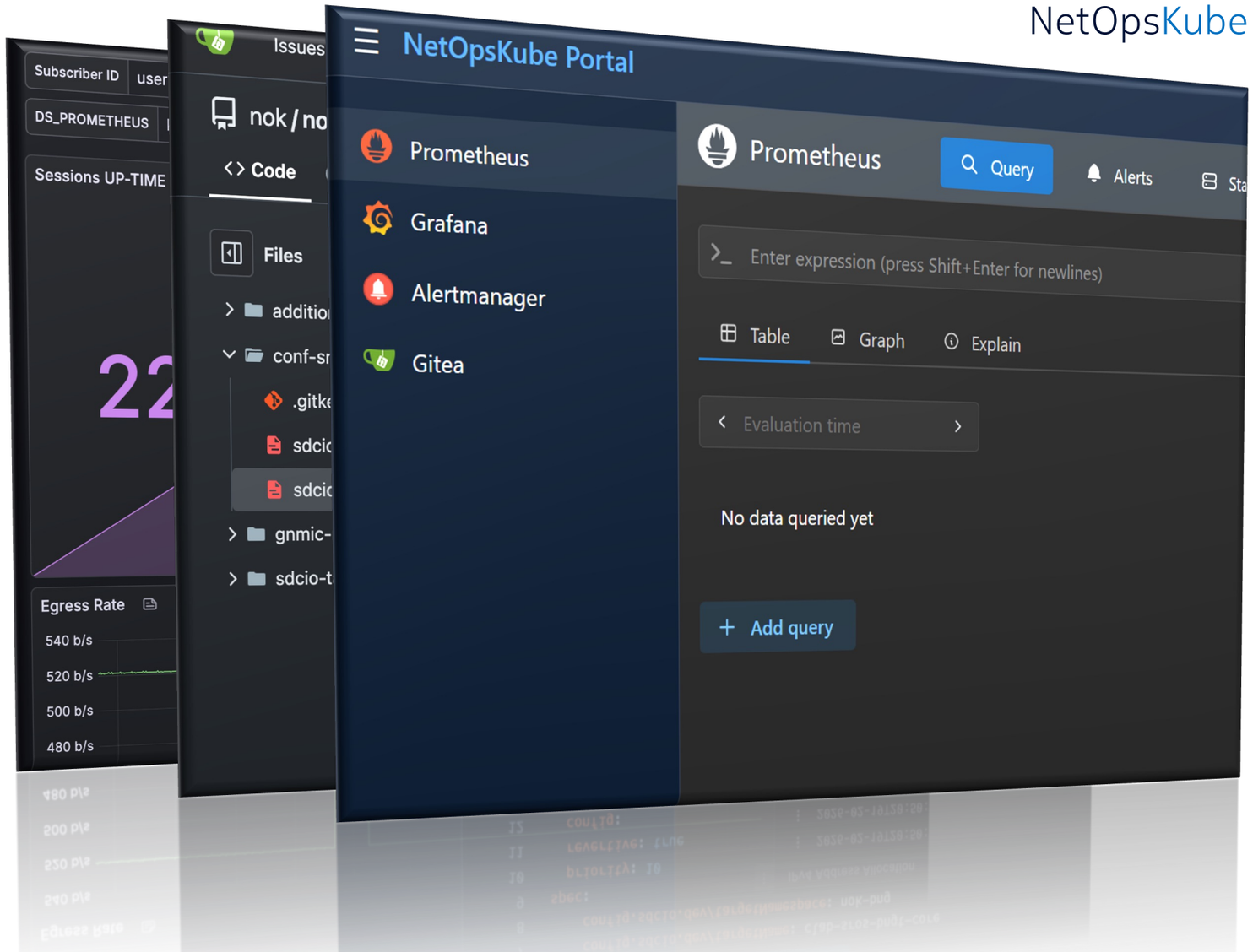
NetOpsKube

1

```
make try-nok-bng
```

2

```
sudo make deploy-clab-bng
```



RtBrick BNG Blaster

SDC

More info

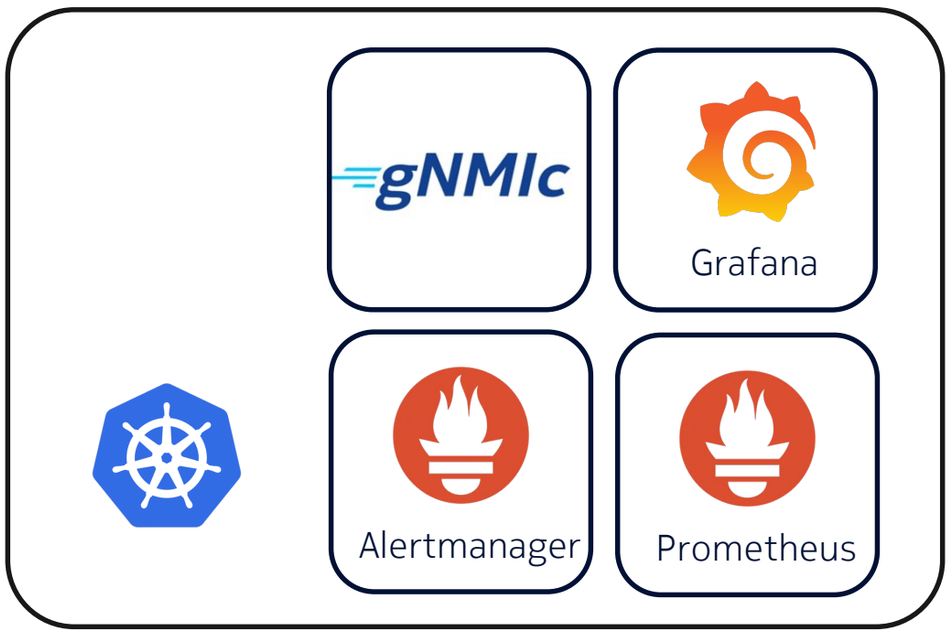
NetOpsKube

github.com/**CSPDevLabs**/**NetOpsKube**

Advanced Observability

Observability Components

NetOpsKube



- **gNMIc:** Collector for gNMI network telemetry.
- **Prometheus:** An open-source time-series database for metrics monitoring and alerting.
- **Grafana:** A multi-platform visualization tool for querying and creating dashboards.
- **Alertmanager:** Handles alert deduplication, grouping, and routing to notification channels.

