



KubeCon

North America 2017

Using Custom Resources to Provide Cloud Native API Management

Frank B Greco Jr, Cloud Native Engineer, *Northwestern Mutual*





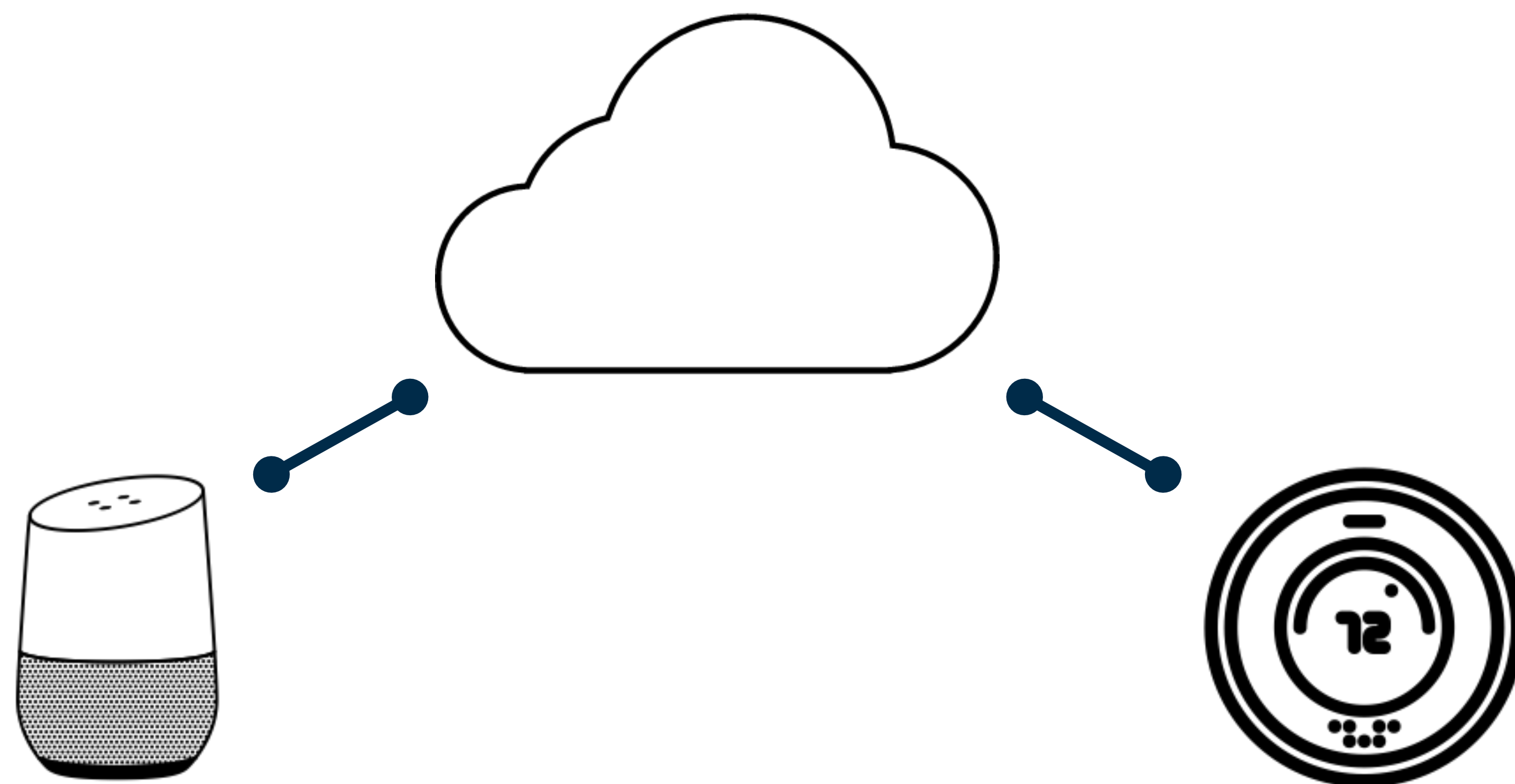
APIs

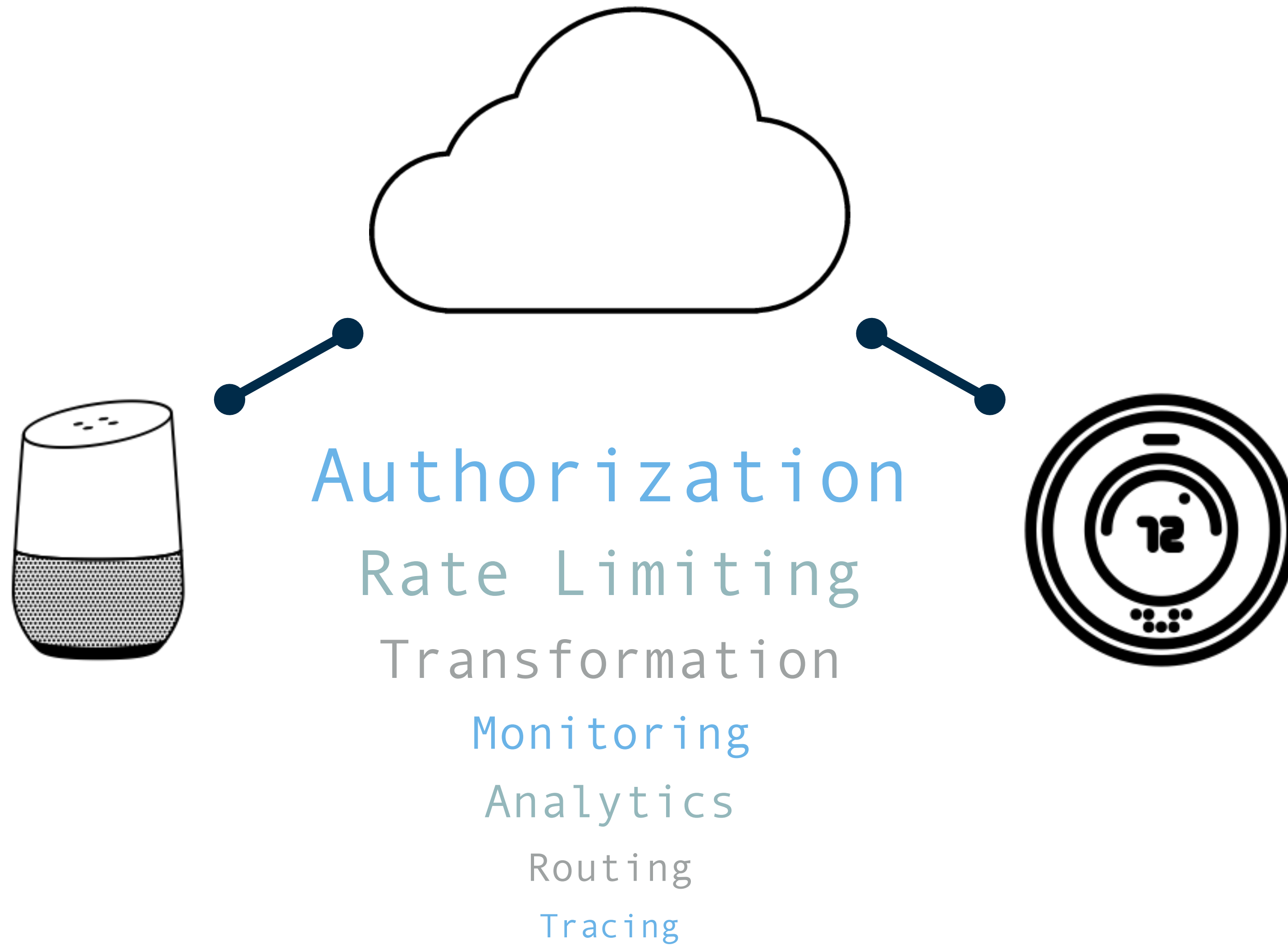


frankgreco



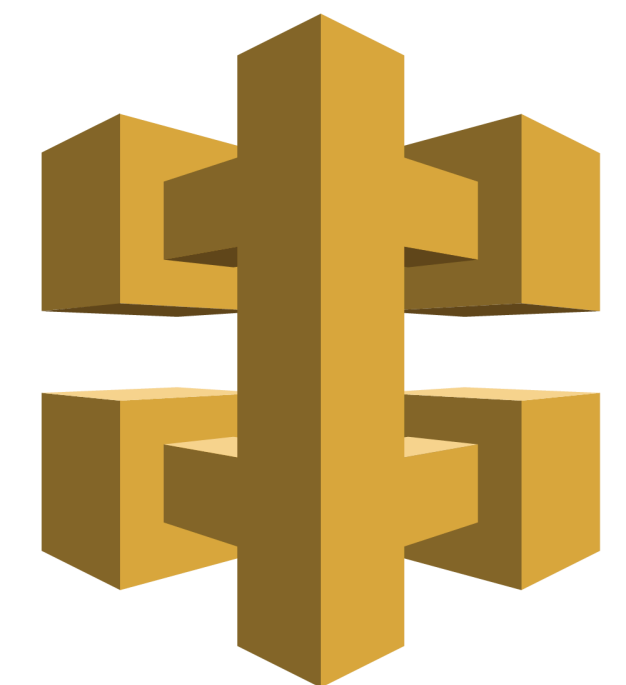
@fbgrecojr







apigee





History

vpc

ec2

elb



frankgreco



@fbgrecojr

vpc

ec2

elb



frankgreco

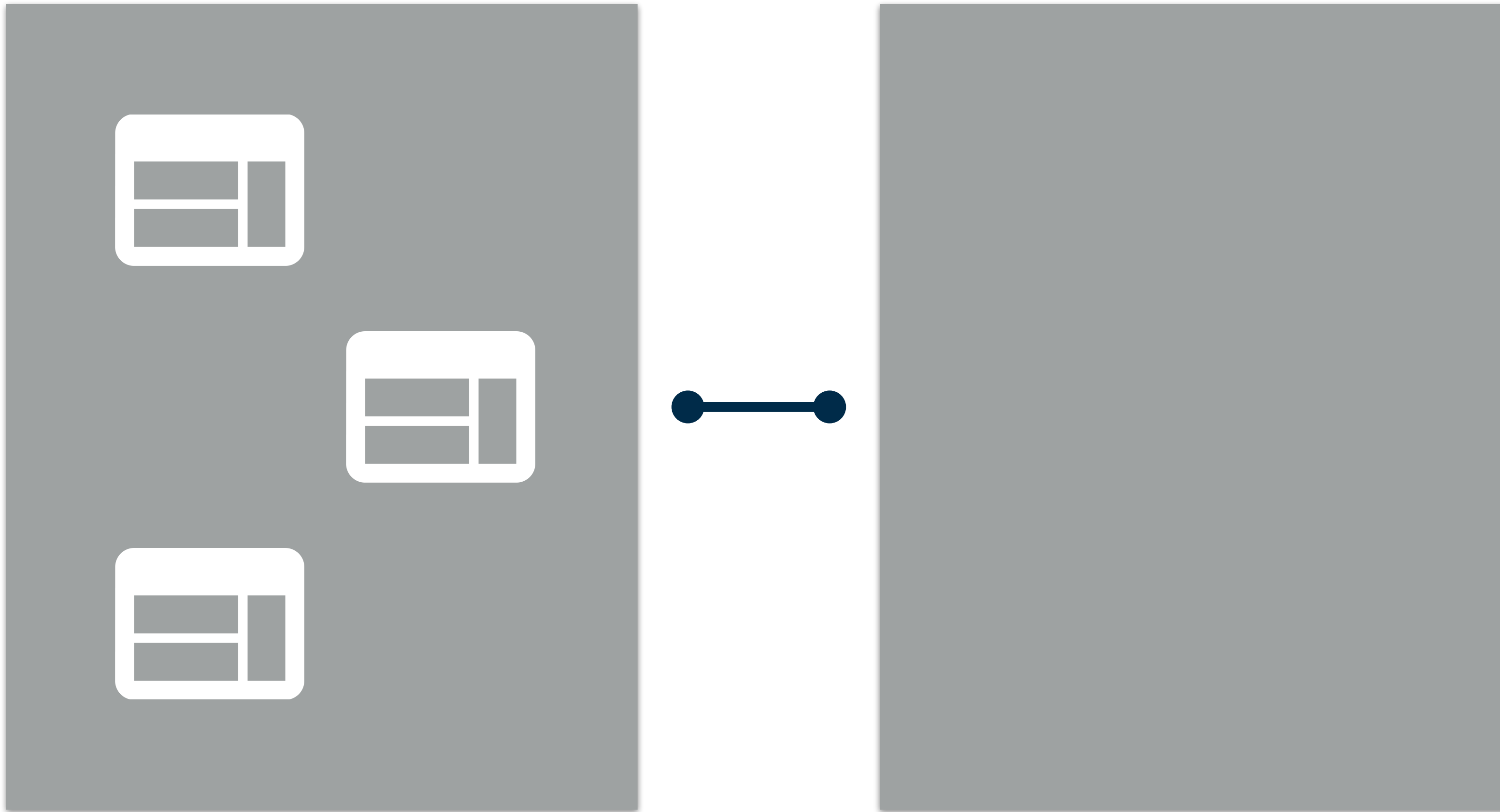


@fbgrecojr

vpc

ec2

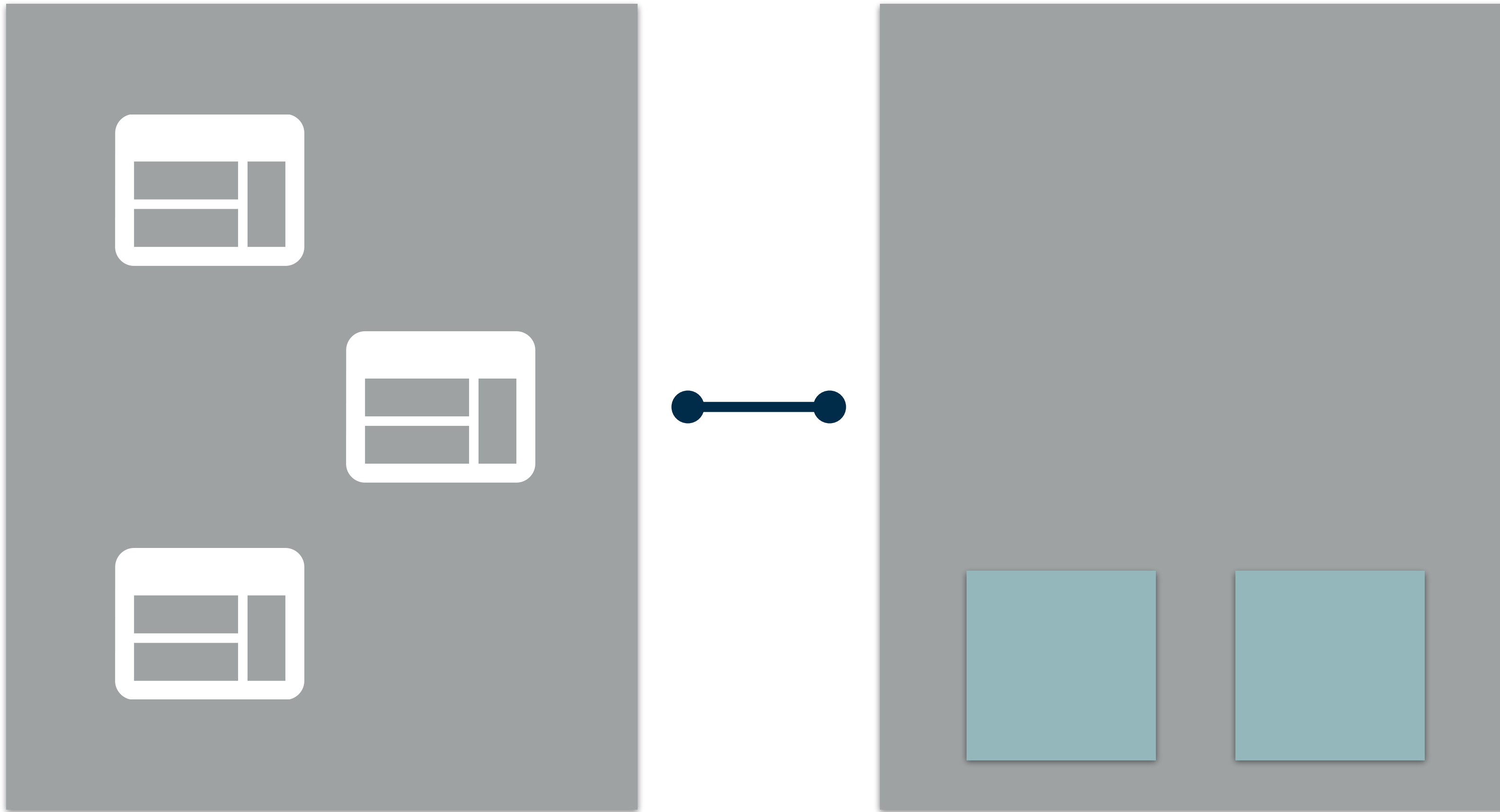
elb



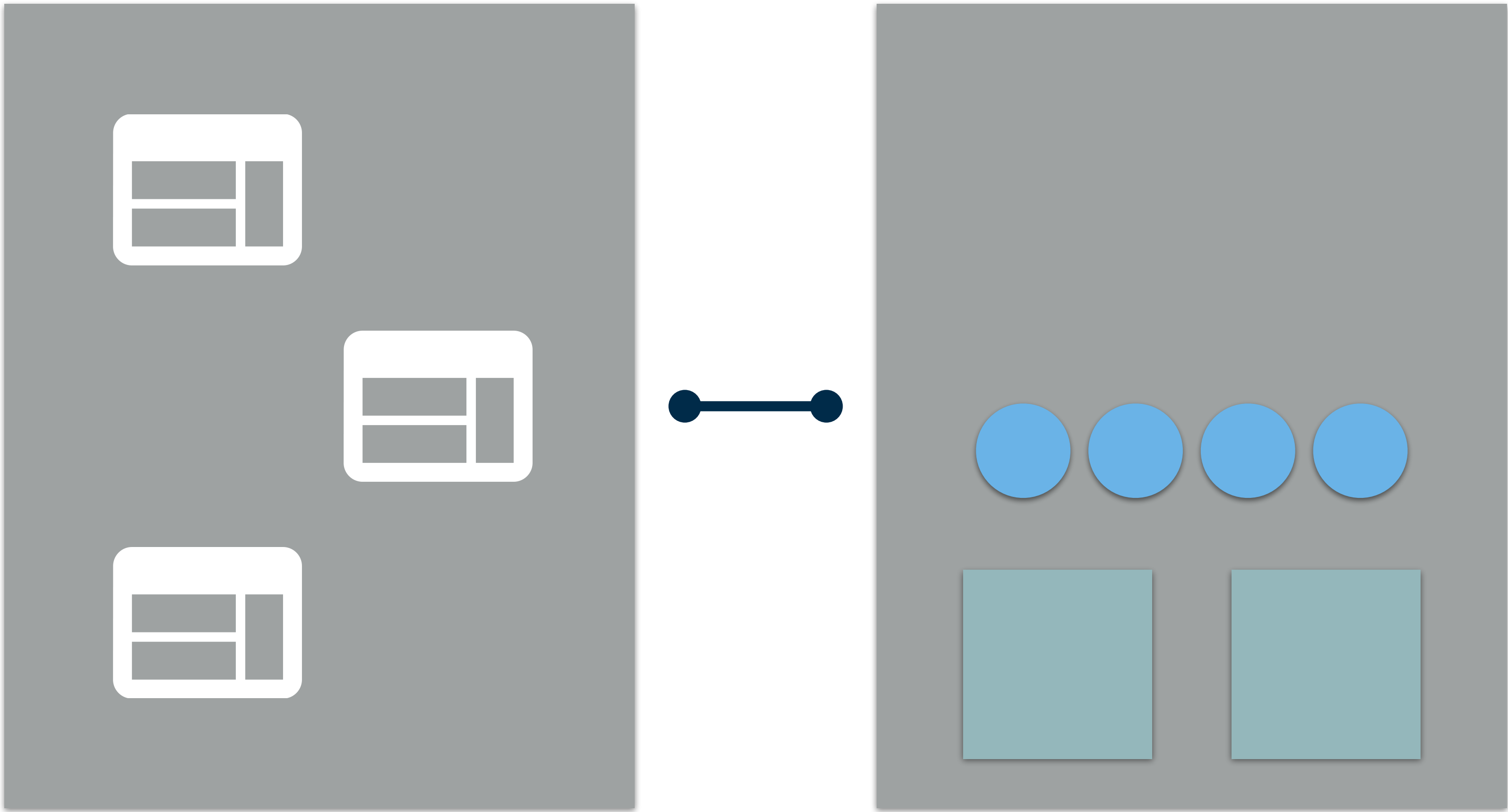
vpc

ec2

