KubeCon | CloudNativeCon

Europe 2020

Virtual

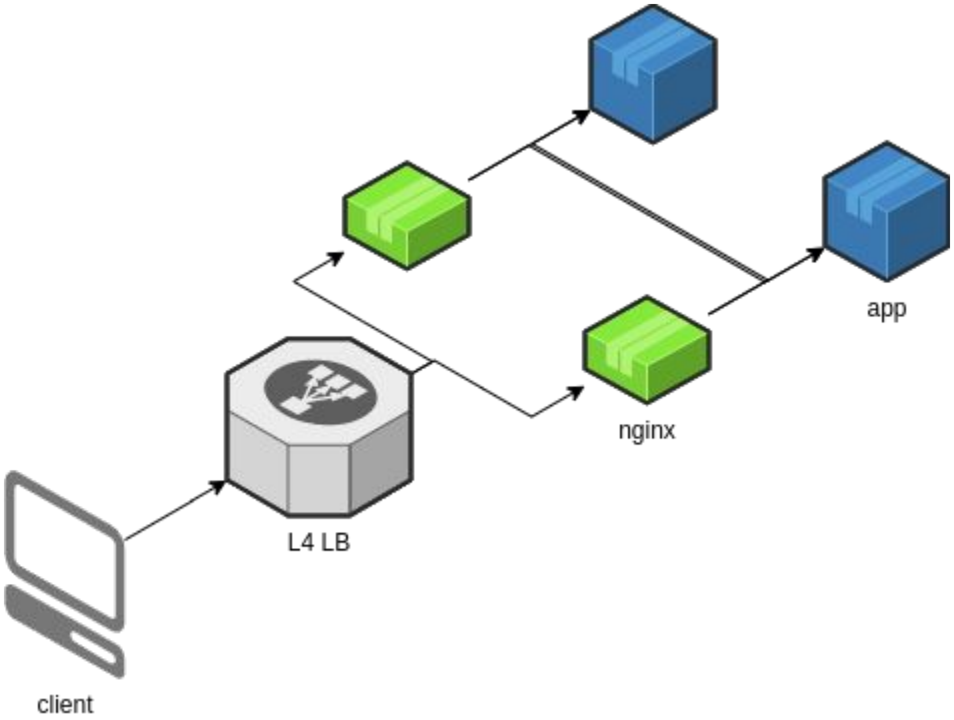
Ingress on the Rails

Alastair Firth, Camunda

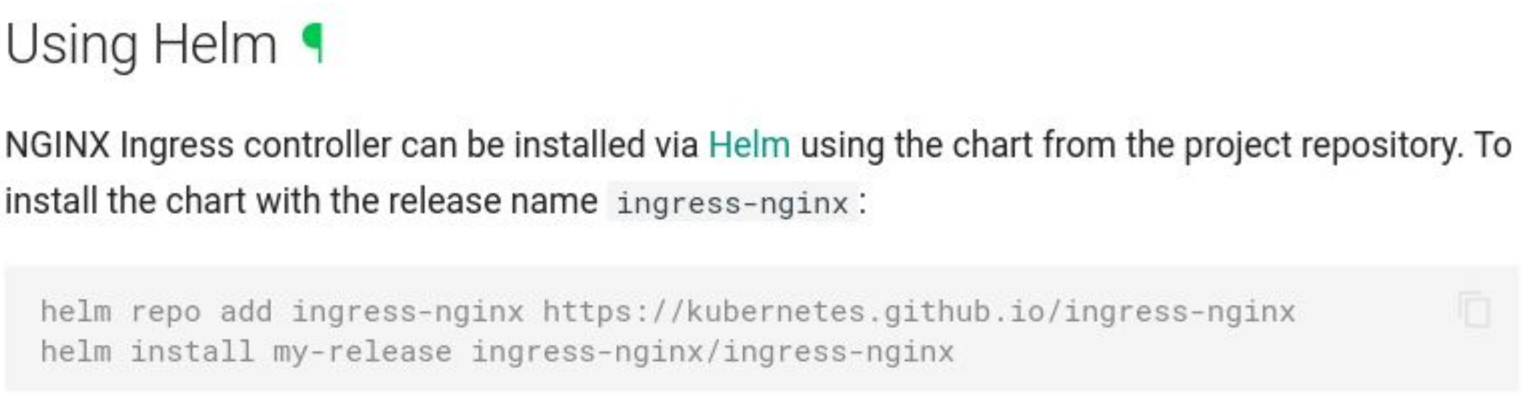
@afirth

slides: github.com/afirth/kceu2020

What you want



https://kubernetes.github.io/ingress-nginx/deploy/#using-helm



Using Helm ¶

NGINX Ingress controller can be installed via Helm using the chart from the project repository. To install the chart with the release name `ingress-nginx`:

```
helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
helm install my-release ingress-nginx/ingress-nginx
```