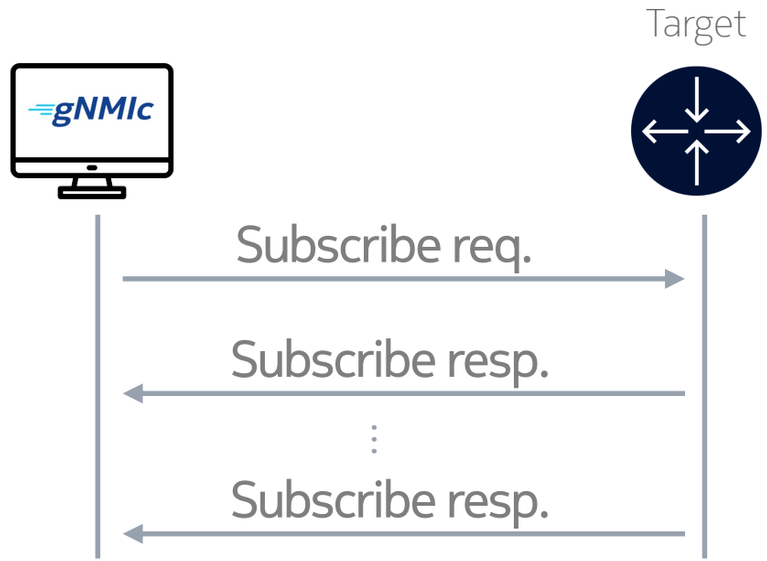
Nokia donated gNMIc to Openconfig

NetOpsKube

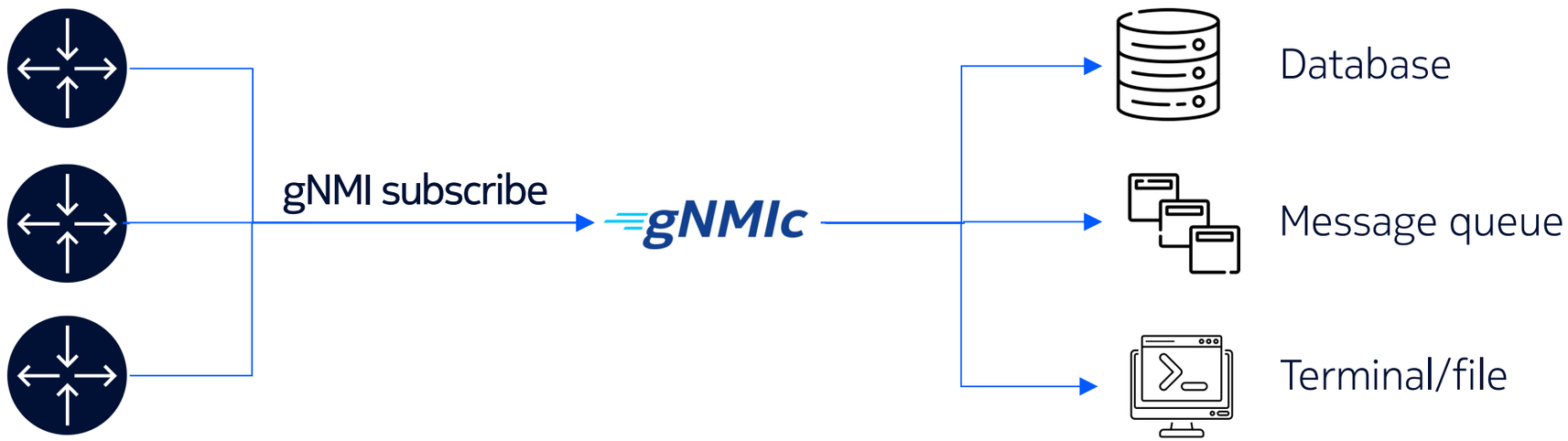
gNMIc

Subscribe Command

- Used by clients that wish to receive updates related to specific objects in the target config or state stores.
- The client creates a subscription that consists of a set of paths and a subscription mode.
- The client collects the streaming telemetry data for further processing and/or storage.



