Prometheus-Operator e Jsonnet

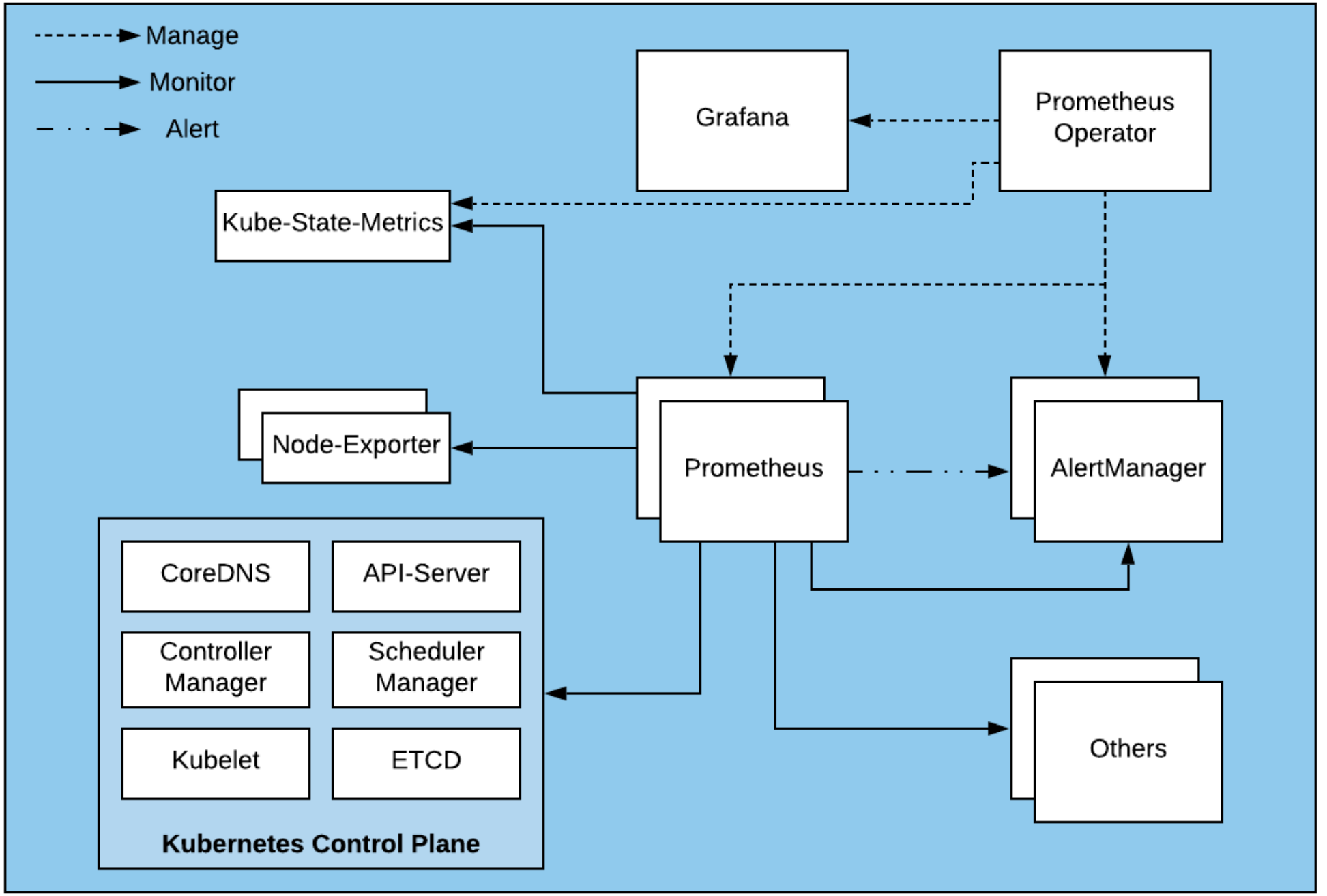
Construindo um stack de monitoramento dinâmico

Carlos Eduardo de Paula - Red Hat

@carlosedp

Maio 2019

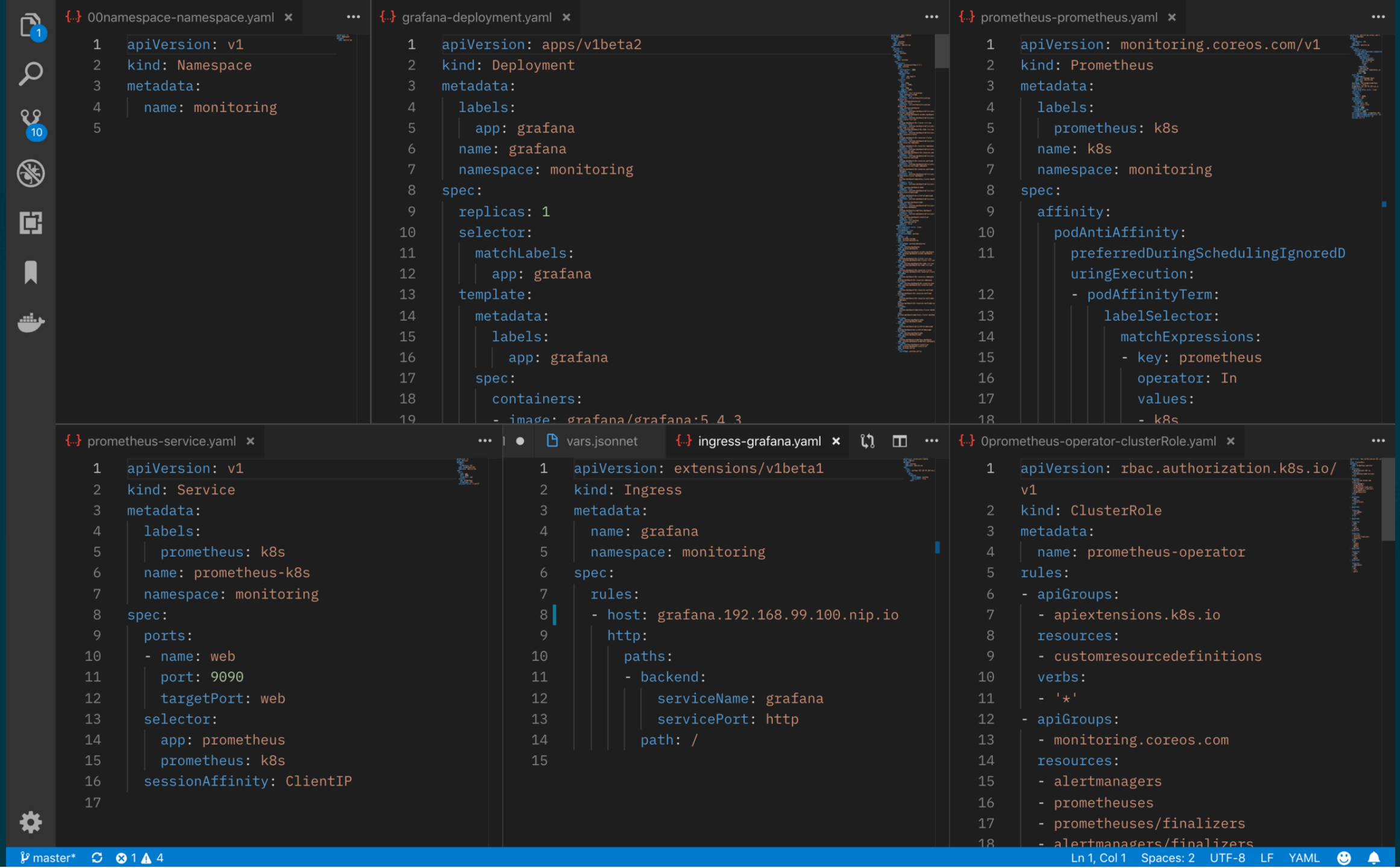
O que é um stack de monitoramento?