https://kubernetes.github.i

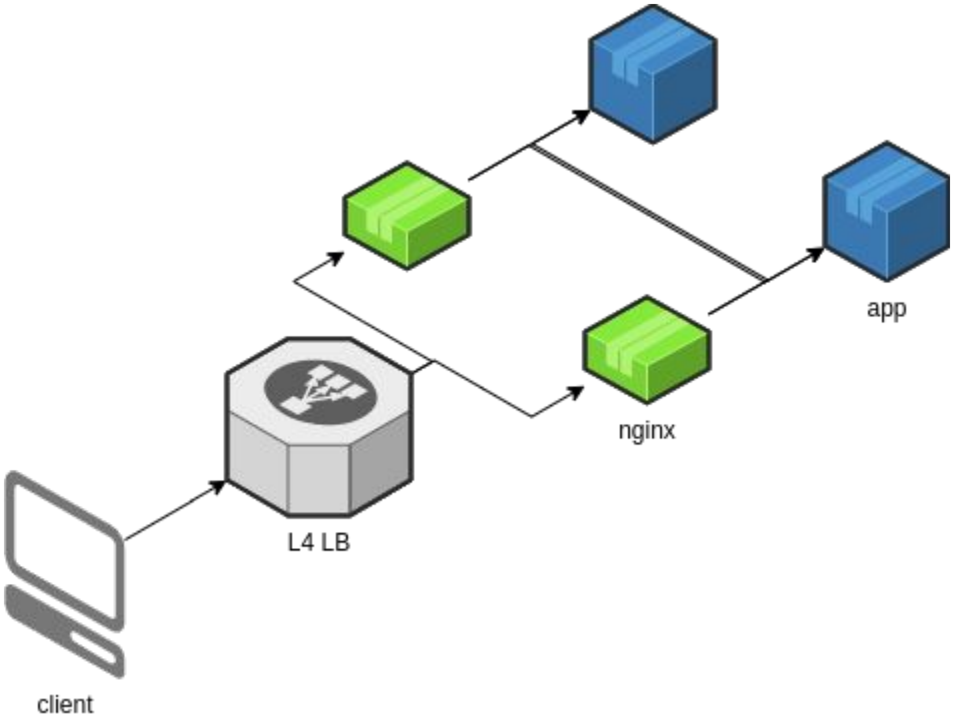
Using Helm ¶

NGINX Ingress control e project repository. To install the chart with th

```
helm repo add ingr                                          ss-nginx
helm install my-re
```

It's easy

What you want



- DNS records
- TLS certs
- Monitoring
- AuthN/AuthZ
- Multiple K8s
- Multiple apps
- Multiple stages
- Multiple nginxs

Overview

- DNS records
  - external-dns
- TLS certs
  - cert-manager
- Monitoring
  - prometheus-operator
- AuthN/AuthZ
  - oauth2-proxy/oauth2-proxy
- Reverse proxy
  - kubernetes/ingress-nginx

Painful bits

- Deployment + secrets
- Following upstream changes
- Discovering non-default options

Painful bits

- Deployment + secrets
- Following upstream changes

```
sops -d *.sops.yaml && (helm template | kustomize build | kubectl apply -f -)
```