NetOpsKube

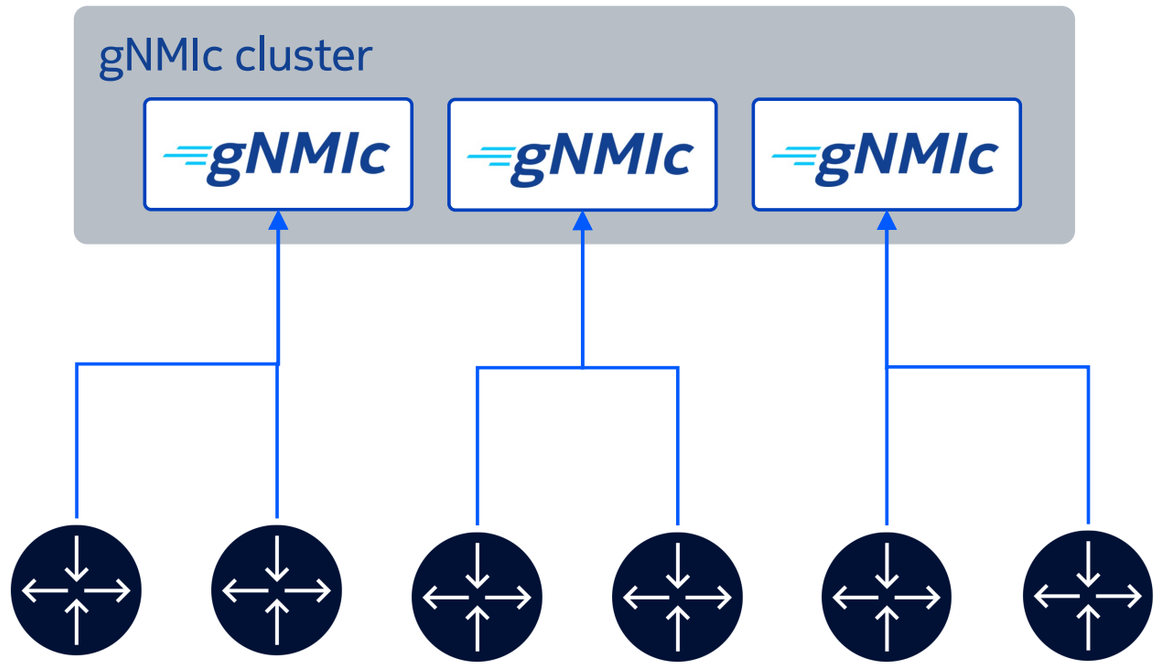


gNMIc

Kubernetes: The more the merrier

NetOpsKube

- High availability
- Scaling
- Automatic target redistribution
- Self healing

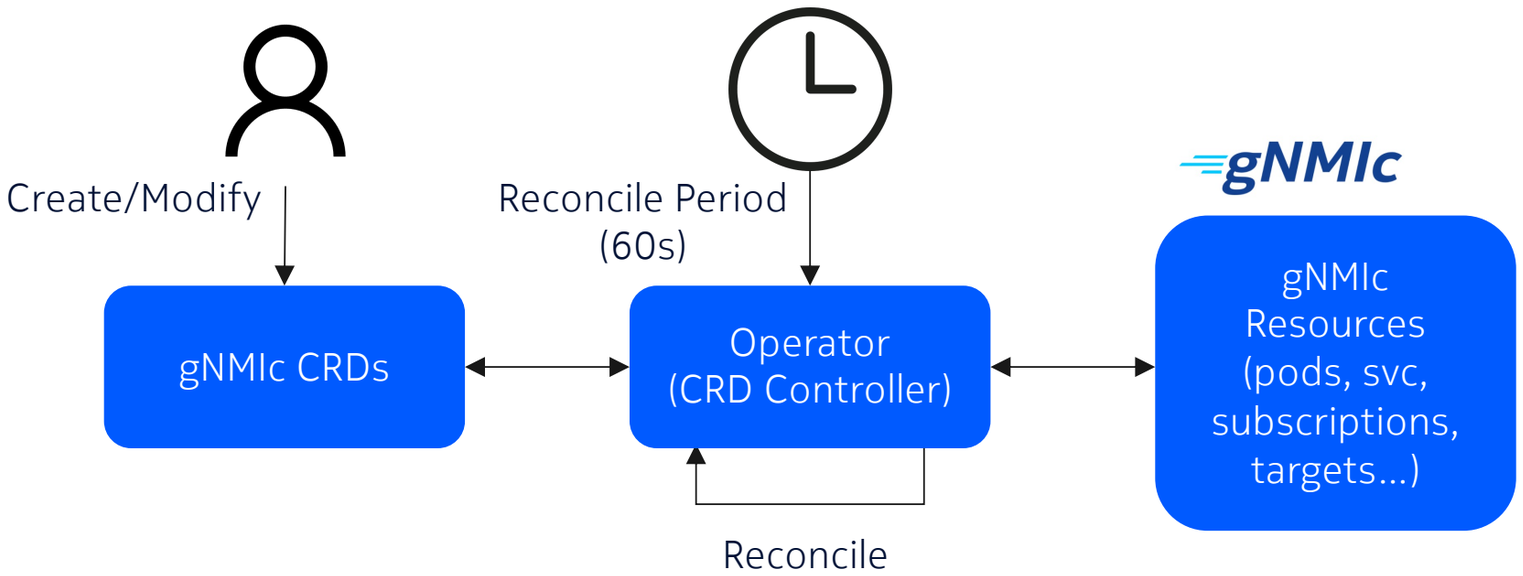


gNMIc

Kubernetes Operator

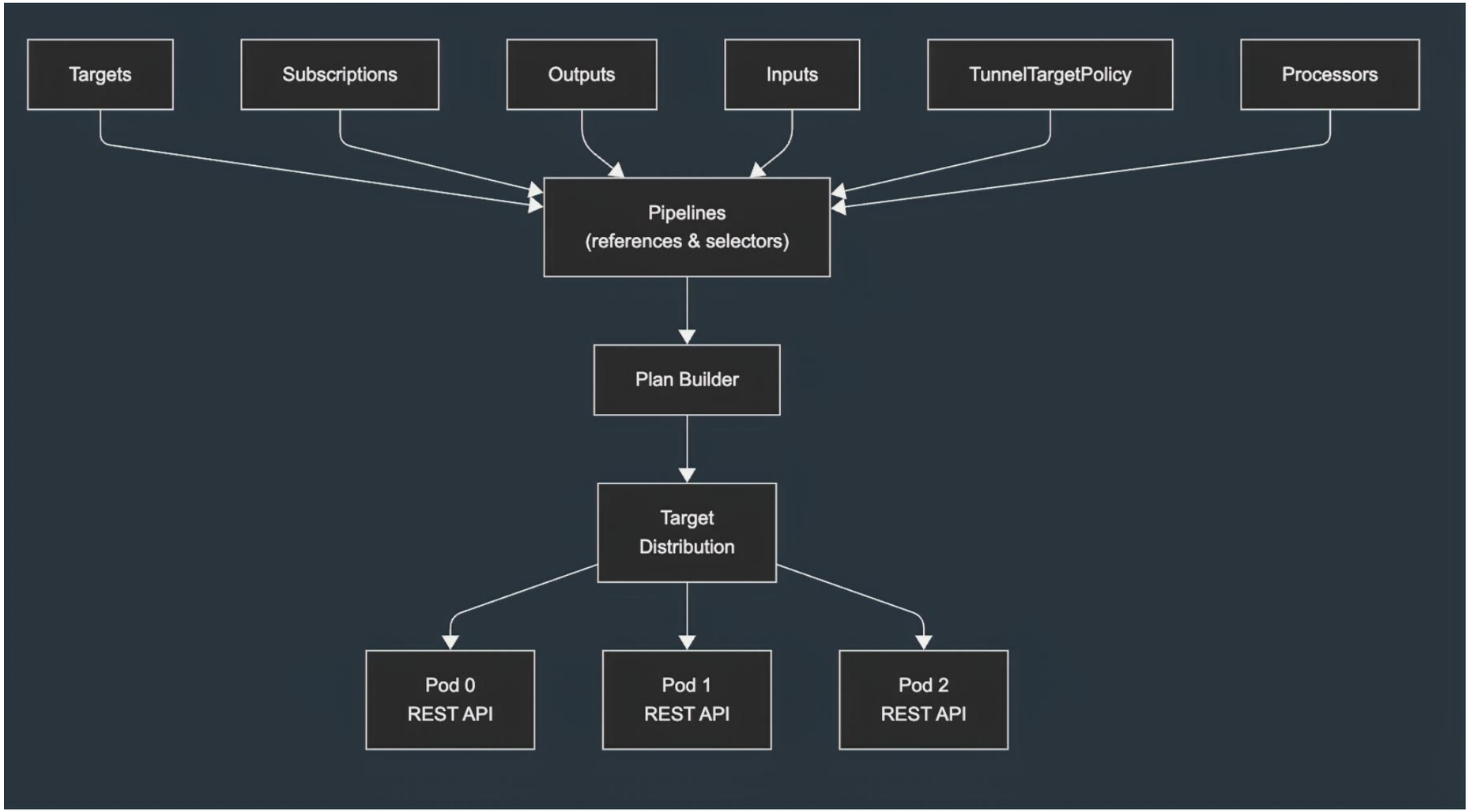
NetOpsKube

- Automates the deployment, configuration, and lifecycle of gNMIc telemetry collectors on Kubernetes
- Describe your telemetry intent using Kubernetes Custom Resources and let the operator compute and apply the effective gNMIc configuration to the right collector pods.