Manage
Monitor
Alert
Grafana
Prometheus Operator
Kube-State-Metrics
Node-Exporter
Prometheus
AlertManager
CoreDNS
API-Server
Controller Manager
Scheduler Manager
Kubelet
ETCD
Kubernetes Control Plane
Others

Stack de Monitoramento

Stack baseado em Prometheus

- Prometheus
- Grafana
- Kube-state-metrics
- Alert-manager
- Node-exporter
- Control Plane (Kubernetes)
  - Kubelet
  - ETCD
  - API-server
  - Controller-manager
  - Scheduler-manager
  - CoreDNS



00namespace-namespace.yaml

```yaml
apiVersion: v1
kind: Namespace
metadata:
  name: monitoring
```

grafana-deployment.yaml

```yaml
apiVersion: apps/v1beta2
kind: Deployment
metadata:
  labels:
    app: grafana
  name: grafana
  namespace: monitoring
spec:
  replicas: 1
  selector:
    matchLabels:
      app: grafana
  template:
    metadata:
      labels:
        app: grafana
    spec:
      containers:
      - image: grafana/grafana:5.4.3
```

prometheus-prometheus.yaml

```yaml
apiVersion: monitoring.coreos.com/v1
kind: Prometheus
metadata:
  labels:
    prometheus: k8s
  name: k8s
  namespace: monitoring
spec:
  affinity:
    podAntiAffinity:
      preferredDuringSchedulingIgnoredDuringExecution:
      - podAffinityTerm:
          labelSelector:
            matchExpressions:
            - key: prometheus
              operator: In
              values:
              - k8s
```

prometheus-service.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  labels:
    prometheus: k8s
  name: prometheus-k8s
  namespace: monitoring
spec:
  ports:
  - name: web
    port: 9090
    targetPort: web
  selector:
    app: prometheus
    prometheus: k8s
  sessionAffinity: ClientIP
```

vars.jsonnet

ingress-grafana.yaml

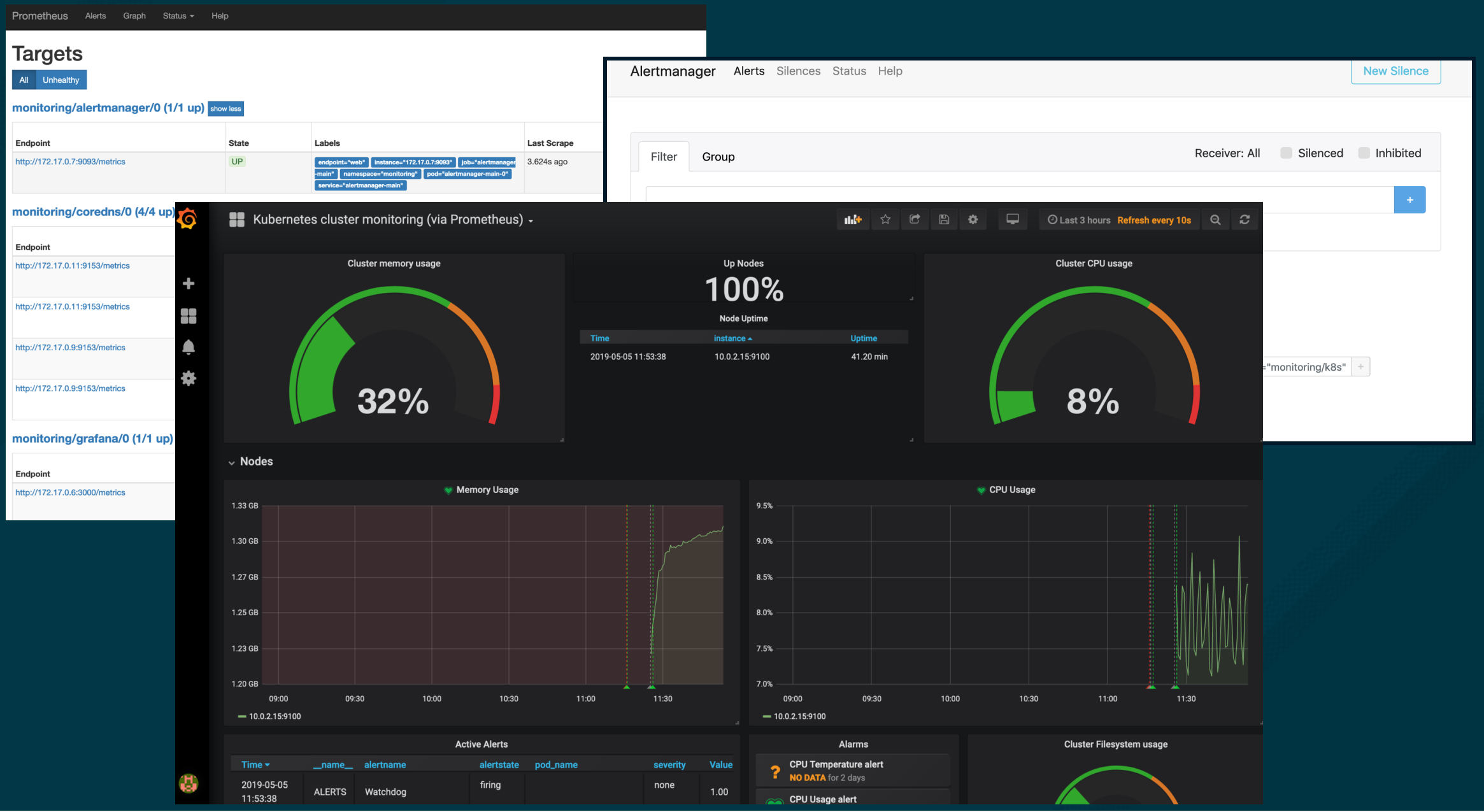
```yaml
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: grafana
  namespace: monitoring
spec:
  rules:
  - host: grafana.192.168.99.100.nip.io
    http:
      paths:
      - backend:
          serviceName: grafana
          servicePort: http
        path: /
```

0prometheus-operator-clusterRole.yaml

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: prometheus-operator
rules:
- apiGroups:
  - apiextensions.k8s.io
  resources:
  - customresourcedefinitions
  verbs:
  - '*'
- apiGroups:
  - monitoring.coreos.com
  resources:
  - alertmanagers
  - prometheuses
  - prometheuses/finalizers
  - alertmanagers/finalizers
```