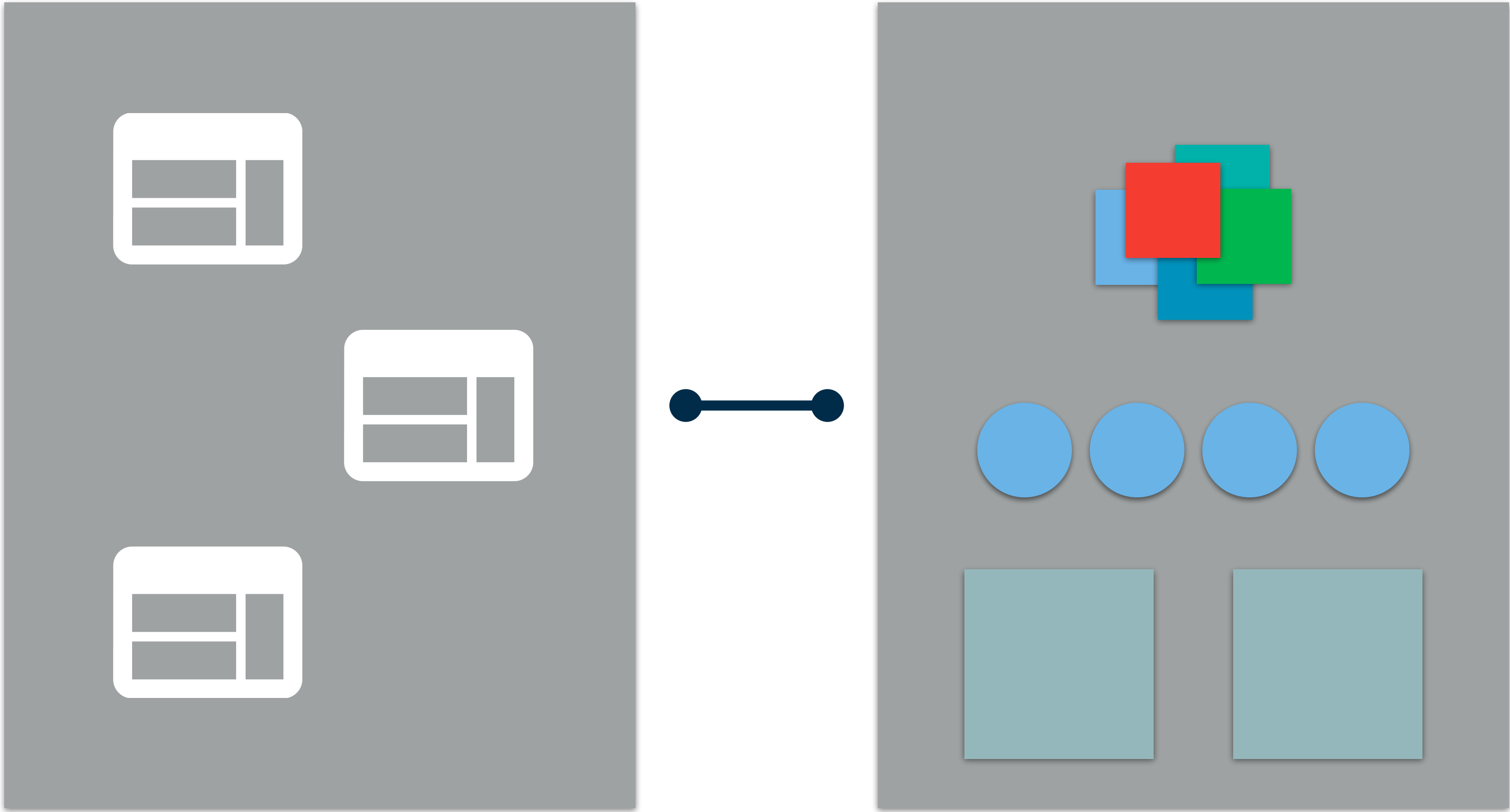
elb



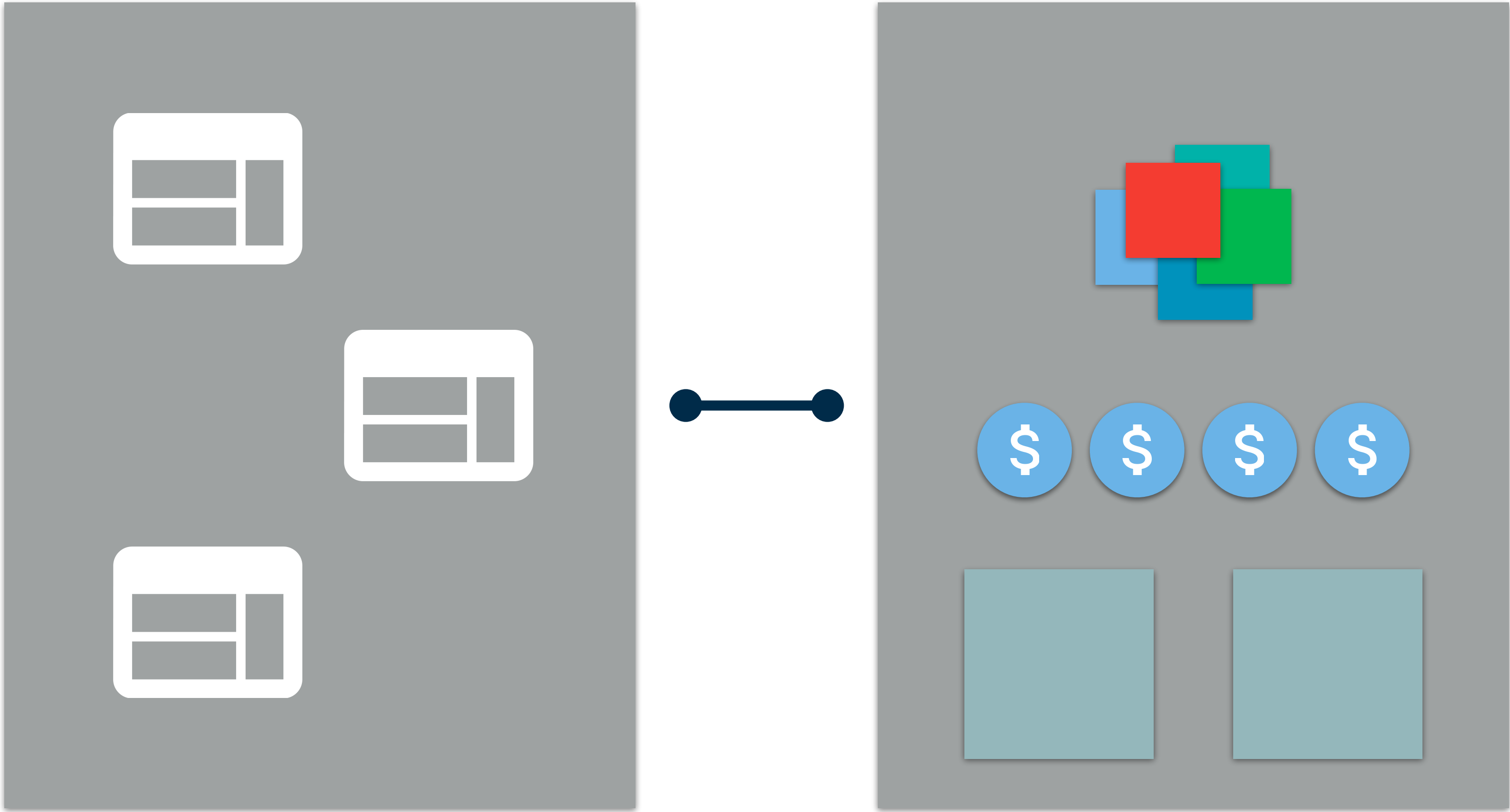
vpc ec2 elb



vpc ec2 elb



vpc ec2 elb

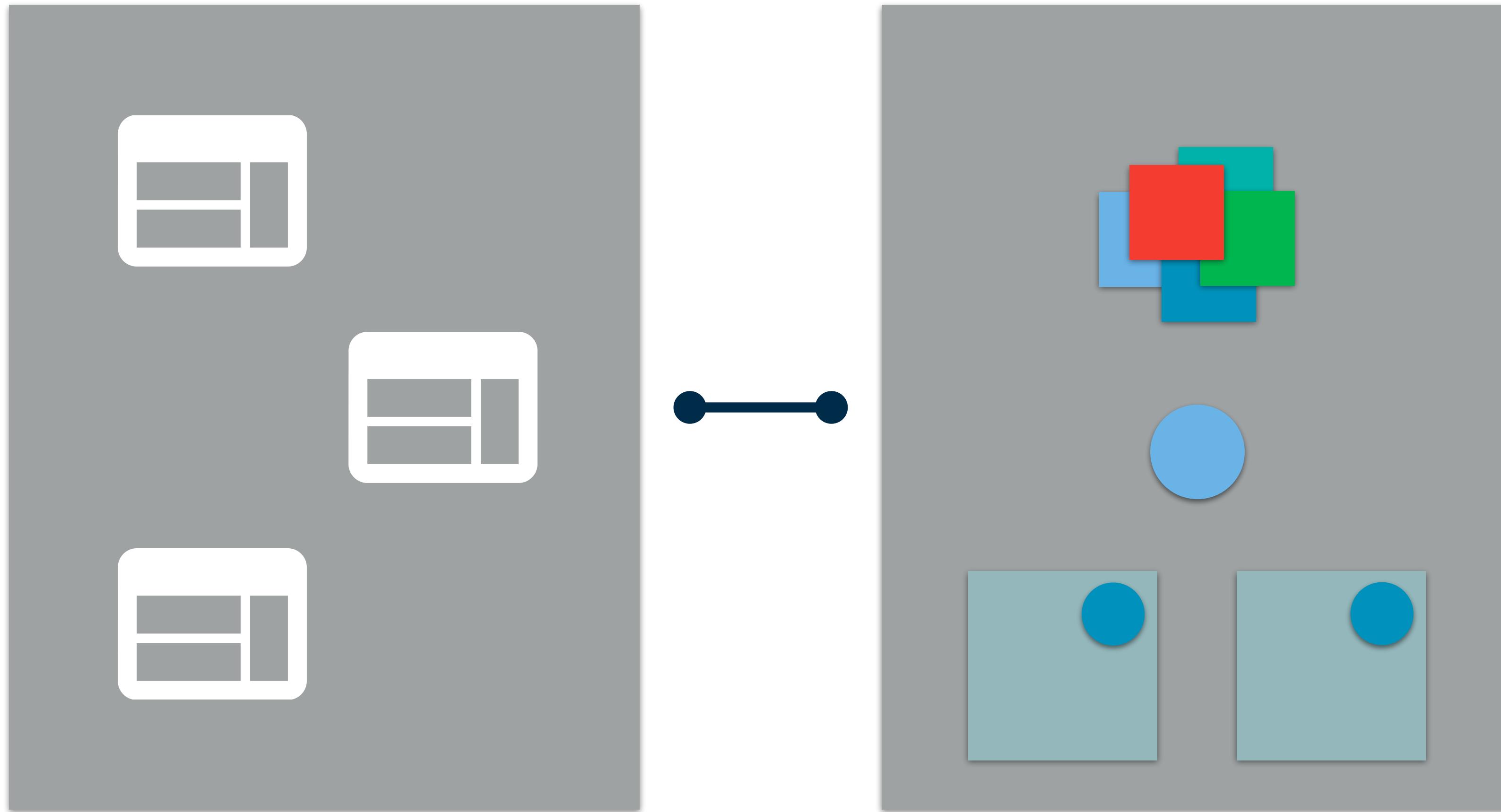


vpc

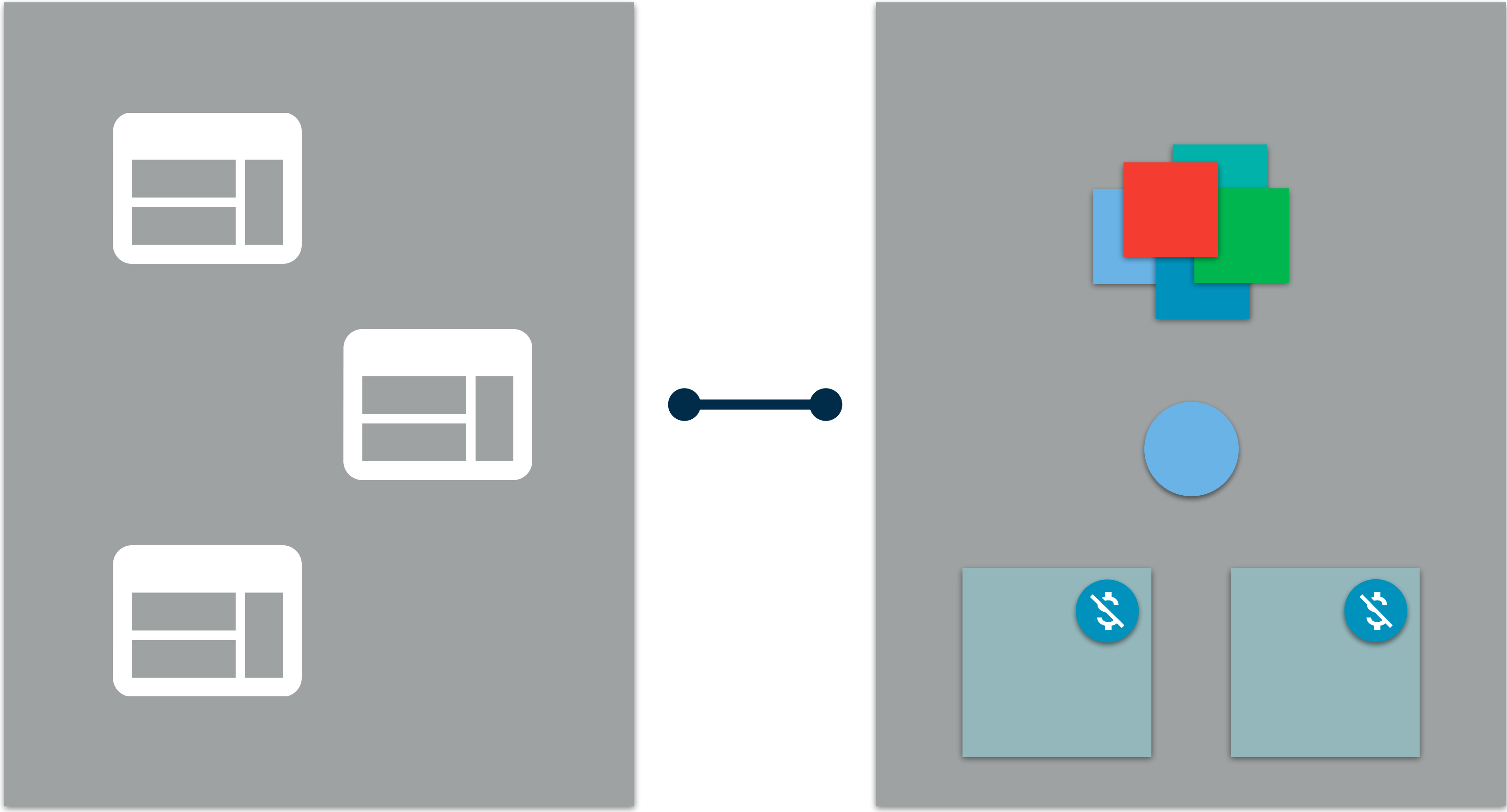
ec2

elb

ingress



vpc ec2 elb ingress





Innovative

Innovative
Lightweight

Innovative
Lightweight
Performance centric



Innovative

Lightweight

Performance centric

Robust api key management



Innovative

Lightweight

Performance centric

Robust api key management

Extensible plugin framework



Innovative

Lightweight

Performance centric

Robust api key management

Extensible plugin framework