gNMIc

Kubernetes Operator: Configuration Flow

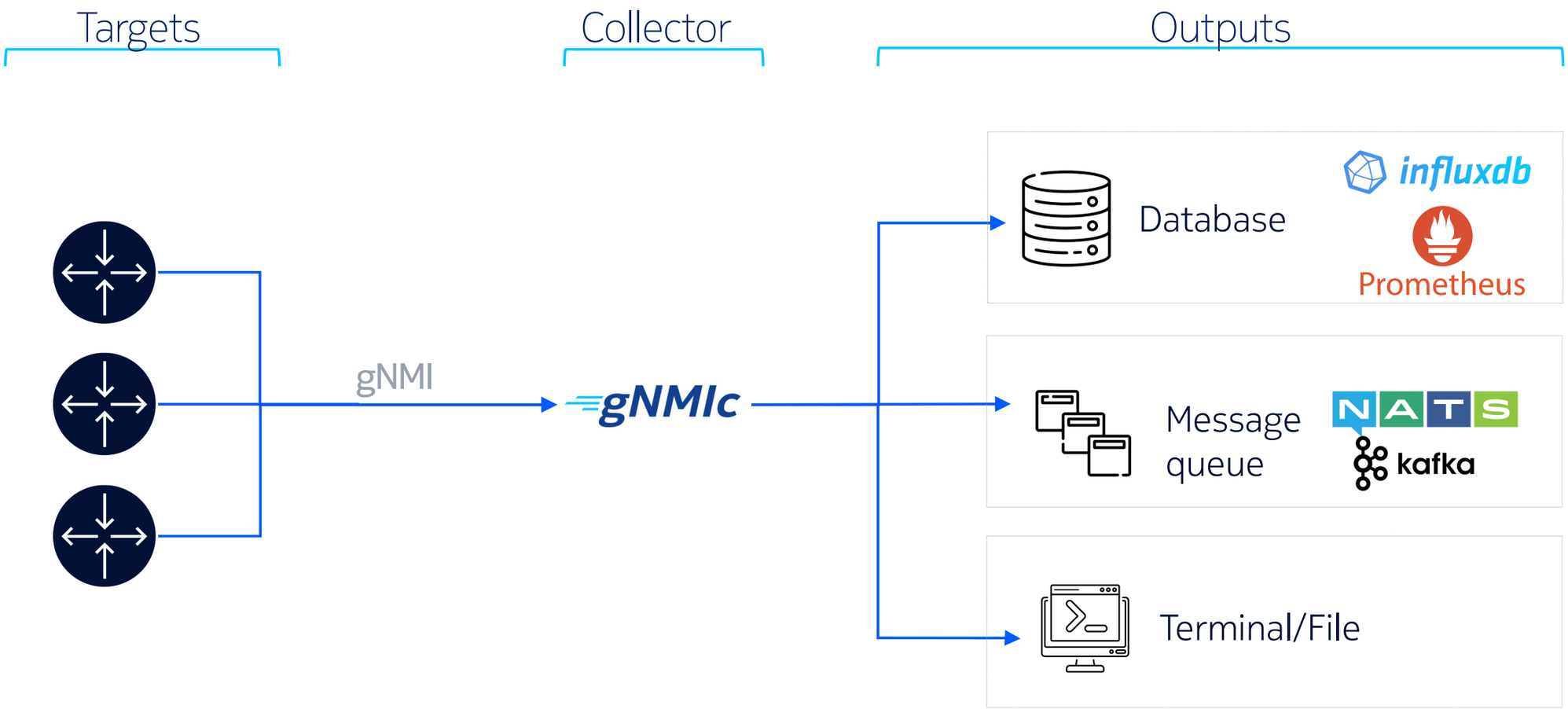


NetOpsKube

gNMIc

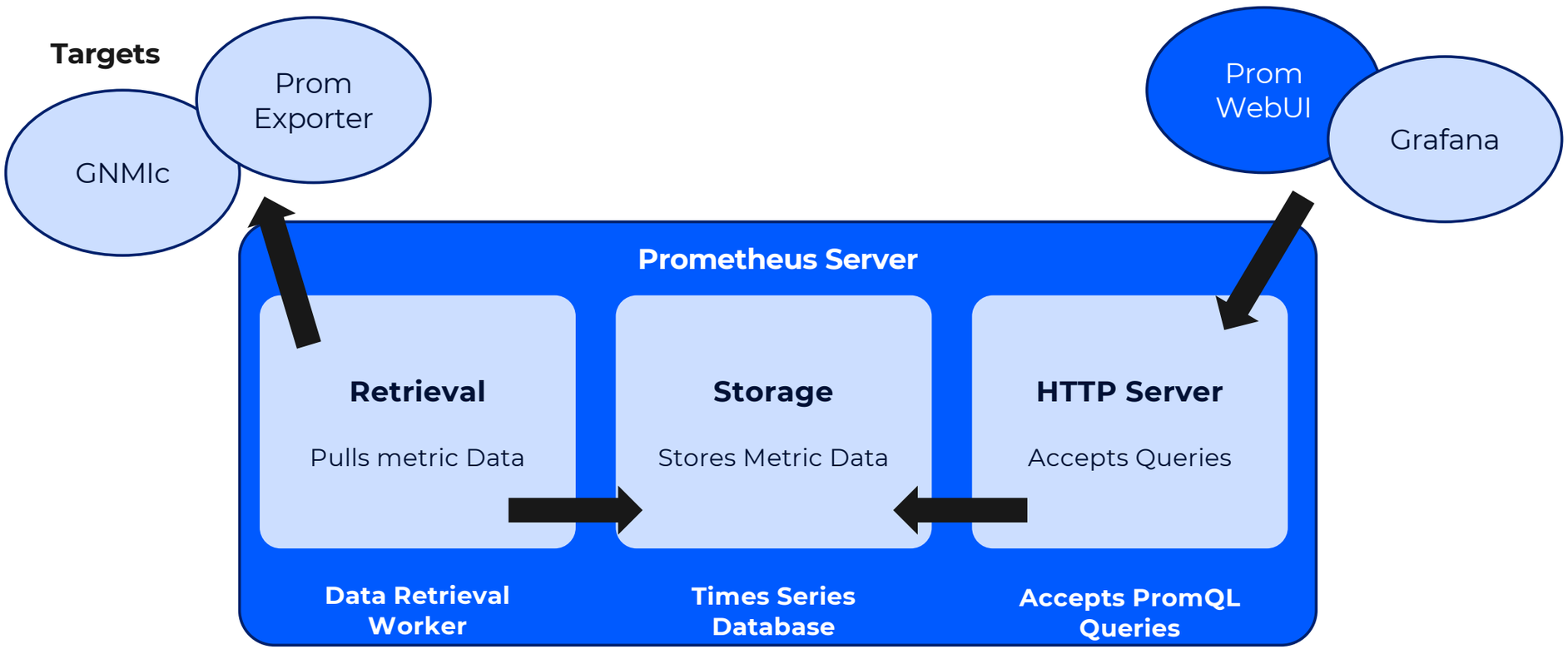
Collector outputs

NetOpsKube



Prometheus Main Components

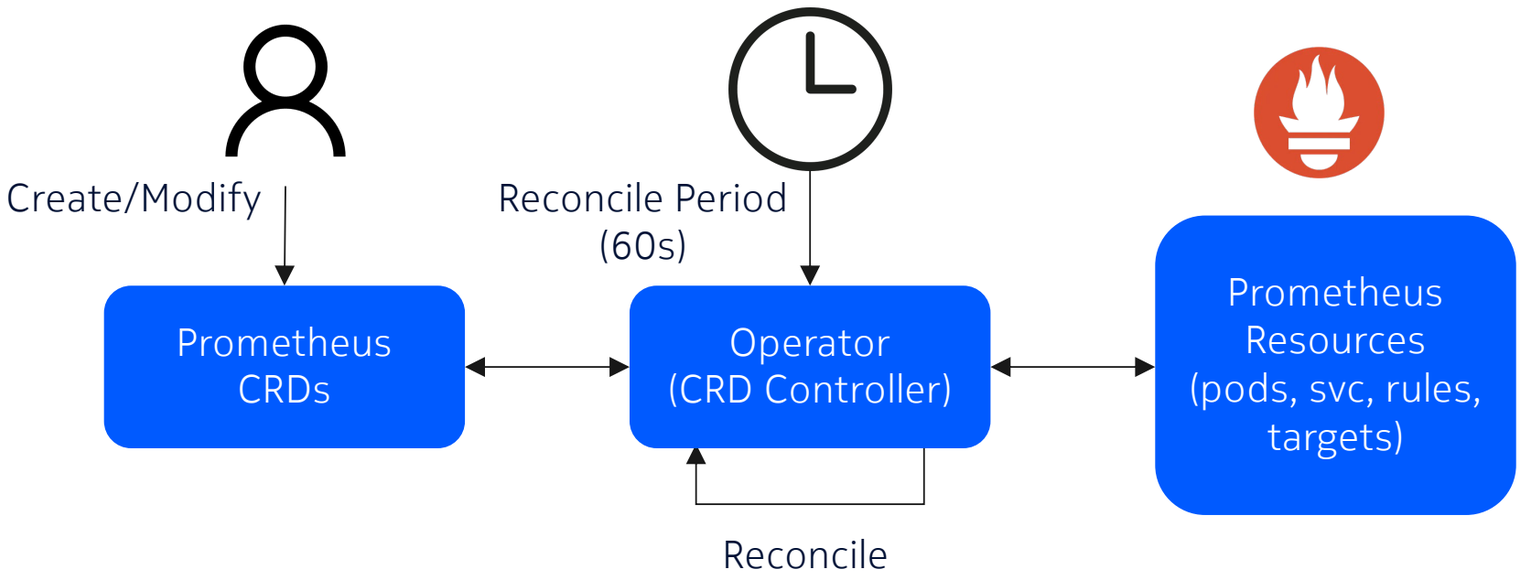
NetOpsKube



Kubernetes Operator

NetOpsKube

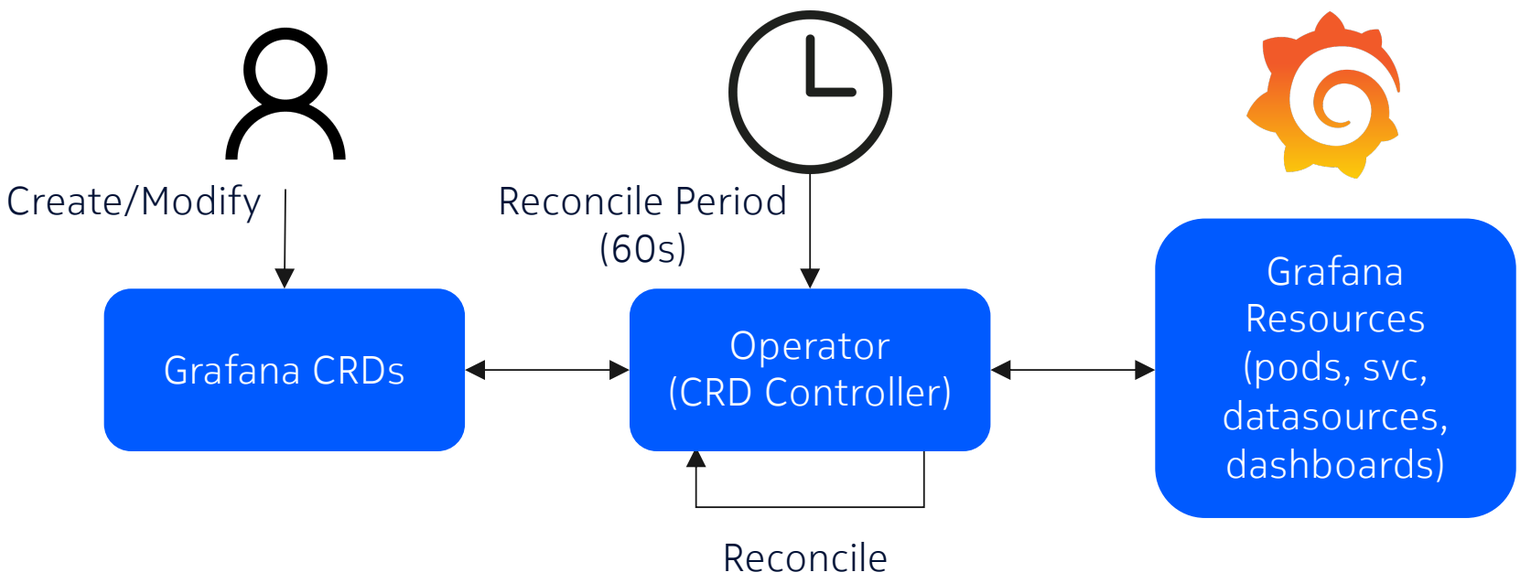
- Configure the fundamentals of Prometheus like versions, persistence, retention policies, and replicas from a native Kubernetes resource.
- Automatically generate monitoring target configurations based on familiar Kubernetes label queries; no need to learn a Prometheus specific configuration language.