master* 1 4 Ln 1, Col 1 Spaces: 2 UTF-8 LF YAML

Saúde do seu cluster



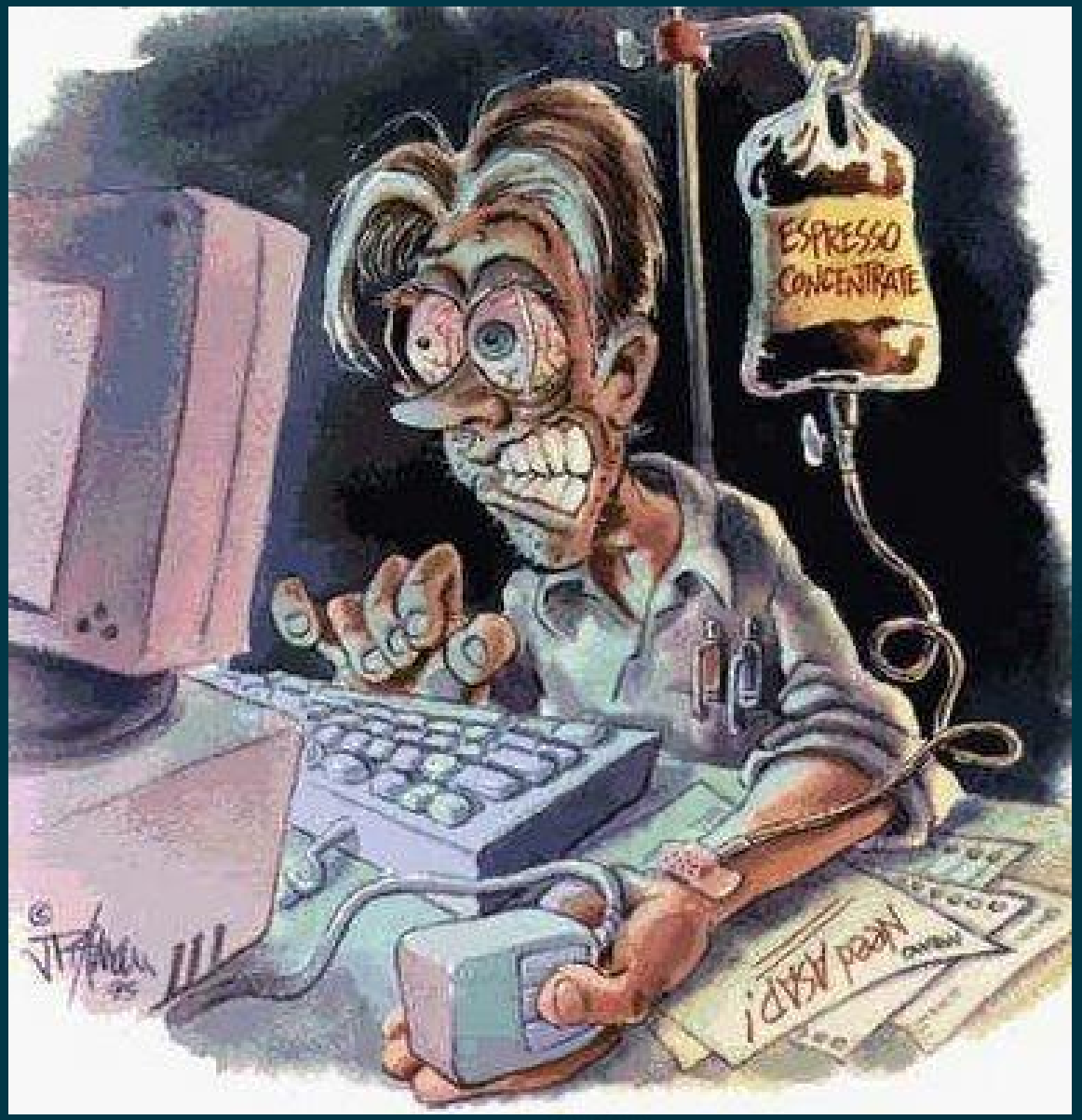
Saúde do seu desenvolvedor

Prometheus Alerts Graph Status Help

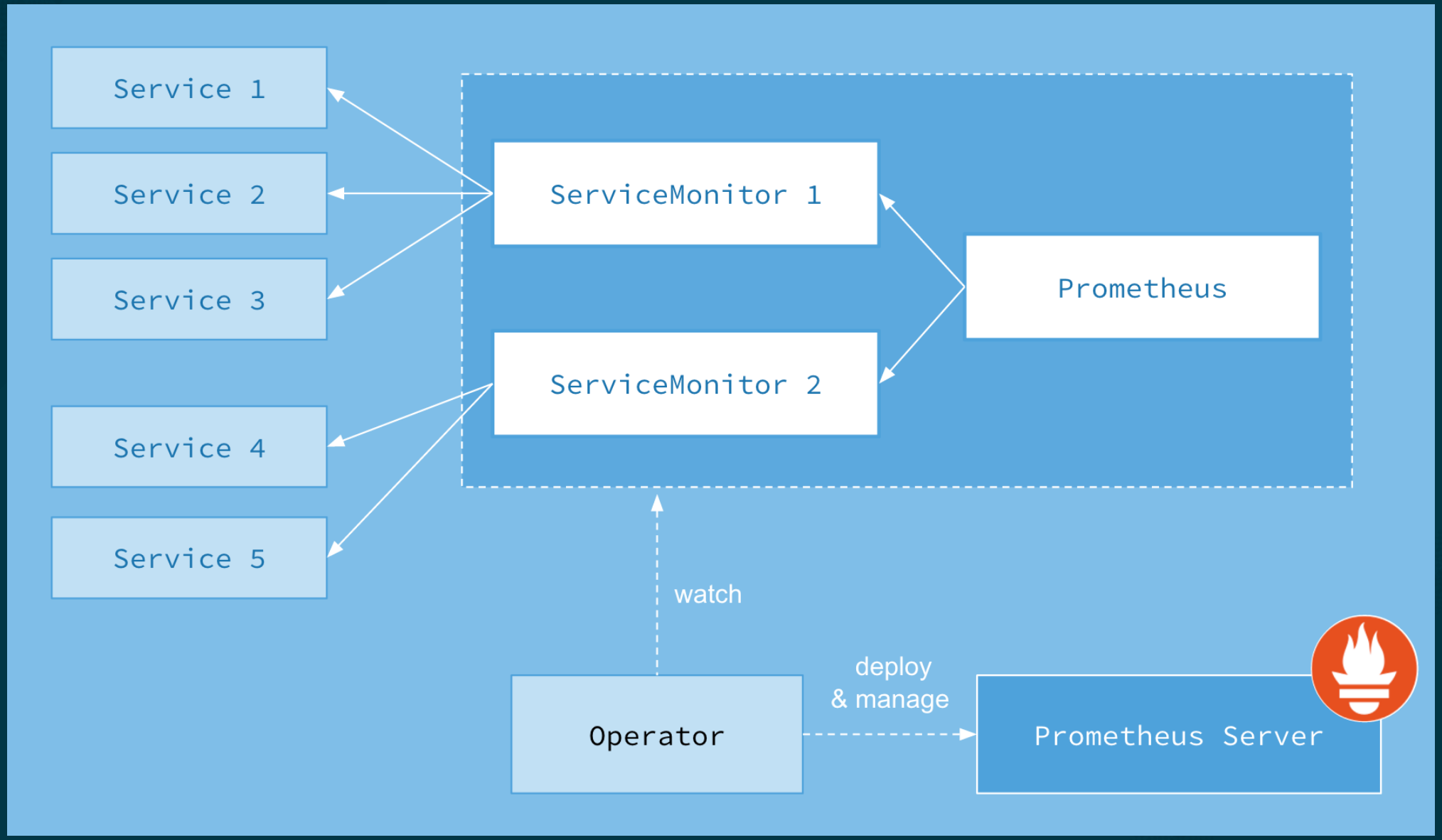
Configuration

Copy to clipboard

```
global:
  scrape_interval: 30s
  scrape_timeout: 10s
  evaluation_interval: 30s
  external_labels:
    prometheus: monitoring/k8s
    prometheus_replica: prometheus-k8s-0
alerting:
  alert_relabel_configs:
  - separator: ;
    regex: prometheus_replica
    replacement: $1
    action: labeldrop
  alertmanagers:
  - kubernetes_sd_configs:
    - role: endpoints
      namespaces:
        names:
        - monitoring
    scheme: http
    path_prefix: /