Kubernetes native



Innovative

Lightweight

Performance centric

Robust api key management

Extensible plugin framework

Kubernetes native

Diverse tooling



Innovative

Lightweight

Performance centric

Robust api key management

Extensible plugin framework

Kubernetes native

Diverse tooling

Easy to automate



Kanali



frankgreco



@fbgrecojr

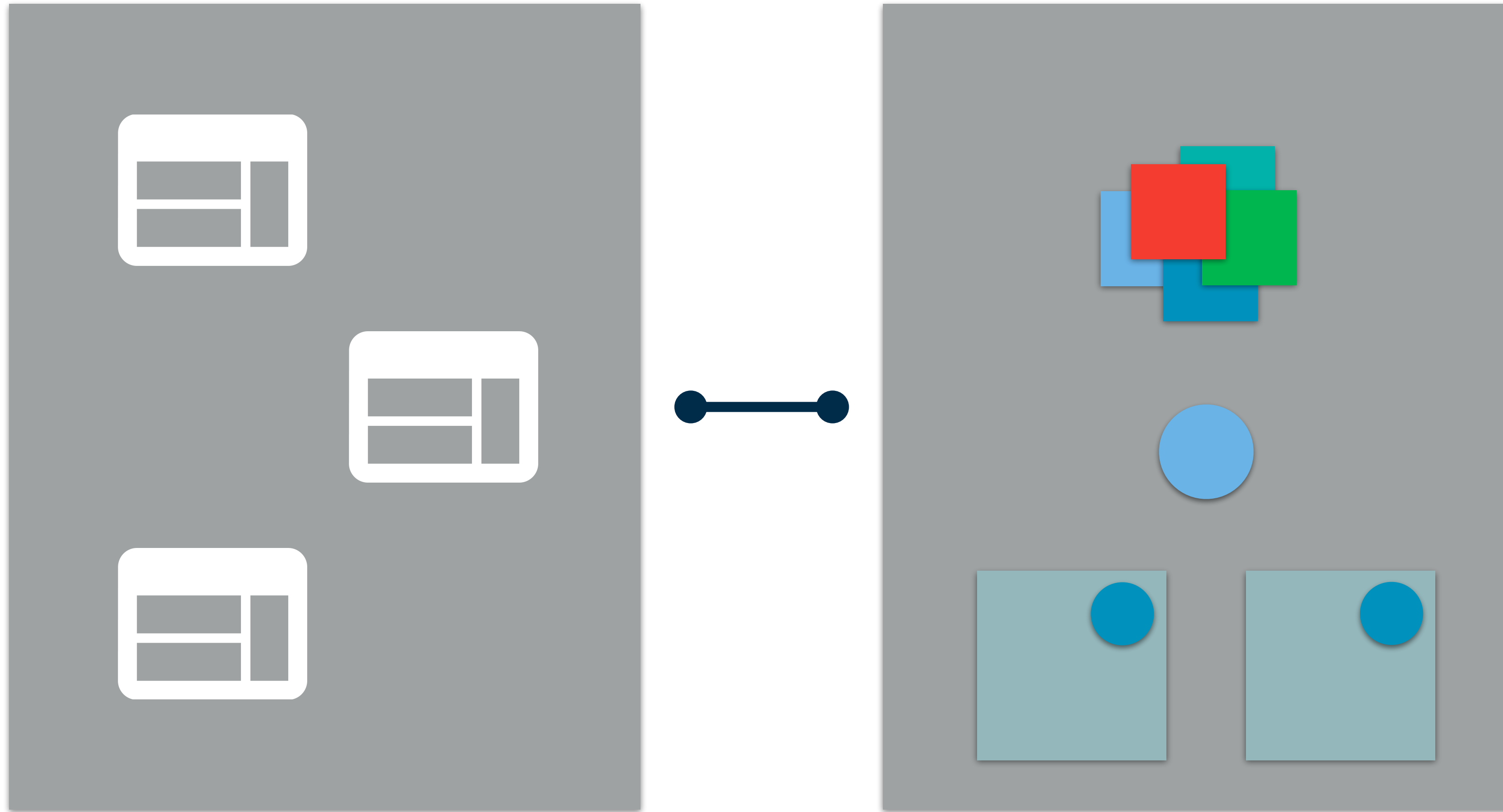
“Kanali is a lightweight,
Kubernetes native API management
gateway that together with network
policies provide a robust, open
source solution to Kubernetes
ingress, API management, and API
security”