Painful bits

- Deployment + secrets
- Following upstrear

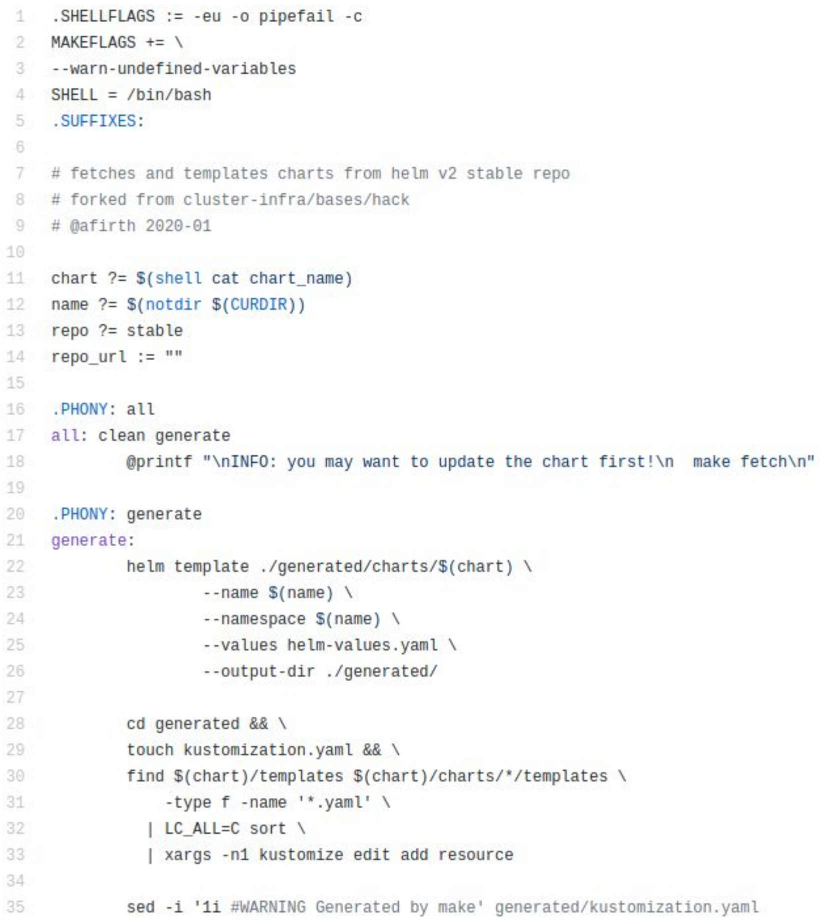
sops -d *.sops.yaml && bectl apply -f -)

It's easy

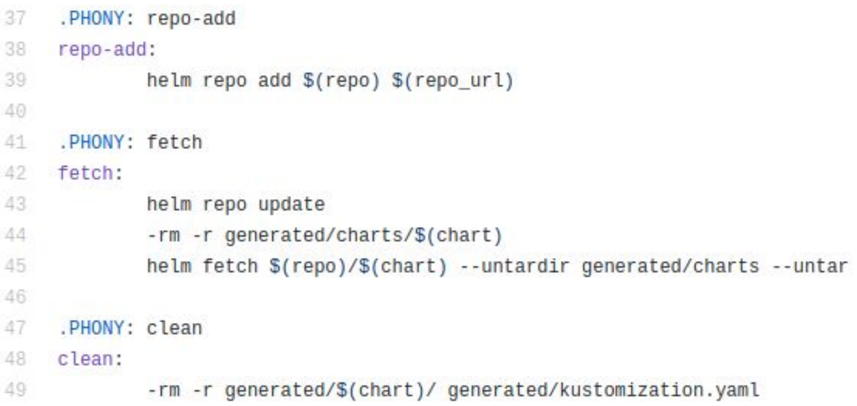
Deployment

- Deployment
  - Helm Chart (global)
    - `helm fetch`
    - `helm template`
    - `kustomize add *`
  - Kustomize Overlays (per cluster)
    - `kustomize add patches and secrets`
  - Generate and commit "plain" manifests (per cluster)
    - `kustomize build`
  - Deploy
    - `kubectl apply`

Deployment - chart makefile



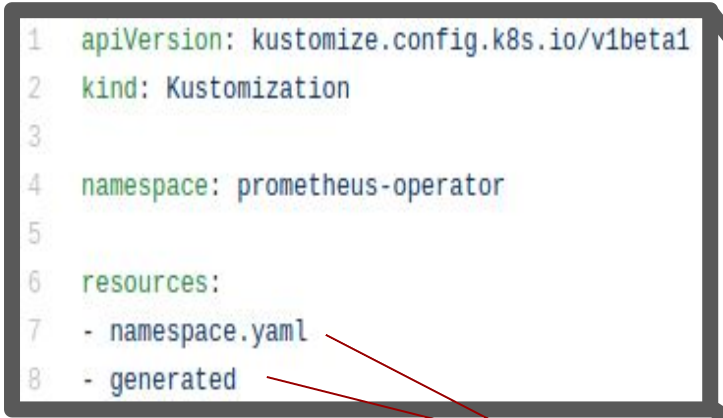
```
.SHELLFLAGS := -eu -o pipefail -c
MAKEFLAGS += \
--warn-undefined-variables
SHELL = /bin/bash
.SUFFIXES:

# fetches and templates charts from helm v2 stable repo
# forked from cluster-infra/bases/hack
# @afirth 2020-01

chart ?= $(shell cat chart_name)
name ?= $(notdir $(CURDIR))
repo ?= stable
repo_url := ""

.PHONY: all
all: clean generate
        @printf "\nINFO: you may want to update the chart first!\n  make fetch\n"

.PHONY: generate
generate:
        helm template ./generated/charts/$(chart) \
                --name $(name) \
                --namespace $(name) \
                --values helm-values.yaml \
                --output-dir ./generated/

        cd generated && \
        touch kustomization.yaml && \
        find $(chart)/templates $(chart)/charts/*/templates \
            -type f -name '*.yaml' \
          | LC_ALL=C sort \
          | xargs -n1 kustomize edit add resource

        sed -i '1i #WARNING Generated by make' generated/kustomization.yaml
```