    timeout: 10s
    relabel_configs:
    - source_labels: [__meta_kubernetes_service_name]
      separator: ;
      regex: alertmanager-main
      replacement: $1
      action: keep
    - source_labels: [__meta_kubernetes_endpoint_port_name]
      separator: ;
      regex: web
      replacement: $1
      action: keep
rule_files:
- /etc/prometheus/rules/prometheus-k8s-rulefiles-0/*.yaml
scrape_configs:
- job_name: monitoring/alertmanager/0
  scrape_interval: 30s
  scrape_timeout: 10s
  metrics_path: /metrics
  scheme: http
```

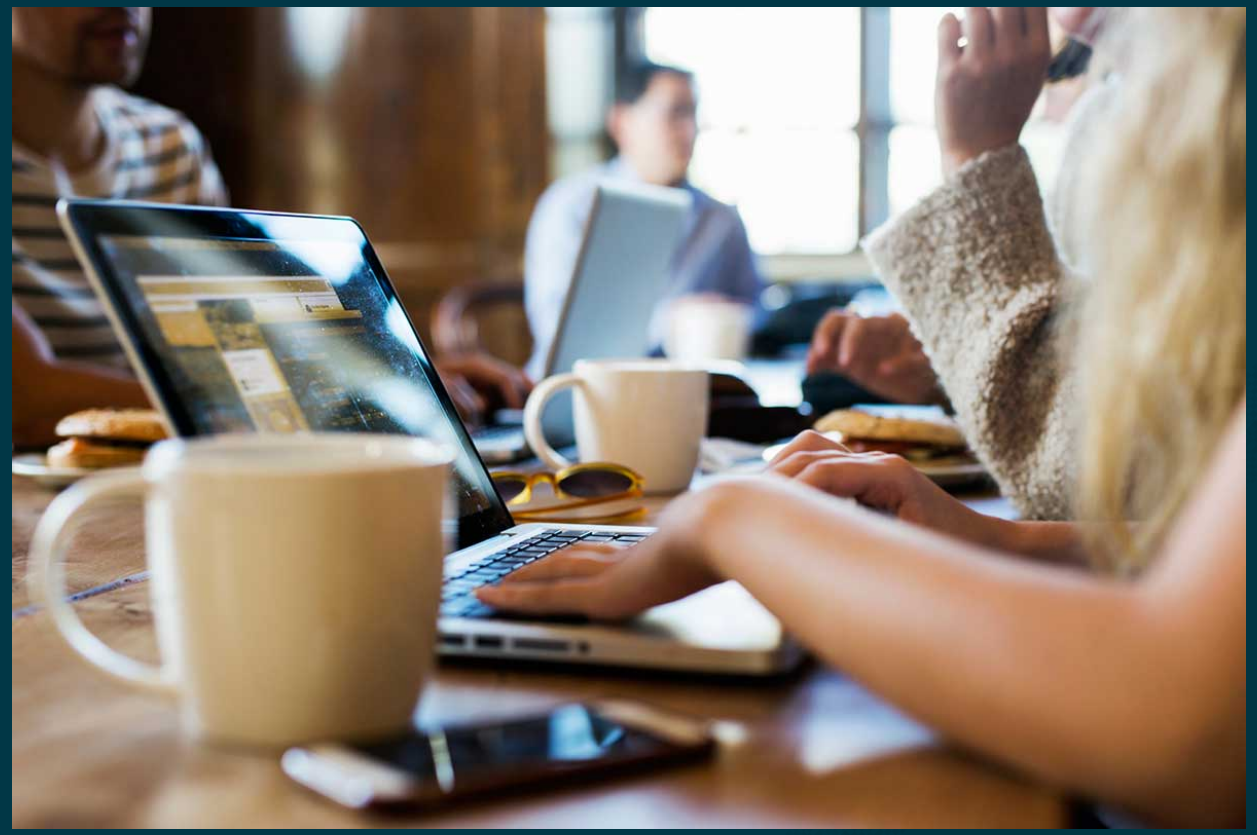


Prometheus-Operator



Expectativa

- Fácil configuração
- Simplicidade no deploy
- Dinamico
- Extensível
- Reutilizável