Kubernetes Operator

NetOpsKube

- Simplifies the processes of installing, configuring, and maintaining Grafana and its resources.
- To manage resources using infrastructure as code or using GitOps workflows



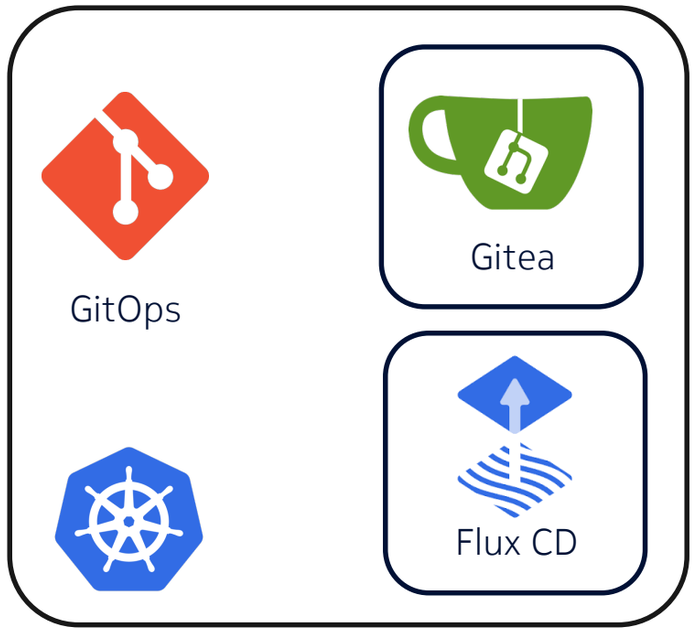
Advanced Observability

Demo

GitOps & Configuration

GitOps Components

NetOpsKube

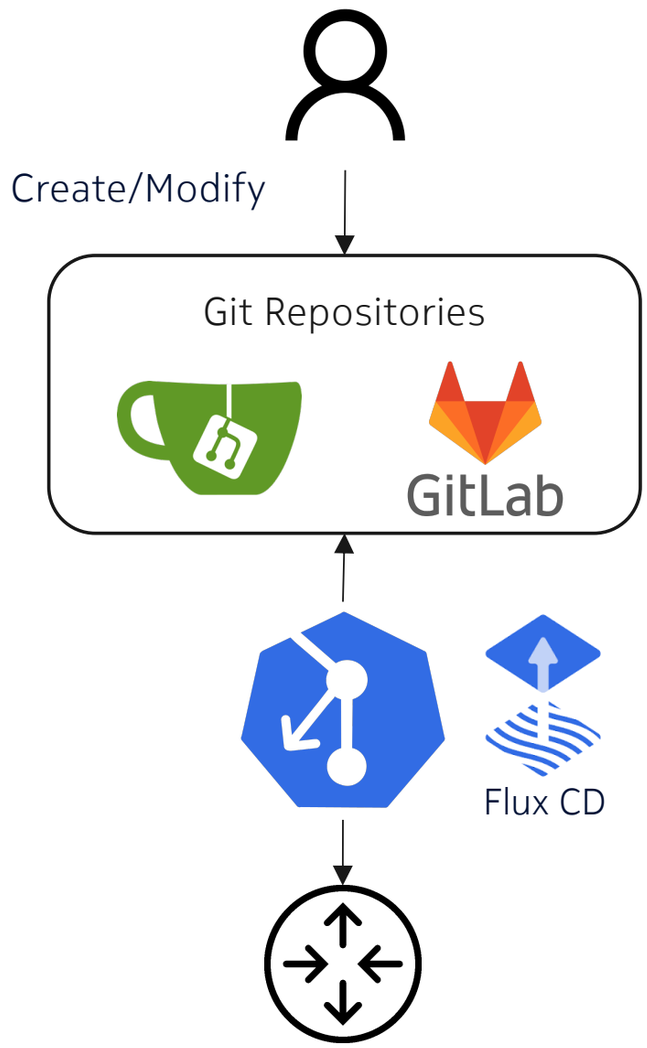


- **Gitea:** A lightweight, self-hosted Git service for hosting code and managing software development projects efficiently (Git Repositories)
- **Flux:** A GitOps tool that automatically synchronizes App States on Kubernetes with configurations stored in a Git repository.

Core Principles of GitOps

NetOpsKube

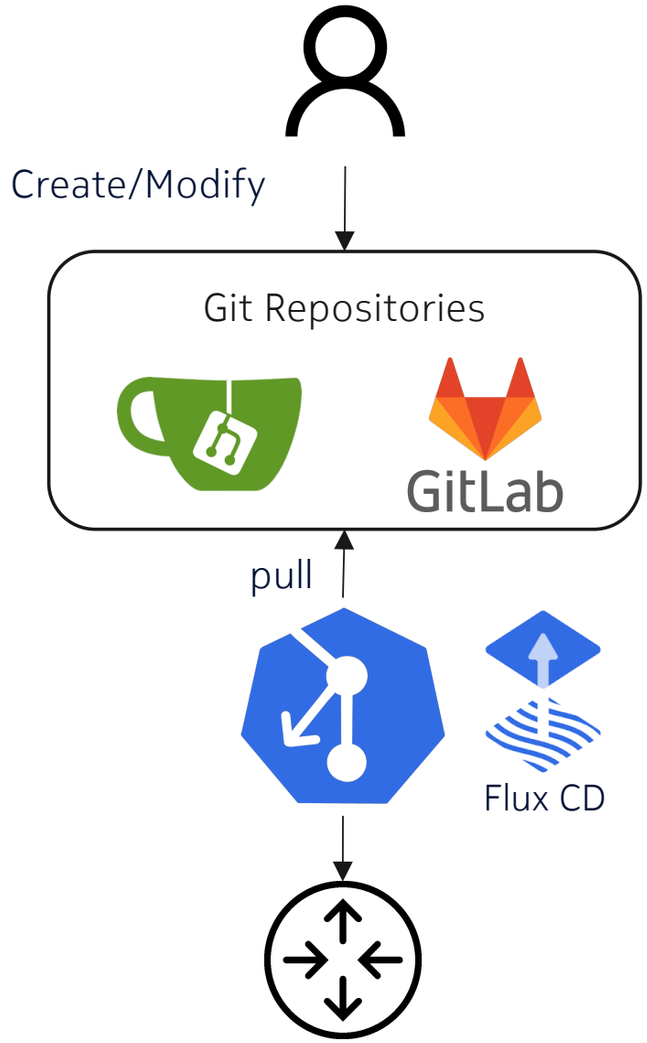
- **Single Source of Truth:** All stored in a Git repository.
- **Declarative Configuration:** You define the desired state.
- **Version Controlled & Auditable:** Every change has a history, making it easy to track who changed what and perform **quick rollbacks** if needed.
- **Continuous Reconciliation:** Constantly compare the live environment with the state defined in Git and automatically **correct** any differences (configuration drift).