vpc

ec2

elb

ingress

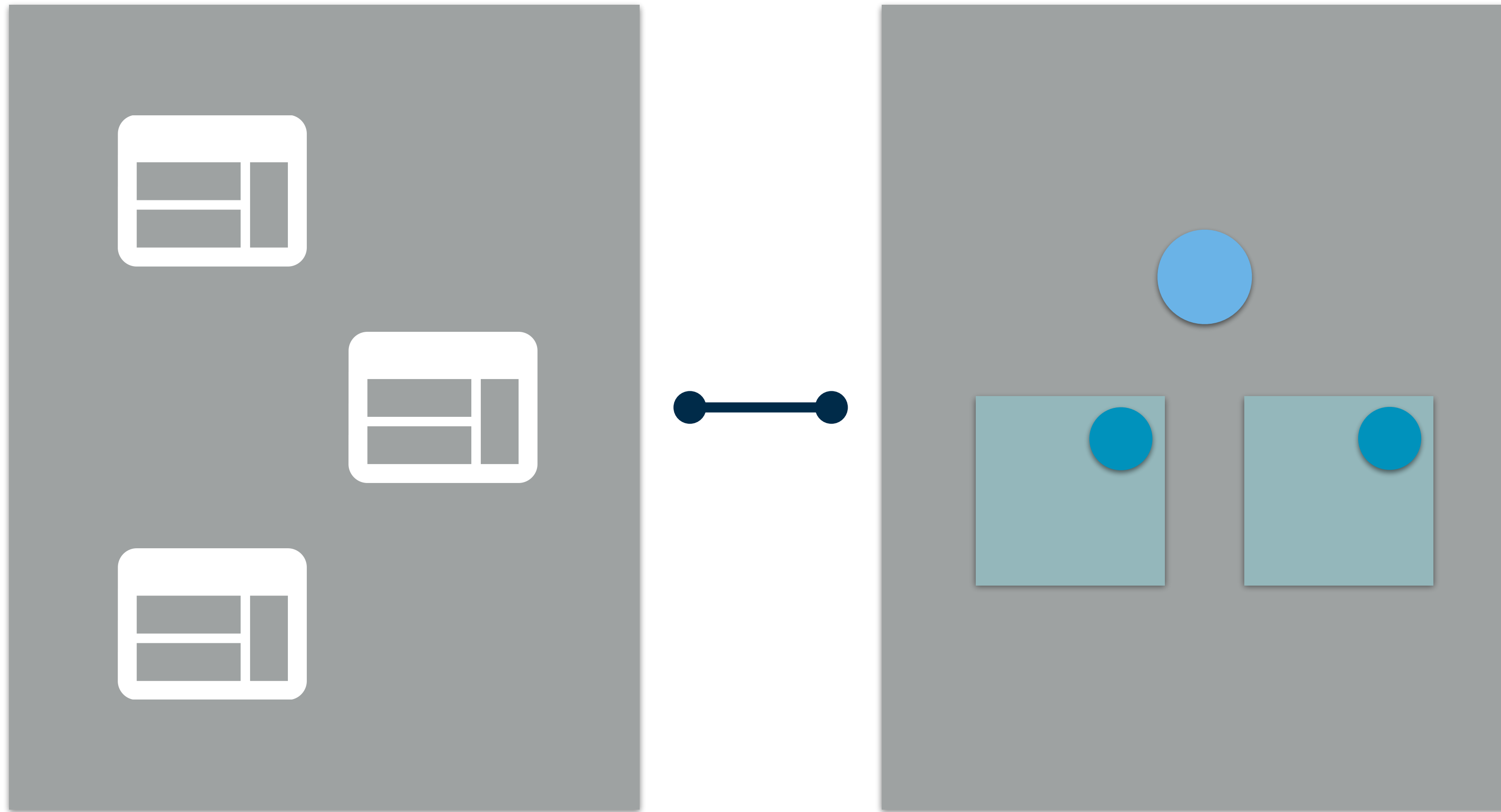


vpc

ec2

elb

kanali

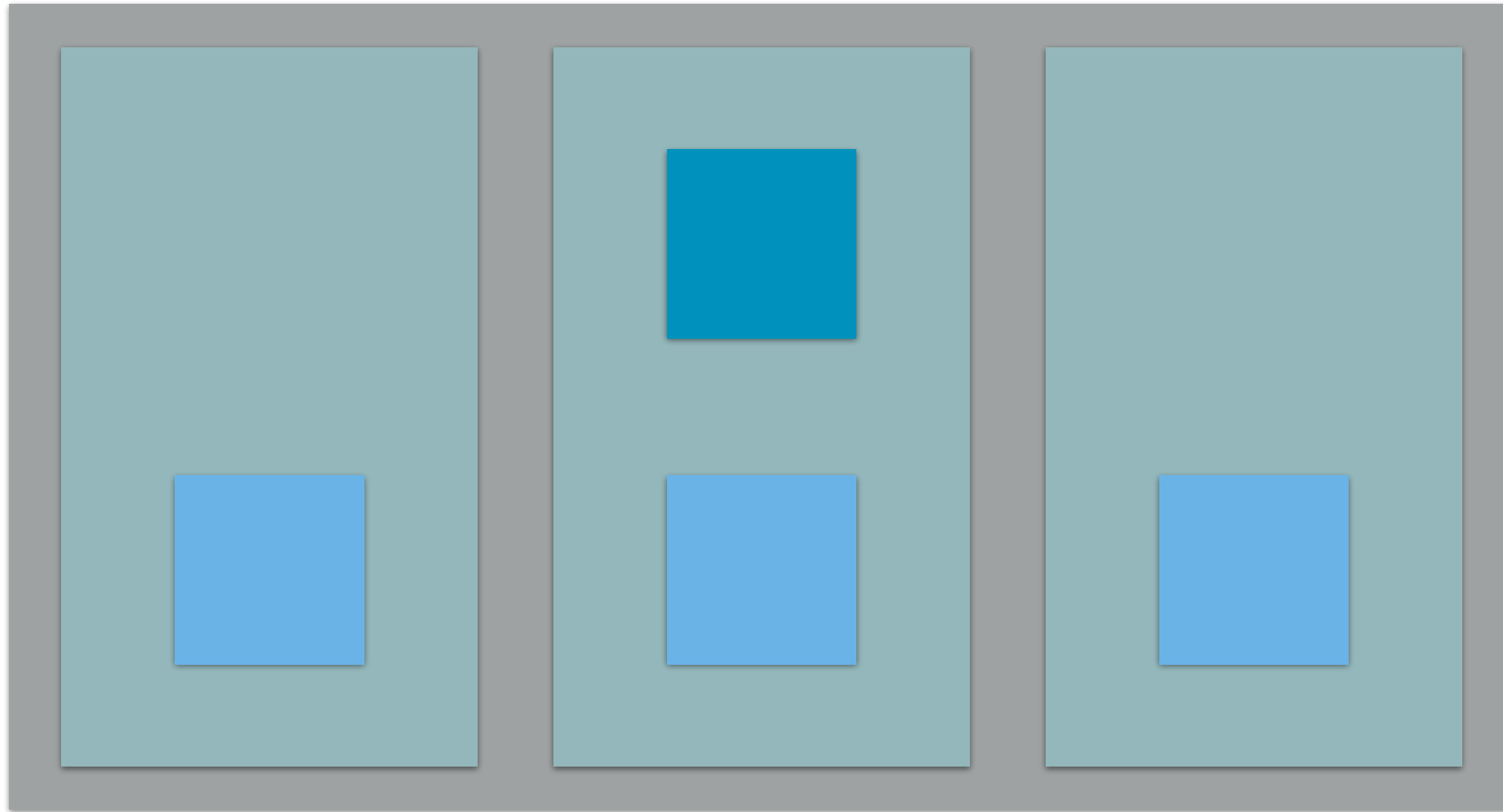


cluster

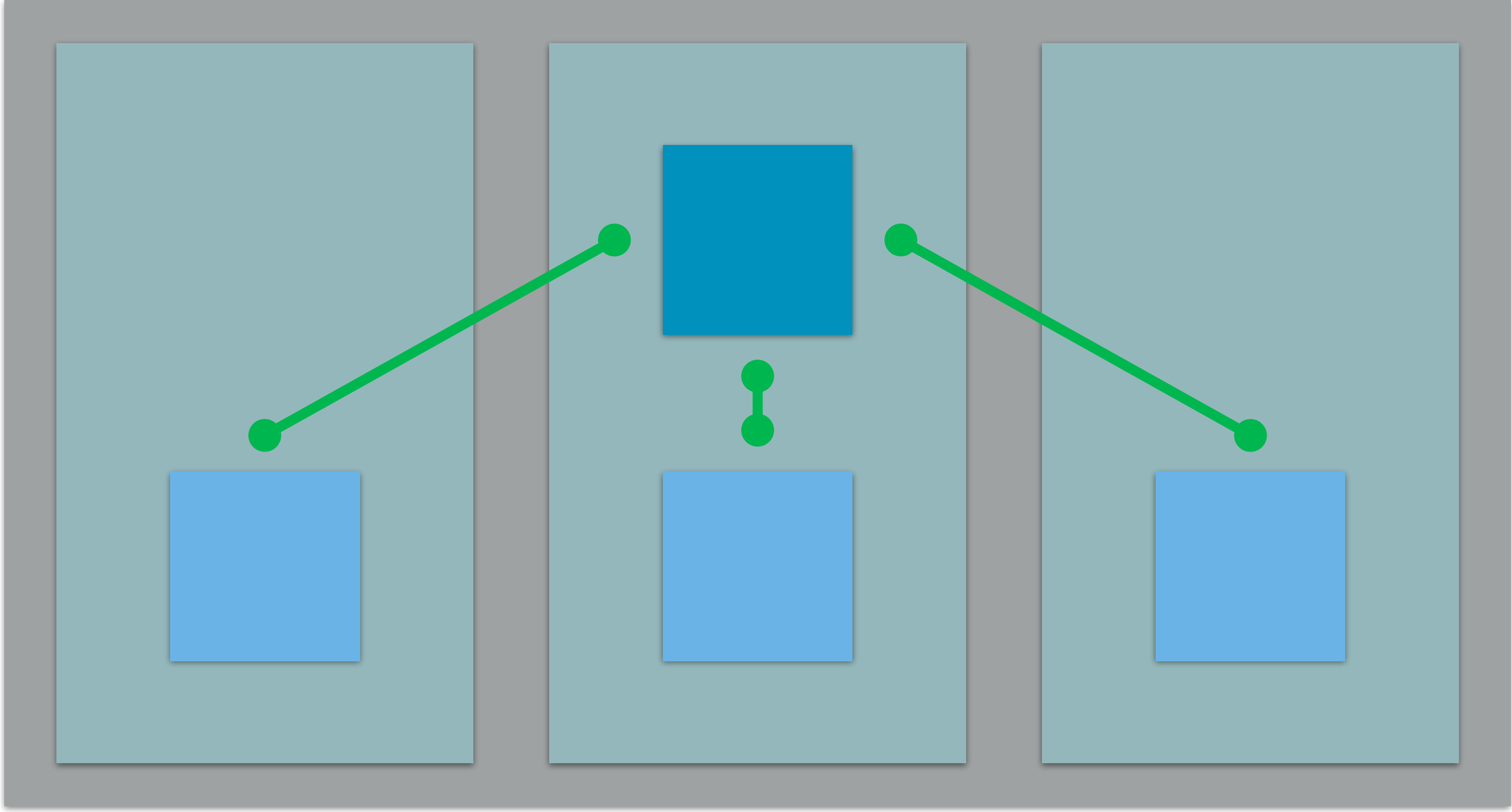
ns

app

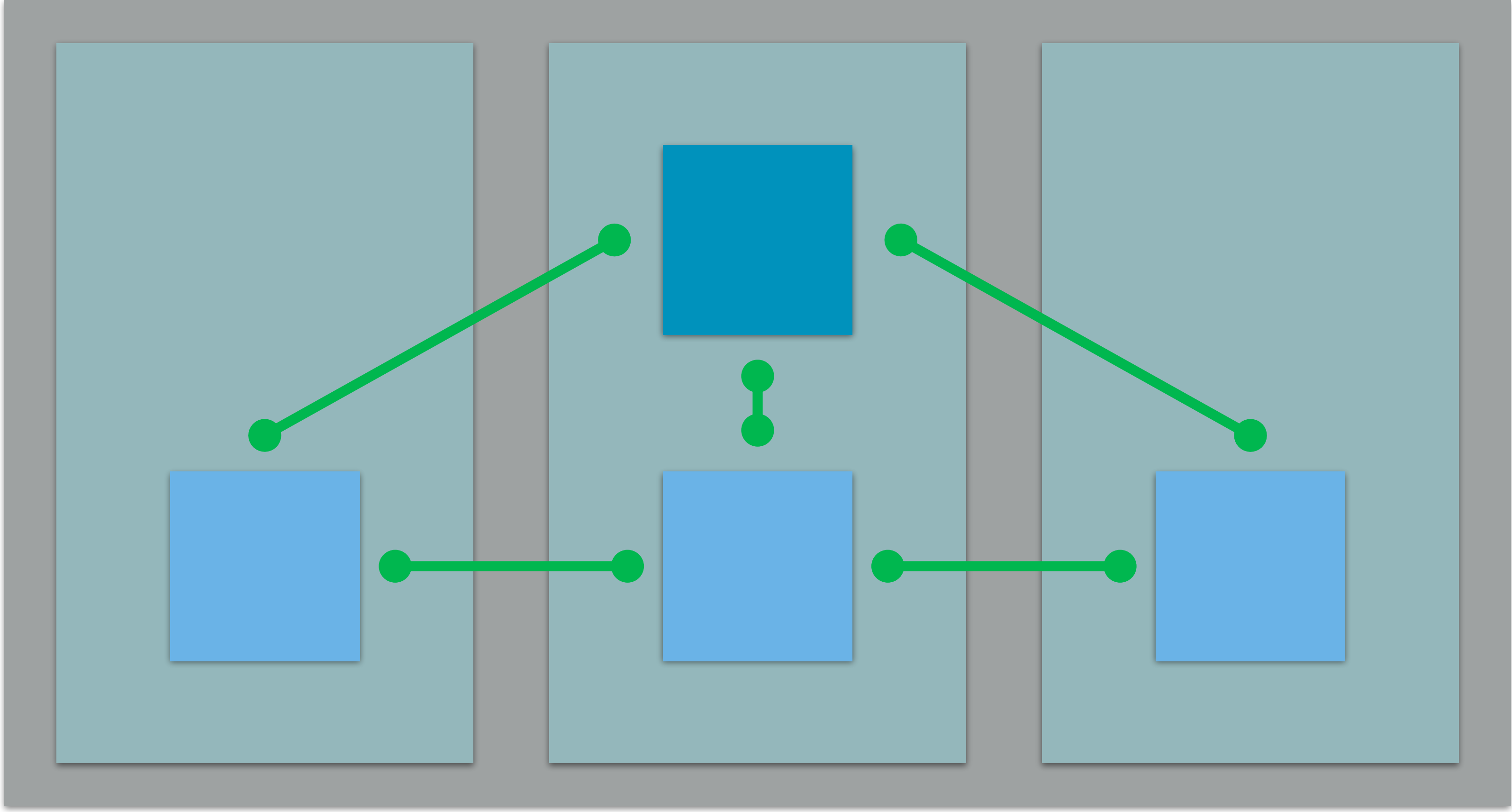
kanali



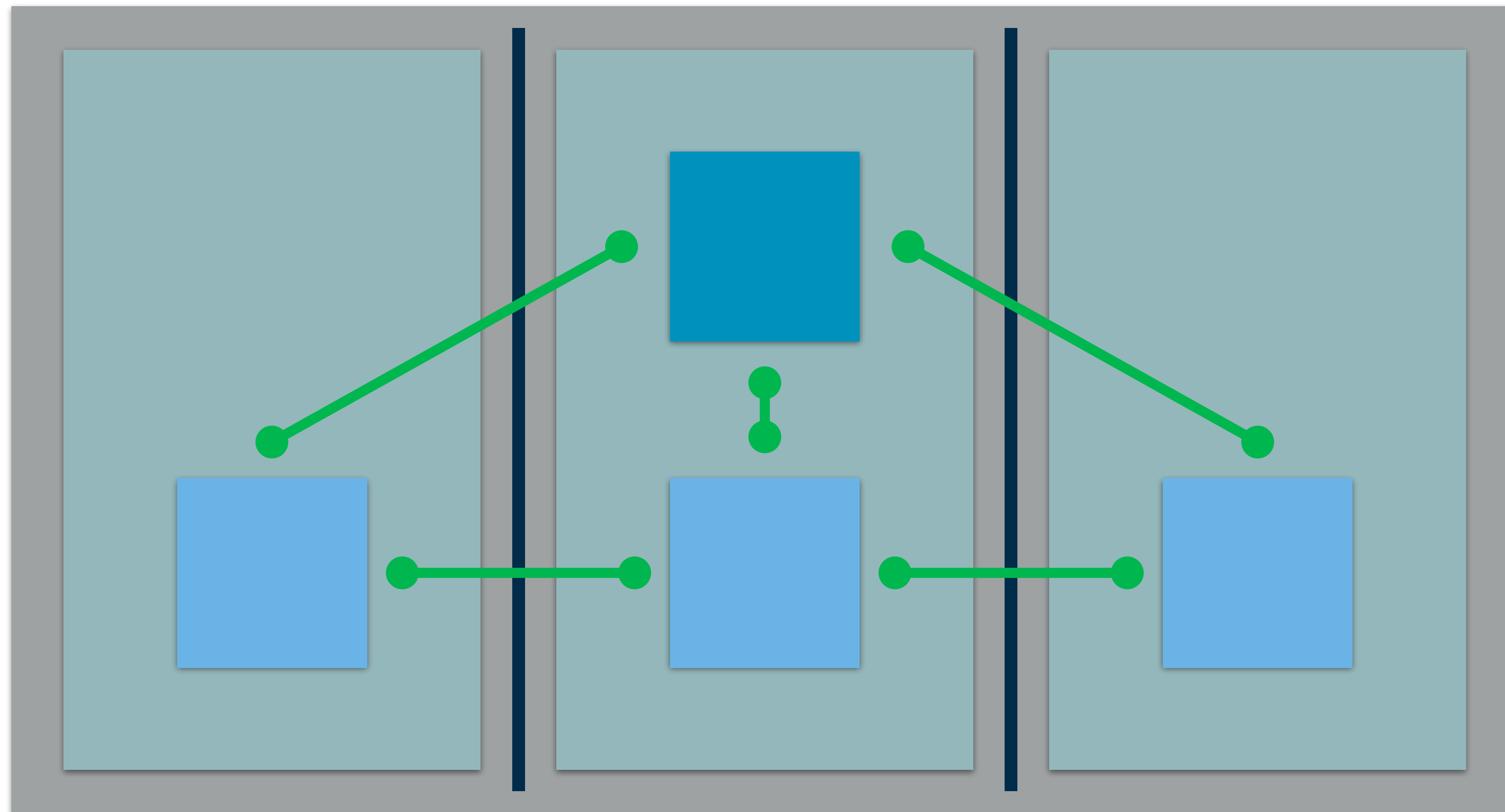
cluster ns app kanali



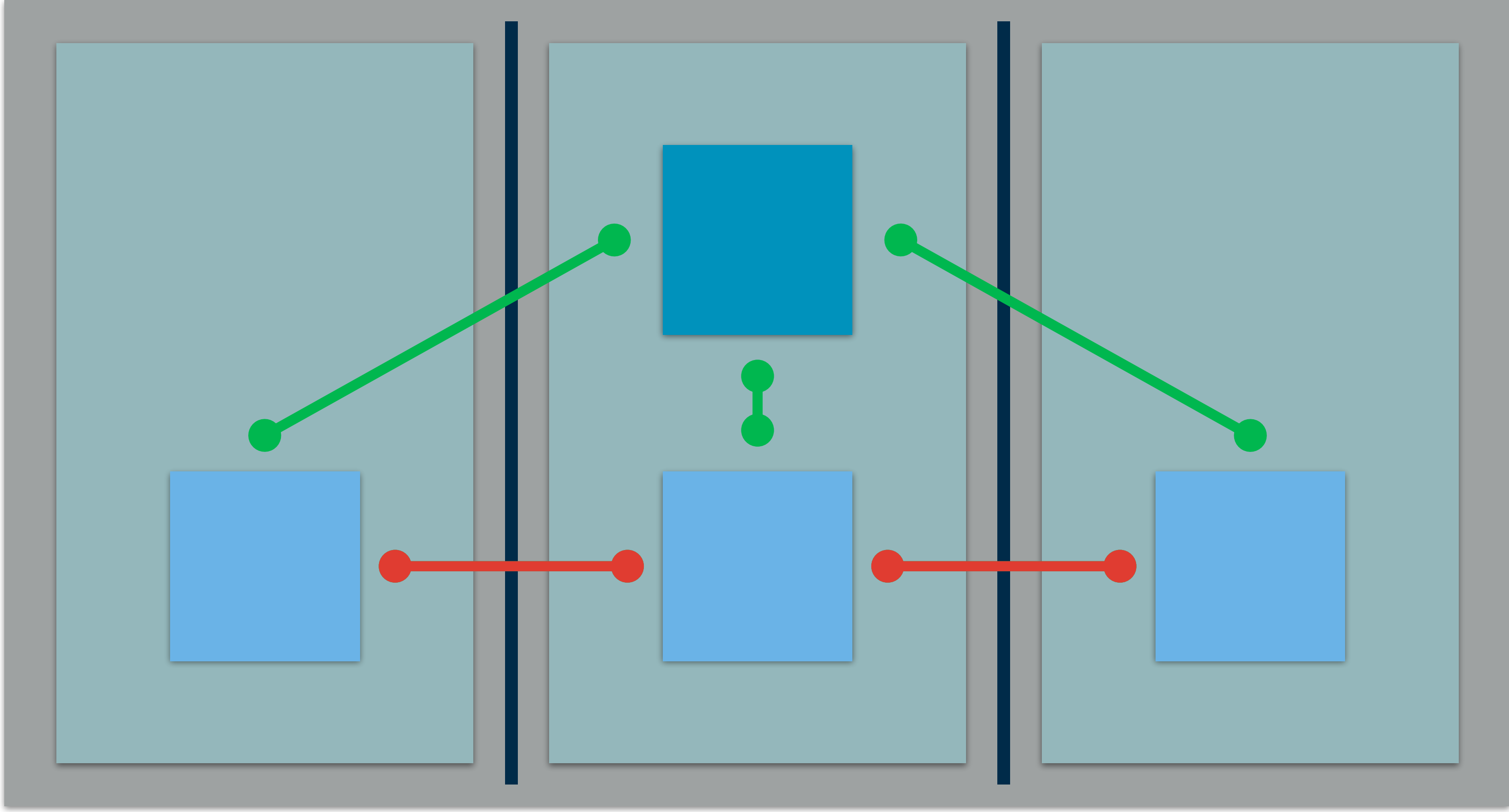