Simples Deploy

0prometheus-operator-deployment.yaml

```yaml
apiVersion: apps/v1beta2
kind: Deployment
metadata:
  labels:
    k8s-app: prometheus-operator
  name: prometheus-operator
  namespace: monitoring
spec:
  replicas: 1
  selector:
    matchLabels:
      k8s-app: prometheus-operator
  template:
    metadata:
      labels:
        k8s-app: prometheus-operator
    spec:
      containers:
      - args:
        - --kubelet-service=kube-system/kubelet
        - --logtostderr=true
        - --config-reloader-image=carlosedp/
        configmap-reload:v0.2.2
        - --prometheus-config-reloader=carlosedp/
        prometheus-config-reloader:v0.28.0
        image: carlosedp/
        prometheus-operator:v0.28.0
        name: prometheus-operator
        ports:
        - containerPort: 8080
```

prometheus-prometheus.yaml

```yaml
apiVersion: monitoring.coreos.com/v1
kind: Prometheus
metadata:
  labels:
    prometheus: k8s
  name: k8s
  namespace: monitoring
spec:
  affinity:
    podAntiAffinity:
      preferredDuringSchedulingIgnoredDuringExecuti
      on:
      - podAffinityTerm:
          labelSelector:
            matchExpressions:
            - key: prometheus
              operator: In
              values:
              - k8s
          namespaces:
          - monitoring
          topologyKey: kubernetes.io/hostname
        weight: 100
  alerting:
    alertmanagers:
    - name: alertmanager-main
      namespace: monitoring
      port: web
  baseImage: carlosedp/prometheus
  externalUrl: http://
```

Simples Deploy

```
NAMESPACE     NAME                                    READY   STATUS    RESTARTS   AGE   IP
kube-system   coredns-fb8b8dccf-gz5rg                 1/1     Running   4          47h   172.
kube-system   coredns-fb8b8dccf-vnv64                 1/1     Running   4          47h   172.
kube-system   default-http-backend-6864bbb7db-tssxt   1/1     Running   1          47h   172.
kube-system   etcd-minikube                           1/1     Running   0          60m   10.0
kube-system   kube-addon-manager-minikube             1/1     Running   1          47h   10.0
kube-system   kube-apiserver-minikube                 1/1     Running   0          60m   10.0
kube-system   kube-controller-manager-minikube        1/1     Running   2          47h   10.0
kube-system   kube-proxy-rnq6m                        1/1     Running   0          59m   10.0
kube-system   kube-scheduler-minikube                 1/1     Running   2          47h   10.0
kube-system   kubernetes-dashboard-79dd6bfc48-kczlc   1/1     Running   4          47h   172.
kube-system   metrics-server-77fddcc57b-4c4h4         1/1     Running   3          47h   172.
kube-system   nginx-ingress-controller-586cdc477c-2v24l   1/1     Running   4          47h   172.
kube-system   storage-provisioner                     1/1     Running   4          47h   10.0
monitoring    alertmanager-main-0                     2/2     Running   3          47h   172.
monitoring    grafana-65468686b6-w9dnz                1/1     Running   1          47h   172.
monitoring    kube-state-metrics-7d65cbfff7-fhtfb     4/4     Running   6          47h   172.
monitoring    node-exporter-29255                     2/2     Running   0          56m   10.0
monitoring    prometheus-adapter-54d6b86d5c-xznt9     1/1     Running   3          47h   172.
monitoring    prometheus-k8s-0                        3/3     Running   1          50m   172.
monitoring    prometheus-operator-7669474f47-7g795    1/1     Running   3          47h   172.
monitoring    smtp-server-677cd65fff-w7llp            1/1     Running   1          47h   172.
```

Dinâmico

```yaml
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  labels:
    k8s-app: node-exporter
  name: node-exporter
  namespace: monitoring
spec:
  endpoints:
  - bearerTokenFile: /var/run/secrets/kubernetes.io/serviceaccount/token
    interval: 30s
    port: https
    scheme: https
    tlsConfig:
      insecureSkipVerify: true
  jobLabel: k8s-app
  selector:
    matchLabels:
      k8s-app: node-exporter
```

Ainda mais agilidade?

Jsonnet

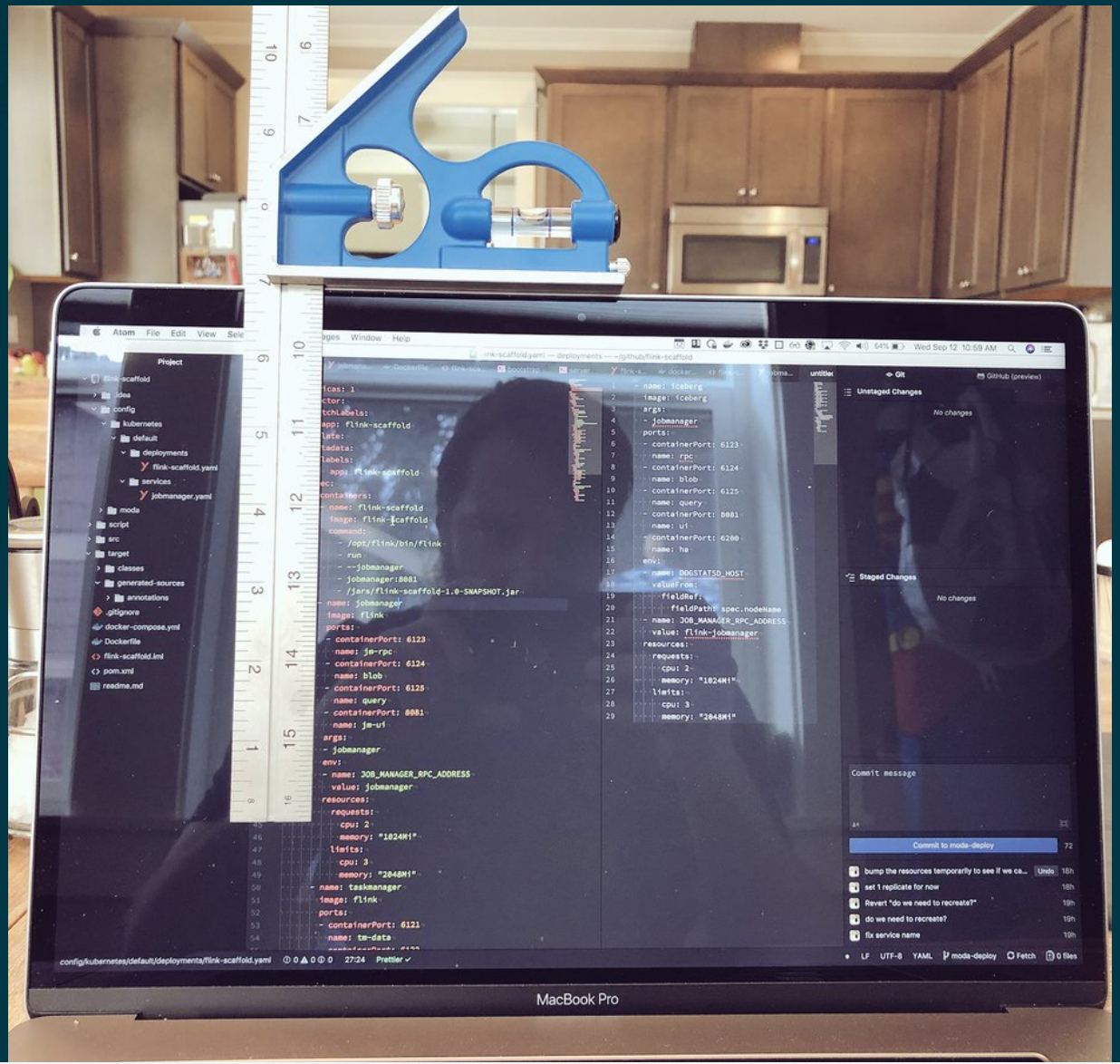
A data templating language for app and tool developers