How FluxCD Works

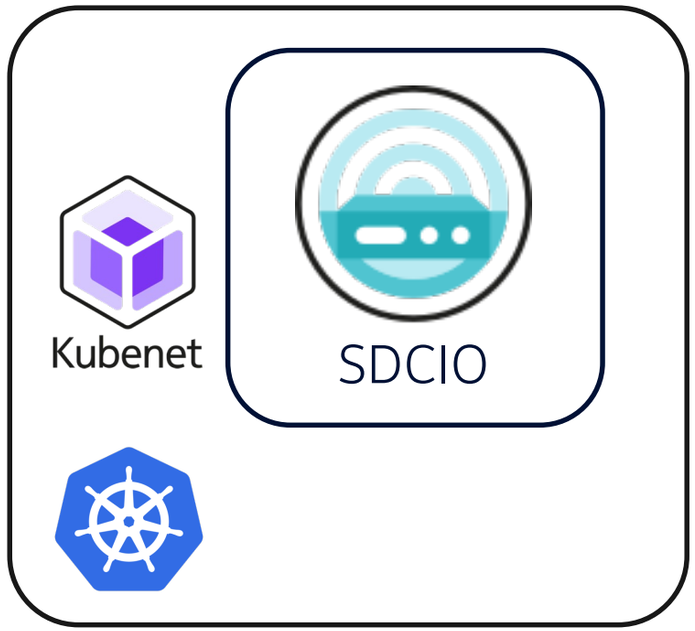
NetOpsKube

- **Pull-Based Model**: Flux "**pulls**" changes from Git, which is more secure as the cluster doesn't need to expose credentials to external systems.
- **Automated Syncing**: Flux monitors your Git or Helm repositories and automatically synchronizes the cluster state whenever it detects a **new commit**.
- **Self-Healing**: If someone manually modifies a resource in the cluster, Flux will **detect the drift and revert it back to the state** defined in Git.



Schema Driven Configuration

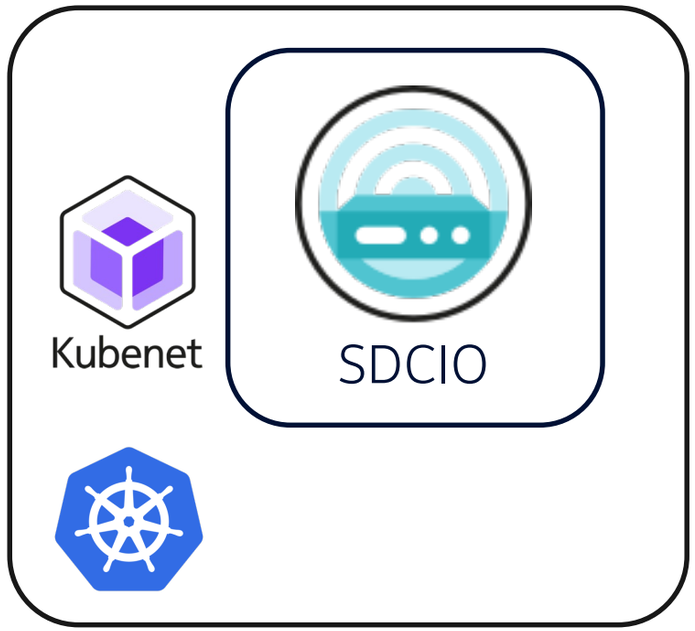
NetOpsKube



- Cloud Native Network Management
- Developed at Nokia, open sourced 2024, available on GitHub
- Community
  - Discord channel for latest news
- Aiming to become a CNCF project

SDCIO - Conf Management Components

NetOpsKube

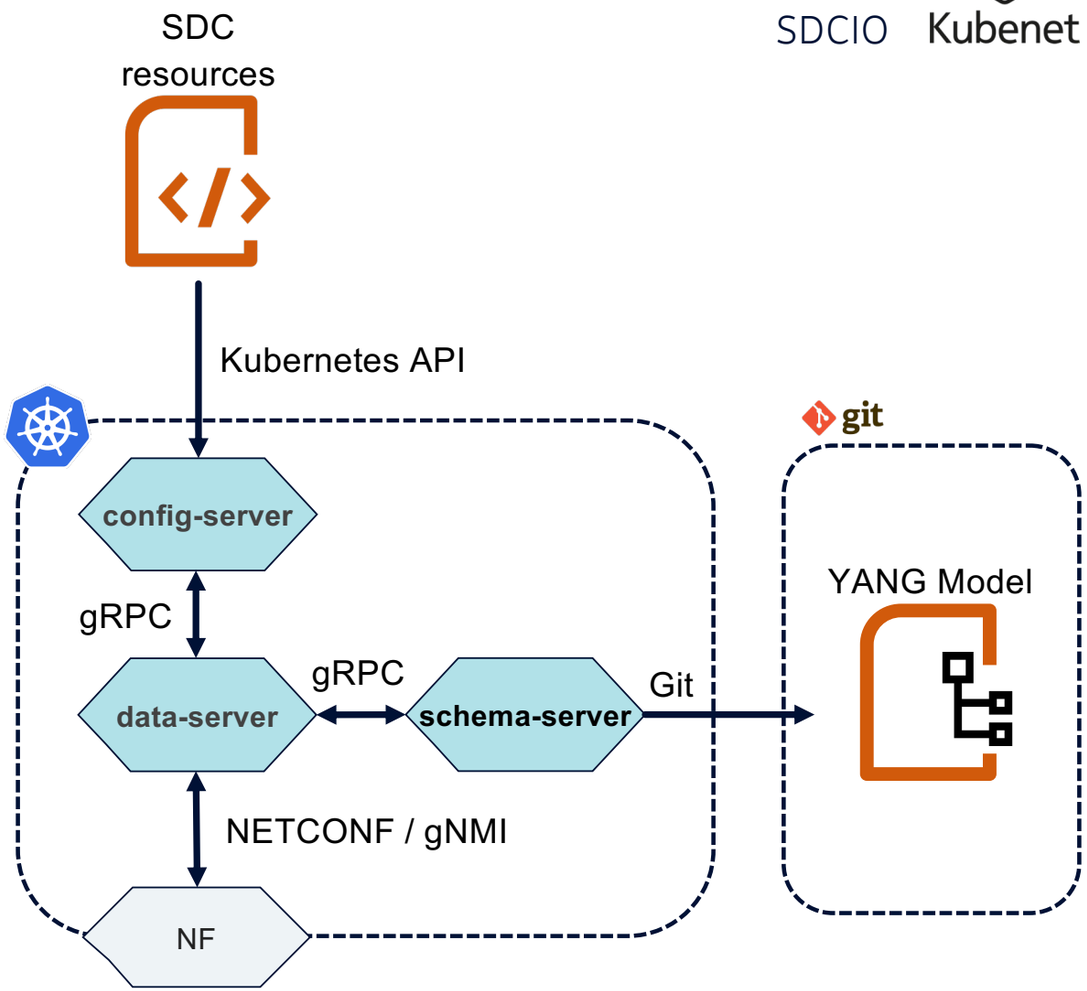


- YANG
- Targets: Physical devices (PNF), Containers (CNF), Virtual Machines (VNF)
- Vendor agnostic
- Flexible deployments: Small, medium or large scale scaled out deployments
- Target Protocols: gNMI, Netconf
- Full Config or Config Snippets
- Declarative Operation
- Deviations (revertive/non revertive)

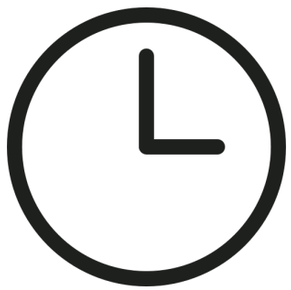
SDCIO: Components

SDCIO Kubenet

NetOpsKube



- Config-Server
  - Aggregated API Server
- Data-server
  - Communicate with NF, maintain in-memory config tree, validation
- Schema-Server
  - path-based schema element store