cluster ns app kanali



cluster ns app kanali



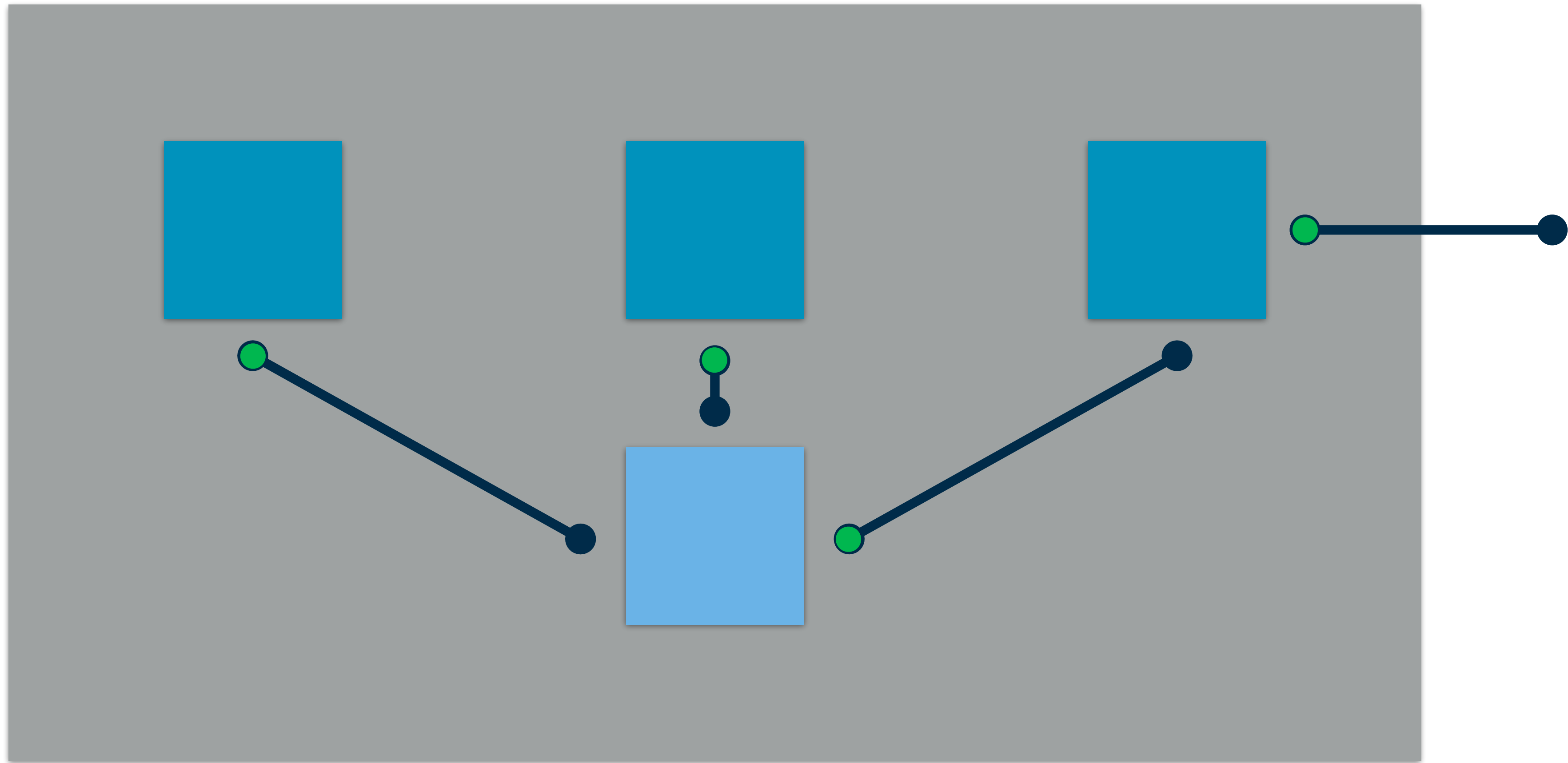
cluster ns app kanali



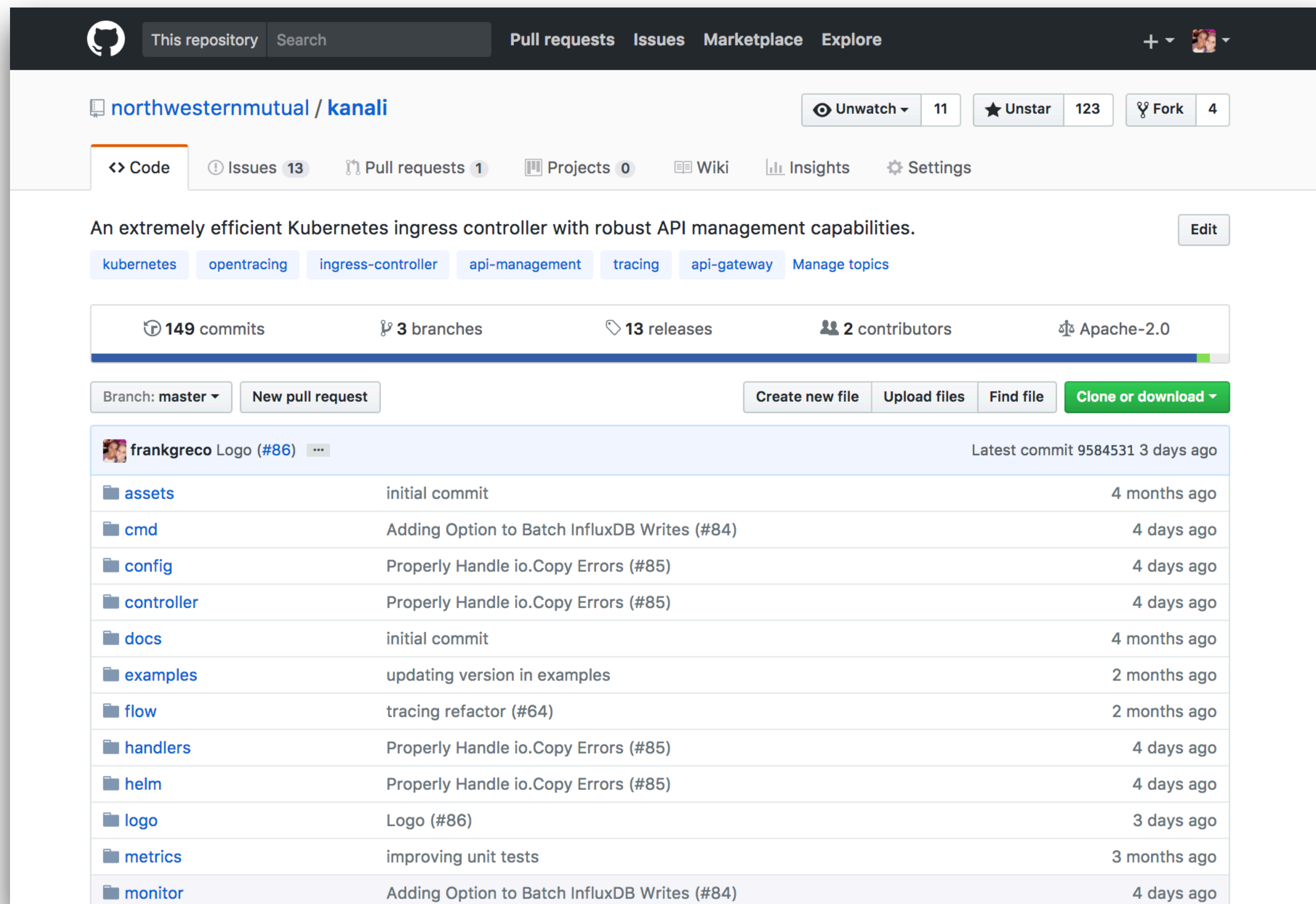
cluster

etcd

kanali



github.com/northwesternmutual/kanali



The screenshot shows the GitHub repository page for `northwesternmutual/kanali`. The repository is described as "An extremely efficient Kubernetes ingress controller with robust API management capabilities." It has 149 commits, 3 branches, 13 releases, 2 contributors, and 123 stars. The repository is licensed under Apache-2.0. The commit history is shown below, with the latest commit by `frankgreco` (commit #86) 3 days ago. The commit history table is as follows:

Commit	Message	Time
assets	initial commit	4 months ago
cmd	Adding Option to Batch InfluxDB Writes (#84)	4 days ago
config	Properly Handle io.Copy Errors (#85)	4 days ago
controller	Properly Handle io.Copy Errors (#85)	4 days ago
docs	initial commit	4 months ago
examples	updating version in examples	2 months ago
flow	tracing refactor (#64)	2 months ago
handlers	Properly Handle io.Copy Errors (#85)	4 days ago
helm	Properly Handle io.Copy Errors (#85)	4 days ago
logo	Logo (#86)	3 days ago
metrics	improving unit tests	3 months ago
monitor	Adding Option to Batch InfluxDB Writes (#84)	4 days ago



/apis/apiextensions.k8s.io/v1beta1/customresourcedefinitions

group name

/apis/apiextensions.k8s.io/v1beta1/customresourcedefinitions

group name

version name

/apis/apiextensions.k8s.io/v1beta1/customresourcedefinitions



frankgreco



@fbgrecojr

group name

version name

resource name

/apis/apiextensions.k8s.io/v1beta1/customresourcedefinitions



frankgreco



@fbgrecojr

group name

version name

resource name

/apis/apiextensions.k8s.io/v1beta1/customresourcedefinitions

/apis/kanali.io/v2/apiproxies



frankgreco



@fbgrecojr

group name

version name

resource name

/apis/apiextensions.k8s.io/v1beta1/customresourcedefinitions

group name

/apis/kanali.io/v2/apiproxies



frankgreco



@fbgrecojr

group name

version name

resource name

/apis/apiextensions.k8s.io/v1beta1/customresourcedefinitions

group name

version name

/apis/kanali.io/v2/apiproxies



frankgreco



@fbgrecojr

group name

version name

resource name

/apis/apiextensions.k8s.io/v1beta1/customresourcedefinitions

group name

version name

resource name

/apis/kanali.io/v2/apiproxies



frankgreco



@fbgrecojr

```
---
kind: ApiProxy
apiVersion: kanali.io/v2alpha1
metadata:
  name: example-proxy-one
  namespace: default
spec:
  source:
    path: /api/v1/example
    virtualHost: foo.bar.com
  target:
    path: /example
  backend:
    service:
      name: example-service
      port: 443
    ssl:
      secretName: certs
  plugins:
  - name: apiKey
    version: v1.0.0
    config:
      bindingName: example-binding
```

```
---
kind: ApiProxy
apiVersion: kanali.io/v2alpha1
metadata:
  name: example-proxy-one
  namespace: default
spec:
  source:
    path: /api/v1/example
    virtualHost: foo.bar.com
  target:
    path: /example
  backend:
    endpoint: https://api.foo.com
  ssl:
    secretName: certs
  plugins:
  - name: apiKey
    version: v1.0.0
    config:
      bindingName: example-binding
```

```
---
kind: ApiProxy
apiVersion: kanali.io/v2alpha1
metadata:
  name: example-proxy-one
  namespace: default
spec:
  source:
    path: /api/v1/example
    virtualHost: foo.bar.com
  target:
    path: /example
  backend:
    mock:
      mockTargetName: example-mock
  ssl:
    secretName: certs
  plugins:
  - name: apiKey
    version: v1.0.0
    config:
      bindingName: example-binding
```

```
---
kind: ApiProxy
apiVersion: kanali.io/v2
metadata:
  name: example-proxy-one
  namespace: default
spec:
  source:
    path: /api/v1/example
    virtualHost: foo.bar.com
  target:
    path: /example
  backend:
    service:
      name: example-service
      port: 443
  ssl:
    secretName: certs
  plugins:
  - name: apiKey
    version: v1.0.0
  config:
    bindingName: example-binding
```

```
---
kind: ApiProxy
apiVersion: kanali.io/v2
metadata:
  name: example-proxy-one
  namespace: default
spec:
  source:
    path: /api/v1/example
    virtualHost: foo.bar.com
  target:
    path: /example
  backend:
    endpoint: https://api.foo.com
  ssl:
    secretName: certs
  plugins:
  - name: apiKey
    version: v1.0.0
  config:
    bindingName: example-binding
```

```
---
kind: ApiProxy
apiVersion: kanali.io/v2
metadata:
  name: example-proxy-one
  namespace: default
spec:
  source:
    path: /api/v1/example
    virtualHost: foo.bar.com
  target:
    path: /example
  backend:
    mock:
      mockTargetName: example-mock
  ssl:
    secretName: certs
  plugins:
  - name: apiKey
    version: v1.0.0
  config:
    bindingName: example-binding
```

```
---
kind: ApiKeyBinding
apiVersion: kanali.io/v2
metadata:
  name: example-binding
  namespace: default
spec:
  keys:
  - name: bobs-apikey
    rate:
      unit: seconds
      amount: 100
    defaultRule:
      global: true
    subpaths:
  - path: /balance
    rule:
      granular:
        verbs:
        - GET
```



```
---
kind: ApiKeyBinding
apiVersion: kanali.io/v2
metadata:
  name: example-binding
  namespace: default
spec:
  keys:
  - name: bobs-apikey
    rate:
      unit: seconds
      amount: 100
    defaultRule:
      global: true
    subpaths:
  - path: /balance
    rule:
      granular:
        verbs:
        - GET
```

```
---
kind: ApiKey
apiVersion: kanali.io/v2
metadata:
  name: bobs-apikey
spec:
  revisions:
  - data: rsaencrypteddata
    status: active
    lastUsed: "Mon May 4 14:32:59 CDT 2018"
  - data: rsaencrypteddata
    status: inactive
    lastUsed: "Mon Aug 1 14:32:59 CDT 2017"
```

```
---
kind: ApiKeyBinding
apiVersion: kanali.io/v2
metadata:
  name: example-binding
  namespace: default
spec:
  keys:
  - name: bobs-apikey
    rate:
      unit: seconds
      amount: 100
    defaultRule:
      global: true
    subpaths:
  - path: /balance
    rule:
      granular:
        verbs:
        - GET
```

```
---
kind: ApiKey
apiVersion: kanali.io/v2
metadata:
  name: bobs-apikey
spec:
  revisions:
  - data: rsaencrypteddata
    status: active
    lastUsed: "Mon May 4 14:32:59 CDT 2018"
  - data: rsaencrypteddata
    status: inactive
    lastUsed: "Mon Aug 1 14:32:59 CDT 2017"
```



```
—
kind: MockTarget
apiVersion: kanali.io/v2
metadata:
  name: example-mock
  namespace: default
spec:
  routes:
  - path: /health
    status: 200
    methods:
    - GET
    headers:
      Content-Type: application/json
    body: |
      {
        "msg": "all systems up and running"
      }
```

```
[tracing]
jaeger_server_url: "jaeger-server.kube-system.svc.cluster.local"
jaeger_agent_url: "jaeger-agent.kube-system.svc.cluster.local"

[analytics]
influx_addr: "influxdb.svc.cluster.local:8086"
influx_db: "kanali"

[plugins]
location: "/"

[plugins.apiKey]
decryption_key_file: "/etc/kanali/key.pem"
header_key: "apikey"

[tls]
cert_file: "/etc/kanali/rsa/cert.pem"
key_file: "/etc/kanali/rsa/key"

[server]
port: 8443
bind_address: "0.0.0.0"

[process]
log_level: "info"

[proxy]
enable_cluster_ip: true
enable_mock_responses: true
upstream_timeout: "0h0m20s"

[proxy.default_header_values]
x-canary-deployment: "stable"
```

```
package plugins

import (
    ...
)

type Plugin interface {
    OnRequest(...) error
    OnResponse(...) error
}
```

```
package plugins

import (
    ...
)

type Plugin interface {
    OnRequest(...) error
    OnResponse(...) error
}
```