- Generate config data
- Side-effect free
- Organize, simplify, unify
- Manage sprawling config

https://jsonnet.org/

```
local Person(name='Alice') = {
  name: name,
  welcome: 'Hello ' + name + '!',
};
{
  person1: Person(),
  person2: Person('Bob'),
}

$ jsonnet file.jsonnet

{
   "person1": {
      "name": "Alice",
      "welcome": "Hello Alice!"
   },
   "person2": {
      "name": "Bob",
      "welcome": "Hello Bob!"
   }
}
```

Geração dinâmica de YAML




```
local Person(name='Alice') = {
  name: name,
  welcome: 'Hello ' + name + '!',
};
{
  person1: Person(),
  person2: Person('Bob'),
}

$ jsonnet file.jsonnet | gojsontoyaml
```

```
person1:
  name: Alice
  welcome: Hello Alice!
person2:
  name: Bob
  welcome: Hello Bob!
```

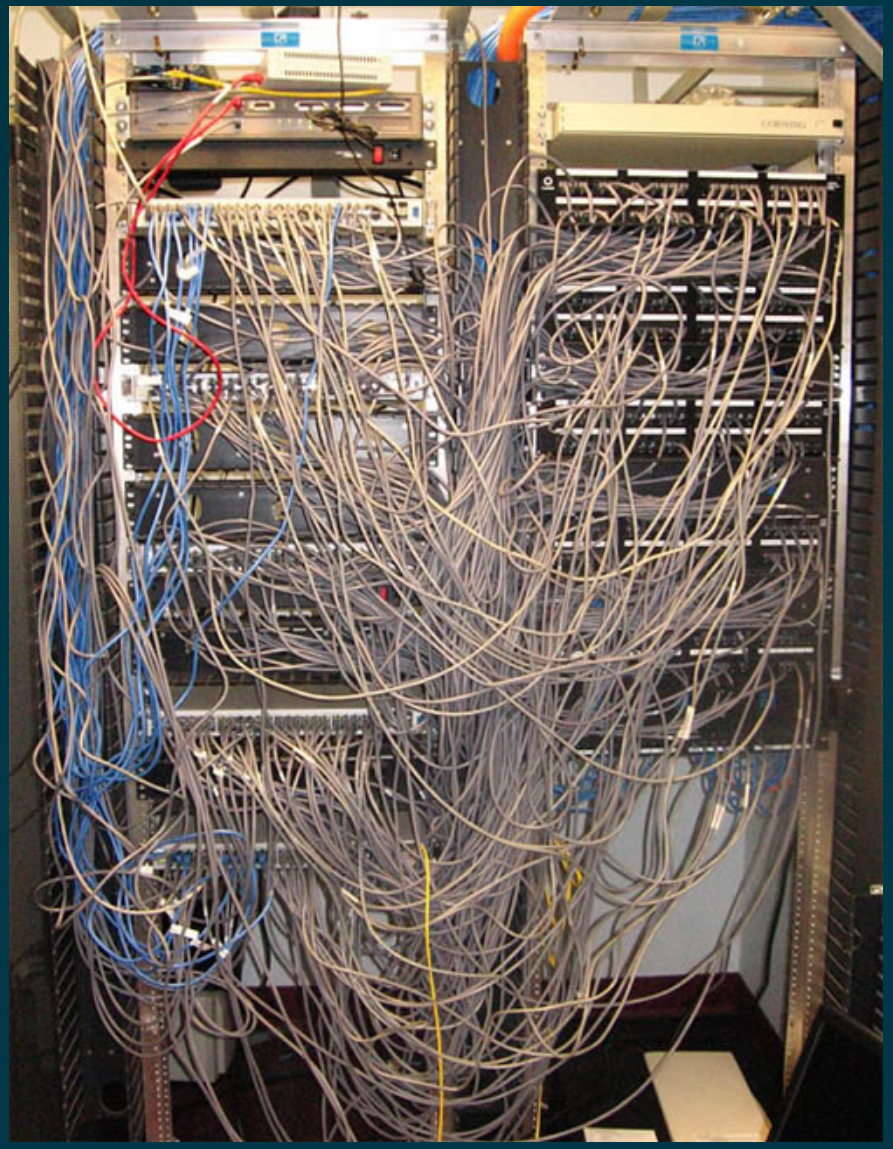
Uso de mixins - Ksonnet

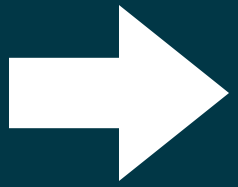
```
local k = import 'ksonnet.beta.3/k.libsonnet';

local kp = {
  _config+:: {
    namespace: 'default',
    appname: "nginx",
    image: "nginx",
    image_version: "1.7.9",
    replicas: 2,
    targetPort: 80,
  },
    genericDeployment:: {
      deployment:
        local deployment = k.apps.v1beta2.deployment;
        local container = k.apps.v1beta2.deployment.mixin.spec.template.spec.containersType;
        local containerPort = container.portsType;

        local podLabels = { 'k8s-app': $._config.appname };
        local nginx =
          container.new($._config.appname, $._config.image + ':' + $._config.image_version) +
          container.withPorts(containerPort.newNamed($._config.appname, $._config.targetPort));

        deployment.new($._config.appname, $._config.replicas, [nginx], podLabels) +
        deployment.mixin.metadata.withNamespace($._config.namespace) +
        deployment.mixin.metadata.withLabels(podLabels),
    }
};
{ ['deployment-' + name]: kp.genericDeployment[name] for name in std.objectFields(kp.genericDeployment) }
```