NetOpsKube

GitOps and Configuration

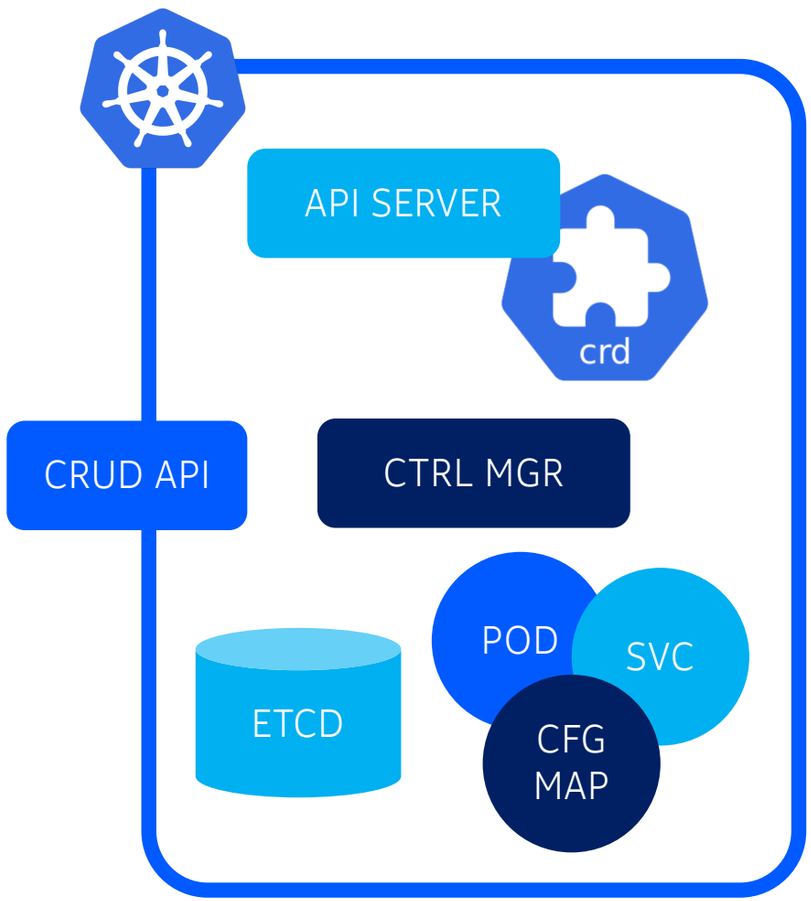
Demo

Kubernetes Benefits

Why Kubernetes?

NetOpsKube

Kubernetes APIs and Resources

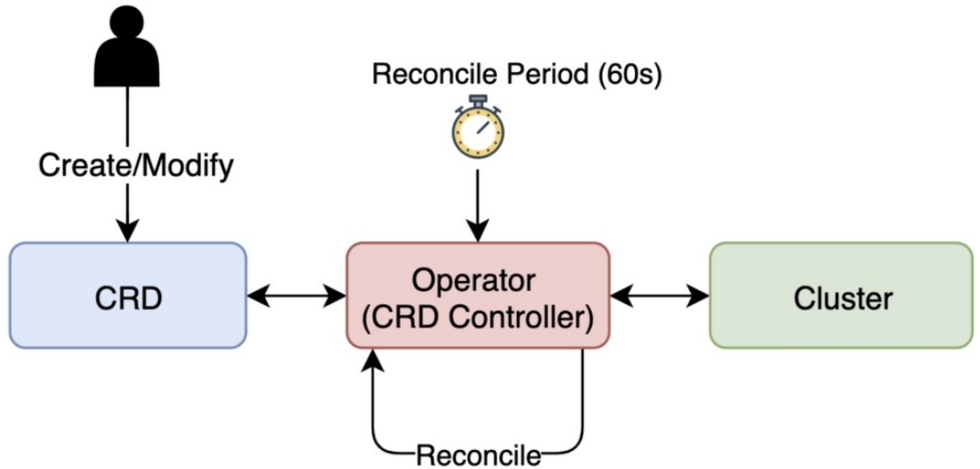


- **APIs Define Resources**
  - APIs define objects in Kubernetes
- **Resources are Key Elements**
  - Pods, Services, ConfigMaps, etc.
- **CRUD Operations**
  - Create, Read, Update, Delete (CRUD) actions for resources
- **Extensive FrameWork**
  - Kubernetes can be extended with custom resources (CRDs).

How KRM Uses Controllers

NetOpsKube

- **Controllers Automate Actions**
  - Controllers handle everything behind the scenes.
- **Continuously Reconcile State**
  - Constantly check and adjust resources.
- **No Direct etcd Access**
  - Controllers communicate via APIs, not directly with data stores.
- **Level-Triggered System**
  - Kubernetes responds to changes at any time.



ConfigMap vs K8s Operator

NetOpsKube

ConfigMap

It works like a file. App needs to restart to load the new changes

VS

CRD/Operator

Operator reconcile gently load changes. No manual restart.

App Components in Kubernetes Cluster

(if we still have enough time left in this tutorial...)

Or time for Q&A

(if we don't)

Public Git Repositories

Org: github.com/CSPDevLabs

NetOpsKube

NetOpsKube → **Playground**: Makefile, Kind Config

kpt → **Kubernetes Packages**: Base, GitOps and BNG packages.

nok-clabs → **Clab use cases & Manifests**: Topo files, startup configs, sdcio, gnmic and other tools configurations related to use cases

nok-controller → **Top level Kubernetes reconciler for Network Elements**

nok-crd-metrics → **Kubernetes reconciler to export CRD Metrics (i.e. SDCIO)**

Base Packages (KPT)

NetOpsKube

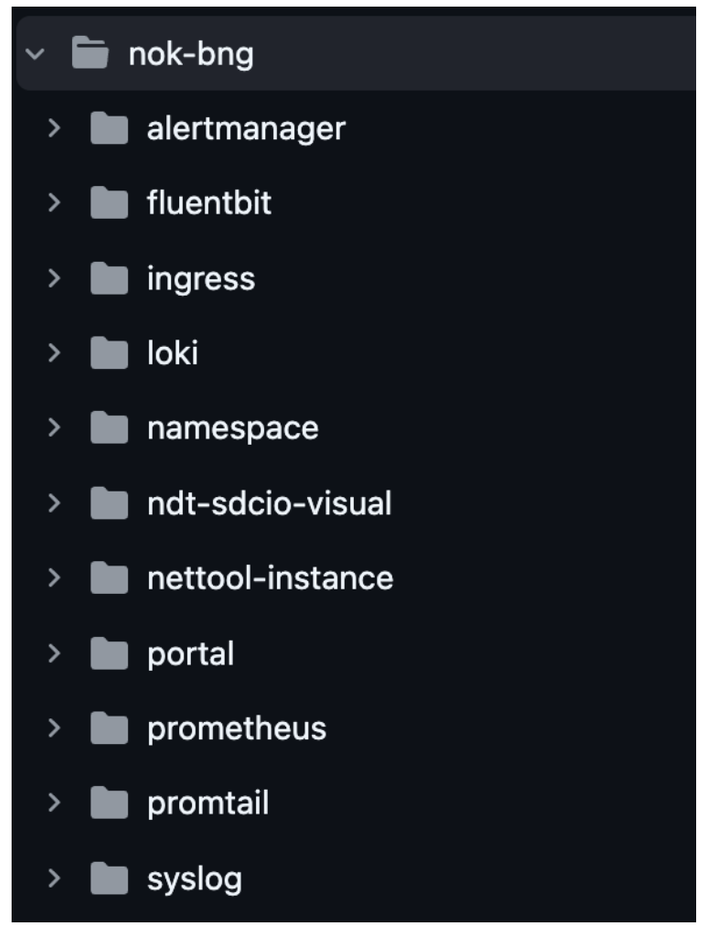
cert-manager	**Shared digital certificates management services (i.e. SDCIO, gnmic)**
grafana-operator	**Grafana Operator CRDs and Reconciler setup manifests**
ingress	**Deploy ngnix controller**
metallb	**MetalLB CRDs and reconciler**
nok-controller	
nok-crd-metrics	
sdcio	**KubeNet SDCIO CRDS, Config Server and Reconciler**