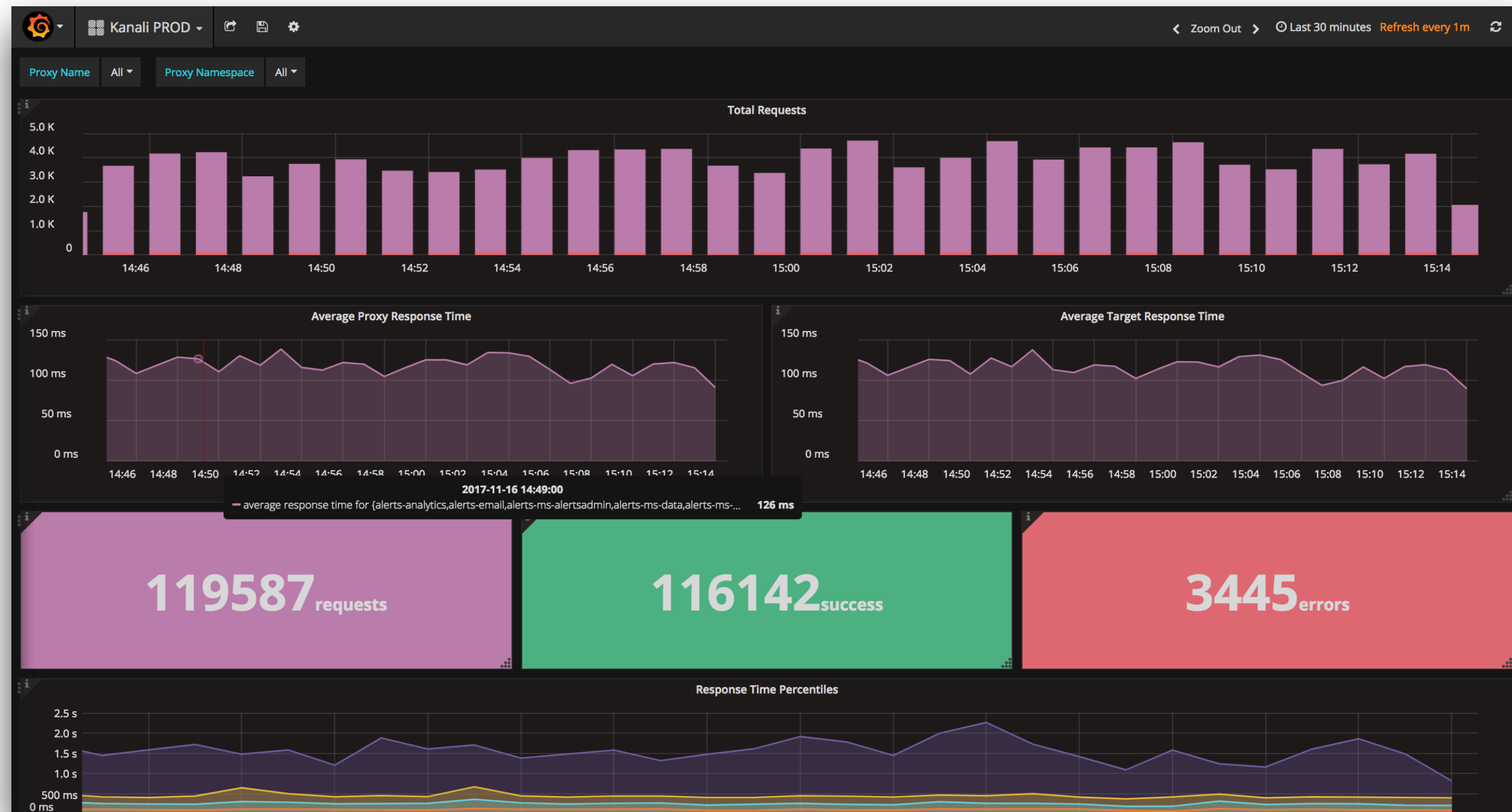




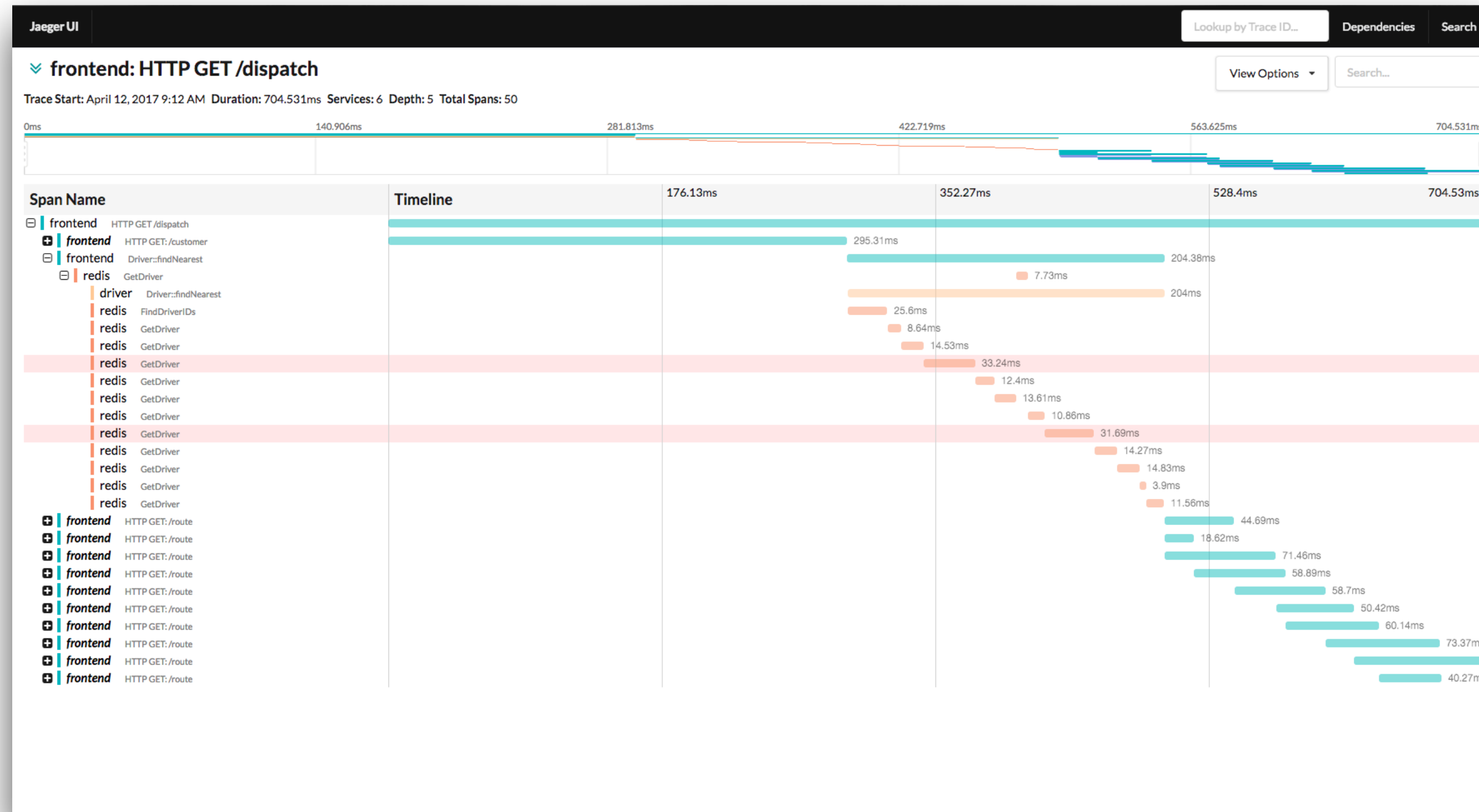
Tooling


```
~ ➤ kanalictl apikey generate --key.name=bobs-apikey --rsa.public_key_file=./public-prod.pem
Here is your api key (you will only see this once): JZuJ1ohgEuk0l02gZZ9WLLgMG5vkzDhD
Corresponding Kubernetes config:
apiVersion: kanali.io/v1
kind: ApiKey
metadata:
  creationTimestamp: null
  name: bobs-apikey
  namespace: default
spec:
  data: cef3e78e0898d08949f6cb878340322dec3f51e62dd511d8883671711fae2b5968e1382ed64f0a4a079a224d1609e175d03b5
a555032e36baa2b6134c6ab9d87736979dbd8b0b4a69e83480e6085e0a25ad527e4498f326dc25b34e1a2c3184d094c153792d0958cb2
0be8136741c2022ce0d5e8472d92e093b3b93d50513480130b53b41fd9e1891b00e10d41d0107aded3596c6341afbebb0aa8b1b7b1124
04fb95056cf446ef646b9ec9a75cdef73424d7f01c4b16facc5180db3451f76a0803dc06a9e63e9159ec76604b00843f56b0869e07598
2ed3ca59df33a166d67f1e19da53cb74f308105afaff9b947b3d5edd2a600ad042d20c97eeda7abf26875a5674d0d732a2560091cb552
090f20243a126e5251d684dbd599befb7c1a8e2d1a1cf0ffaf1c2410503192b44dbd104917d94f8d665c8c1813119ca7432d20a9cef00
fc29415d24bb9eaaa66f12ef1fb586483713fcb93ccef67acec0f358502a85b59c33a83d3ddec1d49f05a3b200542ff2a6fa2d144832
0730133f1dd9eb34fcb72eb7d27d5a8b4cbc84cd2acfe0b4d6a3537f08b0a43af8eac266938a88facbd59217676f6f308906d96bcc7c
d1abd77e625110f0128e2ccbebf81c82c1c5cd51d2356c10491123474d1a5c5127c513211dab787d4423e9c2d976955fdaa2ce6c4bc4
ce6b4b2cad291a830ce609006bb81a0d9ae243c2fc87f3459ca
```

Grafana



Jaeger



Future Work

github.com/northwesternmutual/kanali

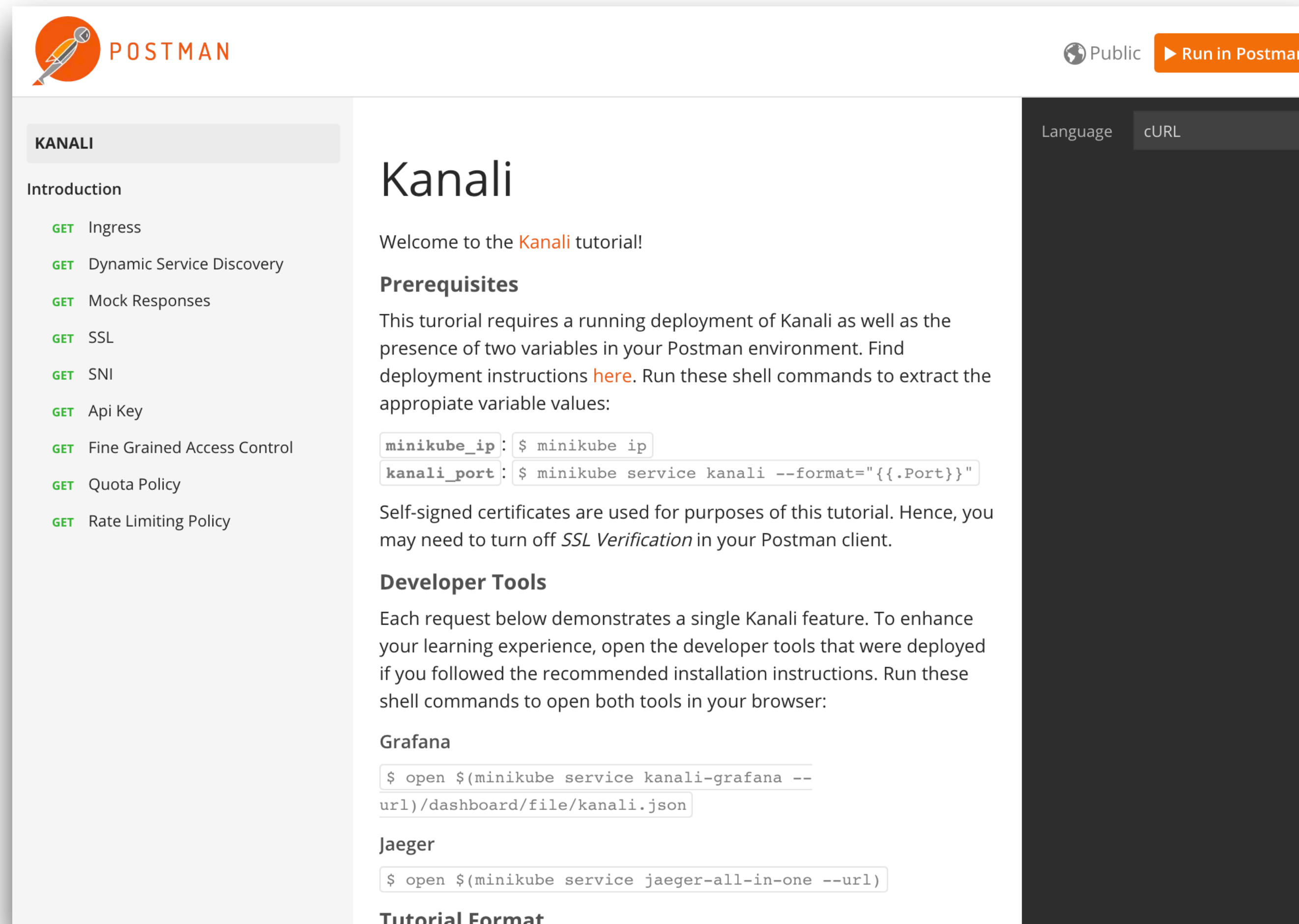
This screenshot shows the GitHub interface for the repository `northwesternmutual/kanali`. The repository has 11 unwatchers, 97 stars, and 1 fork. The 'Issues' tab is active, showing 12 open issues. A search filter `is:issue is:open` is applied. The issues list includes:

- Roadmap Clarifications** (#68) opened 13 days ago by derekperkins (2 comments)
- Non Breaking API Key Rotation** (#63) opened 22 days ago by pfermm (1 comment, enhancement label)
- Graduate Quota/Rate Limiting to Stable** (#56) opened on Sep 4 by frankgreco (8 comments, enhancement label, v2.1.0)
- Update Kubernetes Client** (#46) opened on Aug 19 by frankgreco (2 comments, priority label, v2.0.0)
- Message When Multiple Upstream Services** (#45) opened on Aug 18 by frankgreco (1 comment, enhancement label, v2.0.0)
- Some Requests in Influxdb do not have response code** (#31) opened on Aug 11 by frankgreco (bug label)
- Proxy Protocol Doesn't Work** (#24) opened on Aug 10 by frankgreco (2 comments, hold label)
- Add Contextual Logging with Zap** (#22) opened on Aug 9 by frankgreco (enhancement label, v2.0.0)
- Virtual Host Routing** (enhancement label)



Demo

tutorial.kanali.io



POSTMAN Public [Run in Postman](#)

KANALI

Introduction

- [GET](#) Ingress
- [GET](#) Dynamic Service Discovery
- [GET](#) Mock Responses
- [GET](#) SSL
- [GET](#) SNI
- [GET](#) Api Key
- [GET](#) Fine Grained Access Control
- [GET](#) Quota Policy
- [GET](#) Rate Limiting Policy

Kanali

Welcome to the [Kanali](#) tutorial!

Prerequisites

This tutorial requires a running deployment of Kanali as well as the presence of two variables in your Postman environment. Find deployment instructions [here](#). Run these shell commands to extract the appropriate variable values:

```
minikube_ip: $ minikube ip
kanali_port: $ minikube service kanali --format="{{.Port}}"
```

Self-signed certificates are used for purposes of this tutorial. Hence, you may need to turn off *SSL Verification* in your Postman client.

Developer Tools

Each request below demonstrates a single Kanali feature. To enhance your learning experience, open the developer tools that were deployed if you followed the recommended installation instructions. Run these shell commands to open both tools in your browser:

Grafana

```
$ open $(minikube service kanali-grafana --url)/dashboard/file/kanali.json
```

Jaeger

```
$ open $(minikube service jaeger-all-in-one --url)
```

Tutorial Format



? Q&A

“What about Istio?”