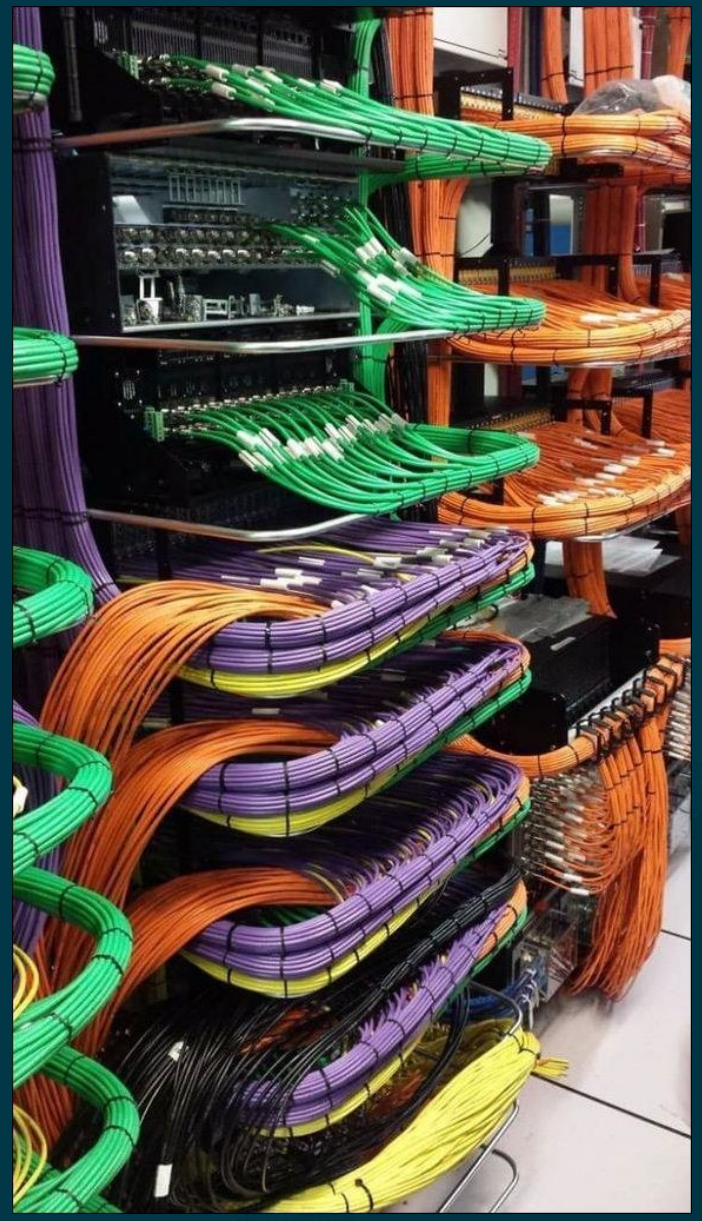
https://github.com/ksonnet/ksonnet-lib

http://g.bryan.dev.hepti.center/

Organização do código







Mixins - Kube-prometheus

vars.jsonnet

```
{
  // Enable or disable additional modules
  installModules: {
    'arm-exporter': false,
    metallb: false,
    traefik: false,
    'ups-exporter': false,
    'elastic-exporter': false,
  },

  // Setting these to false, defaults to emptyDirs
  enablePersistence: {
    prometheus: false,
    grafana: false,
  },

  // Domain suffix for the ingresses
```

elasticsearch_exporter.jsonnet

```
local k = import 'ksonnet/ksonnet.beta.3/k.libsonnet';

{
  _config+:: {
    namespace: 'monitoring',
    replicas: 1,

    versions+:: {
      elasticExporter: '1.0.2',
    },

    imageRepos+:: {
      elasticExporter: 'justwatch/elasticsearch_exporter',
    },

    prometheus+:: {
      namespaces+:: ['logging'],
```

Loading...

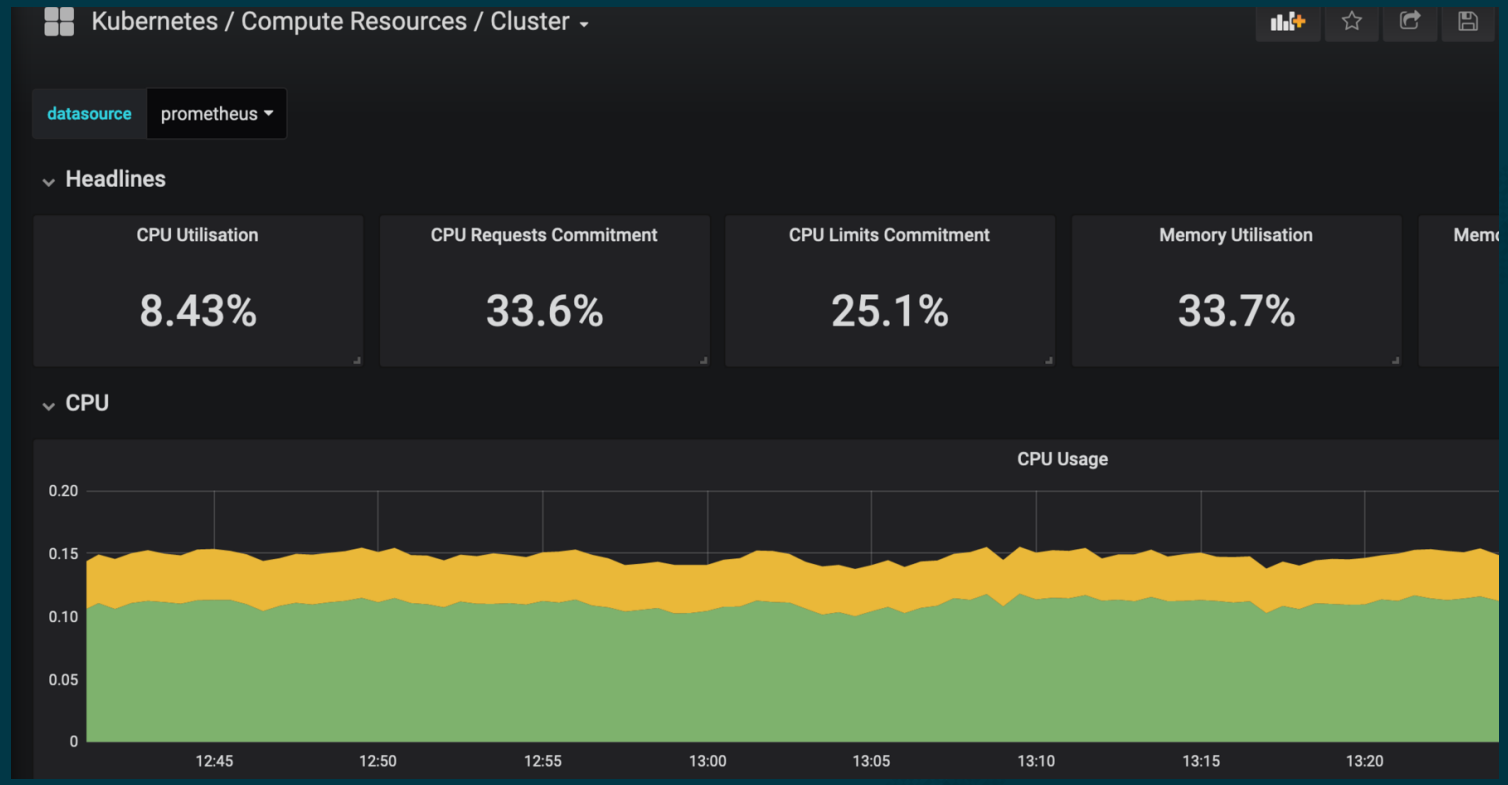
main.jsonnet

```
local k = import 'ksonnet/ksonnet.beta.3/k.libsonnet';
local vars = import 'vars.jsonnet';

local kp = (import 'kube-prometheus/kube-prometheus.libsonnet')
           + (import 'kube-prometheus/kube-prometheus-anti-affinity.libsonnet')
           + (import 'kube-prometheus/kube-prometheus-kops-coredns.libsonnet')
           + (import 'kube-prometheus/kube-prometheus-kubeadm.libsonnet')
           // Use http Kubelet targets. Comment to revert to https
           + (import 'kube-prometheus/kube-prometheus-insecure-kubelet.libsonnet')
           + (import 'base_operator_stack.jsonnet')
           + (import 'smtp_server.jsonnet')
           // Additional Modules
           + (if vars.installModules['arm-exporter'] then (import 'arm_exporter.jsonnet') else {})
           + (if vars.installModules.metallb then (import 'metallb.jsonnet') else {})
           + (if vars.installModules.traefik then (import 'traefik.jsonnet') else {})
           + (if vars.installModules['ups-exporter'] then (import 'ups_exporter.jsonnet') else {})
           + (if vars.installModules['elastic-exporter'] then (import 'elasticsearch_exporter.jsonnet') else {})
```