Use Case Package (nok-bng)

Each use case have its own namespace

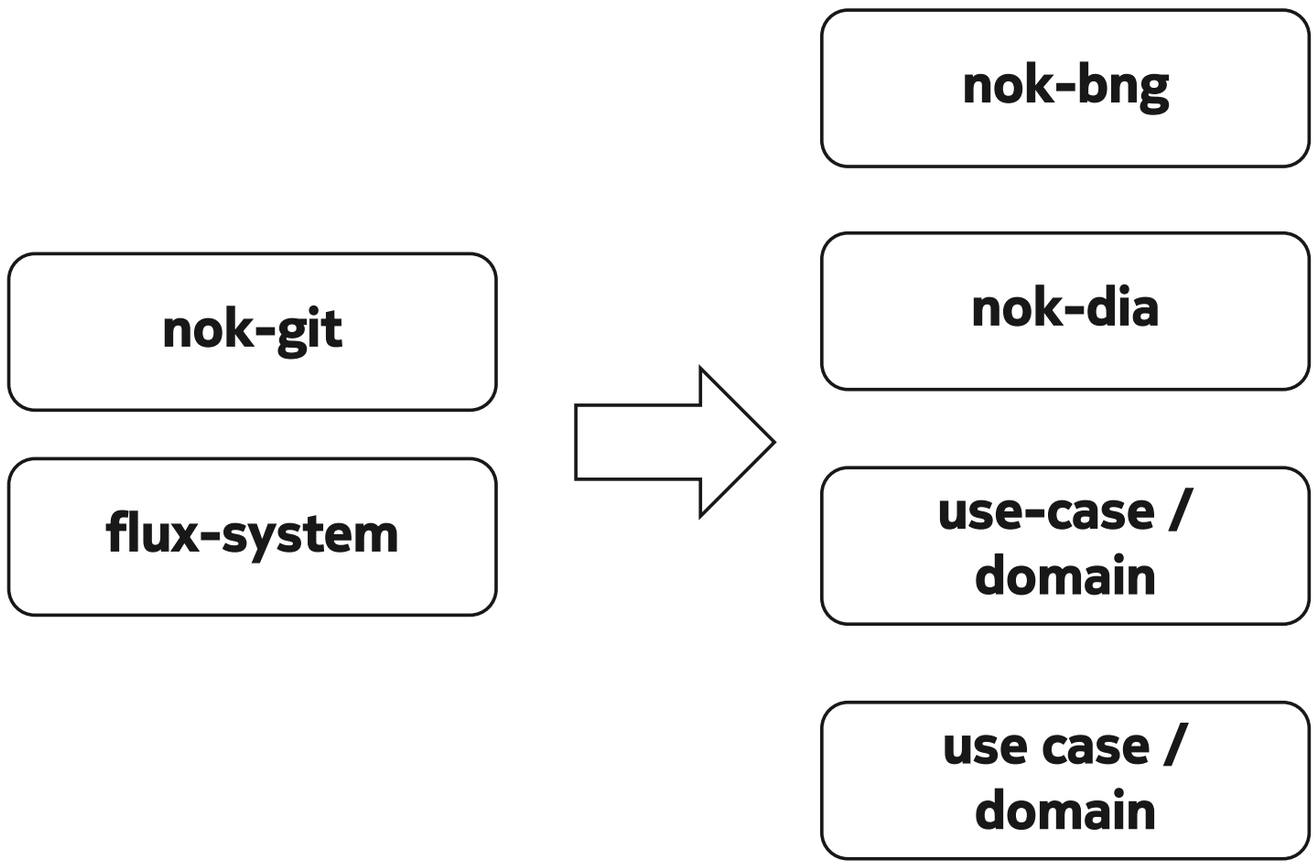
NetOpsKube

Base package



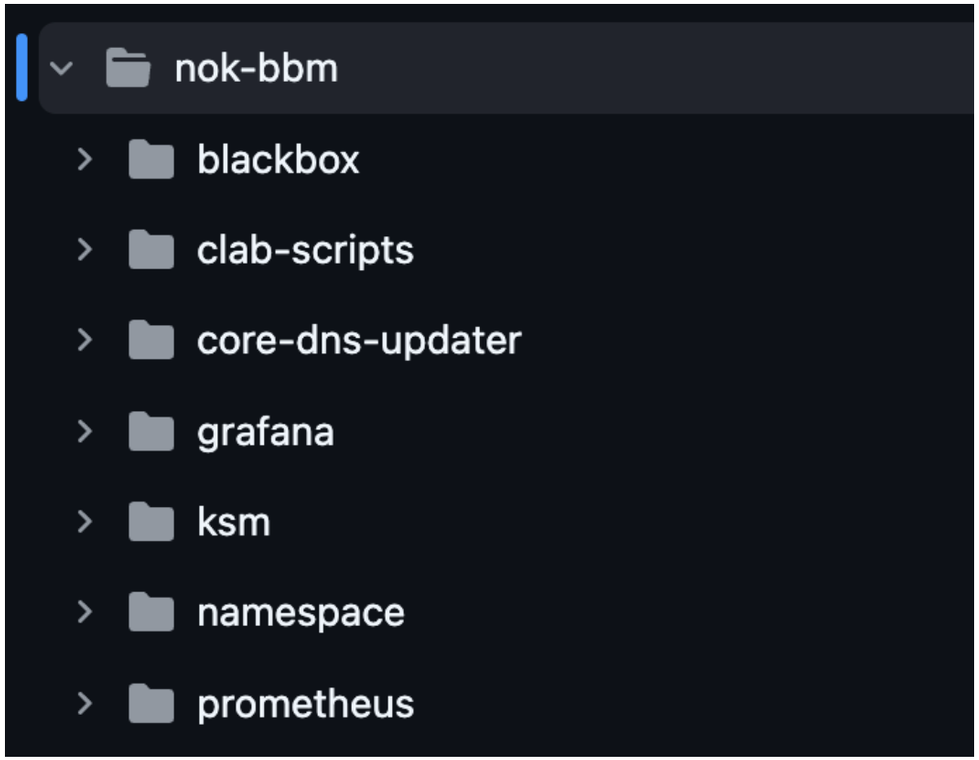
Other packages and namespaces

NetOpsKube



BBM: Self-monitoring

NetOpsKube



Monitoring Network Targets, Kubernetes Cluster and Pod/Services

NetOpsKube Tutorial

NetOps Stack for Core/Aggr of Any Size

Mauricio (Mau) Rojas

Broadband Network Gateway (BNG) part 2

BNG Role in the Network

NetOpsKube

- **Standards-Based Interoperability:** BNGs follow industry standards (Broadband Forum TR-101/TR-459), allowing service providers to mix hardware and software from multiple vendors
- **Subscriber Aggregation Point:** Acts as the primary gateway where diverse access technologies (Fiber, Copper, Wireless) converge to access the Service Provider's core network.
- **Session Management:** Identifies and manages individual user sessions using standard protocols like PPPoE or IPoE to establish connectivity.
- **Policy Enforcement:** Applies Quality of Service (QoS) and bandwidth throttling on a per-subscriber basis to ensure fair usage and tiered service levels.

Broadband Network Gateway (BNG) part 3

Logical Functions and Standards

NetOpsKube

- **Authentication & Accounting:** Interfaces with central AAA servers (RADIUS) to verify user credentials and track data usage for billing.
- **IP Address Management:** Dynamically assigns and manages IP addresses for end-user devices via DHCP or local address pools.
- **Control and User Plane Separation (CUPS):** Follows industry-standard architectures to decouple subscriber management logic from high-speed data forwarding, enabling independent scaling of each.