https://github.com/coreos/kube-prometheus

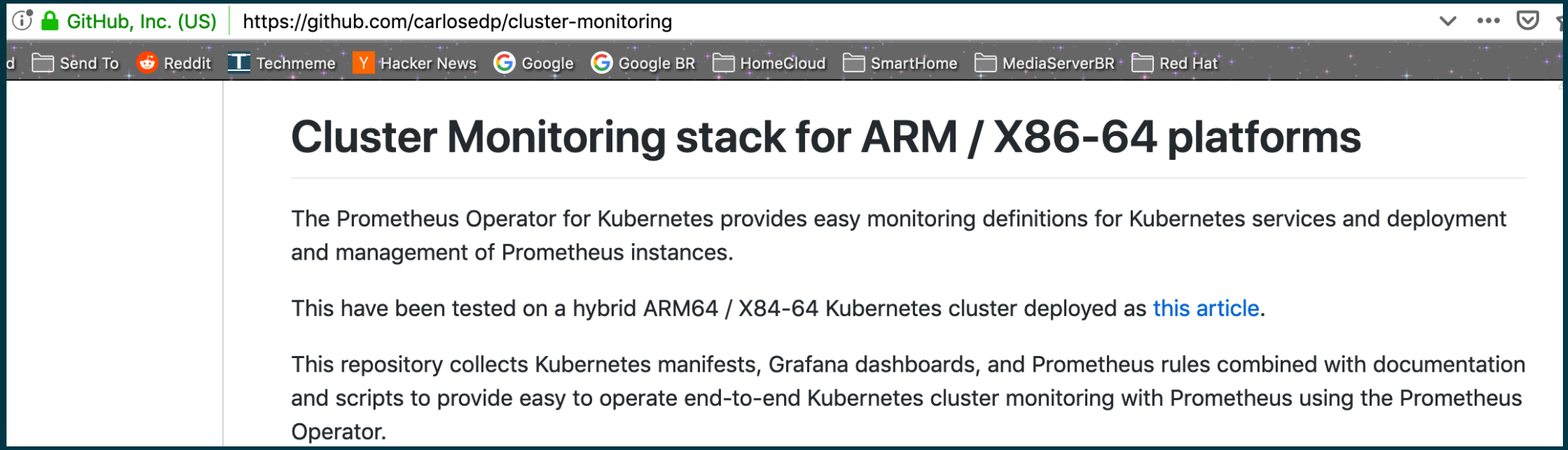
Mixins - Kube-prometheus

- Prometheus Operator
- Prometheus em HA (opcional)
- AlertManager em HA (opcional)
- Node-exporter
- Kube-state-metrics
- Alertas pré-configurados
- Dashboards pré-definidos



https://github.com/coreos/kube-prometheus

Extensível - Cluster Monitoring



- Prometheus, Grafana, KSM, node-exporter, etc
- Exporters opcionais (ElasticSearch, Traefik, MetalLB)
- Persistência em disco configurável
- Configuração de dominio ingress
- Imagens multi-arquitetura (Intel/ARM/ARM64)

https://github.com/carlosedp/cluster-monitoring

Cluster Monitoring

Vantagens

- Reuso de bibliotecas
- Tamanho do codebase
- Efêmero

```
➜ wc -l *.jsonnet
  140 arm_exporter.jsonnet
  227 base_operator_stack.jsonnet
  140 ceph_exporter.jsonnet
  188 elasticsearch_exporter.jsonnet
   37 image_sources_versions.jsonnet
   41 main.jsonnet
   58 metallb.jsonnet
   63 smtp_server.jsonnet
   46 traefik.jsonnet
   80 ups_exporter.jsonnet
   23 vars.jsonnet
 1043 total
```

```
➜ wc -l manifests/*
    4 manifests/00namespace-namespace.yaml
 2411 manifests/0prometheus-operator-0alertmanagerCustomResourceDefi
 3199 manifests/0prometheus-operator-0prometheusCustomResourceDefini
  343 manifests/0prometheus-operator-0prometheusruleCustomResourceDe
  291 manifests/0prometheus-operator-0servicemonitorCustomResourceDe
   68 manifests/0prometheus-operator-clusterRole.yaml
   12 manifests/0prometheus-operator-clusterRoleBinding.yaml
   44 manifests/0prometheus-operator-deployment.yaml
   15 manifests/0prometheus-operator-service.yaml
    5 manifests/0prometheus-operator-serviceAccount.yaml
   14 manifests/0prometheus-operator-serviceMonitor.yaml
   32 manifests/alertmanager-alertmanager.yaml
    8 manifests/alertmanager-secret.yaml
```

...

```
   37 manifests/prometheus-serviceMonitorApiserver.
   19 manifests/prometheus-serviceMonitorCoreDNS.ya
   23 manifests/prometheus-serviceMonitorKubeContro
   18 manifests/prometheus-serviceMonitorKubeSchedu
   26 manifests/prometheus-serviceMonitorKubelet.ya
   38 manifests/smtp-server-deployment.yaml
   14 manifests/smtp-server-service.yaml
28682 total
```

https://github.com/carlosedp/cluster-monitoring

Demo

```
$ minikube start \
    --network-plugin=cni \
    --enable-default-cni \
    --container-runtime=cri-o \
    --bootstrapper=kubeadm \
    --memory=4096 \
    --cpus=4 \
    --cache-images=true

$ minikube addons enable ingress
$ git clone https://github.com/carlosedp/cluster-
monitoring

$ cd cluster-monitoring
```

Hack hack hack... (minikube ip, change the ingress address var)

```
$ make vendor && make deploy
```

Obrigado!

Dúvidas?

Carlos Eduardo de Paula

twitter.com/carlosedp

github.com/carlosedp

medium.com/@carlosedp

http(s)?:\/\/.*\.(com|net|org|io)\/(@)?carlosedp