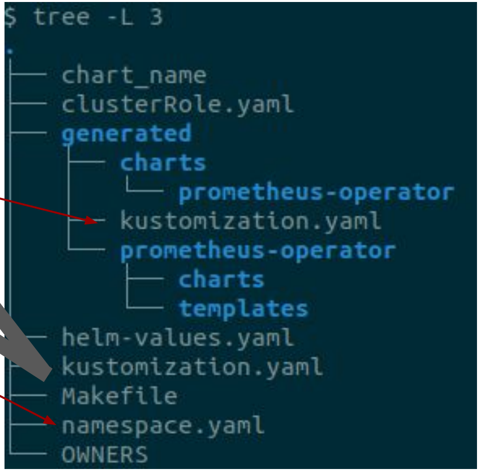
```
.PHONY: repo-add
repo-add:
        helm repo add $(repo) $(repo_url)

.PHONY: fetch
fetch:
        helm repo update
        -rm -r generated/charts/$(chart)
        helm fetch $(repo)/$(chart) --untardir generated/charts --untar

.PHONY: clean
clean:
        -rm -r generated/$(chart)/ generated/kustomization.yaml
```

github.com/afirth/kceu2020

Deployment - chart makefile



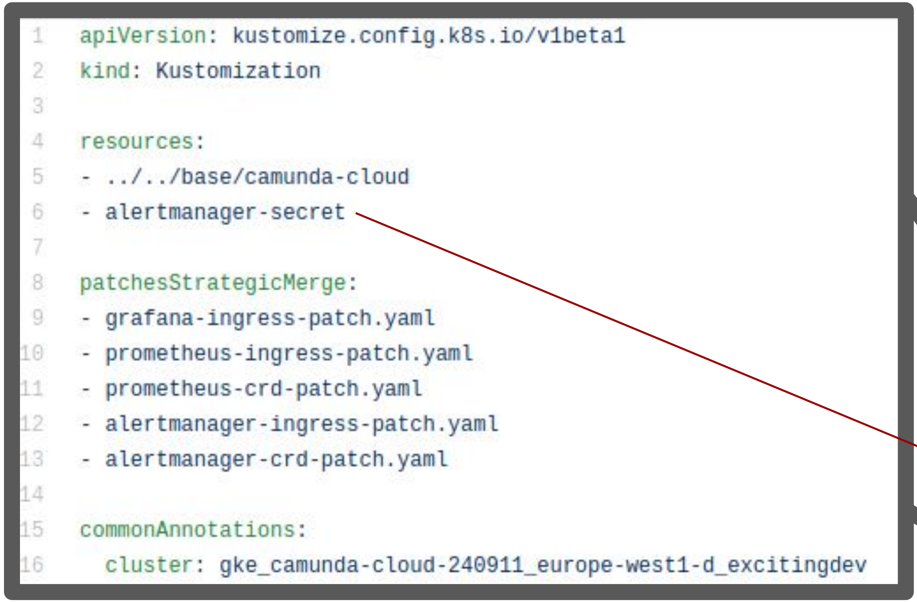
```yaml
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization

namespace: prometheus-operator

resources:
- namespace.yaml
- generated
```



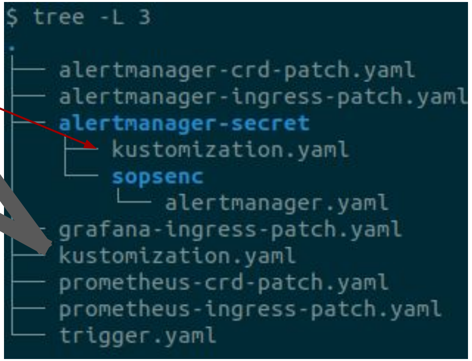
```
$ tree -L 3
.
├── chart_name
├── clusterRole.yaml
├── generated
│   ├── charts
│   │   └── prometheus-operator
│   ├── kustomization.yaml
│   └── prometheus-operator
│       ├── charts
│       └── templates
├── helm-values.yaml
├── kustomization.yaml
├── Makefile
├── namespace.yaml
└── OWNERS
```

github.com/afirth/kceu2020

Deployment - overlay



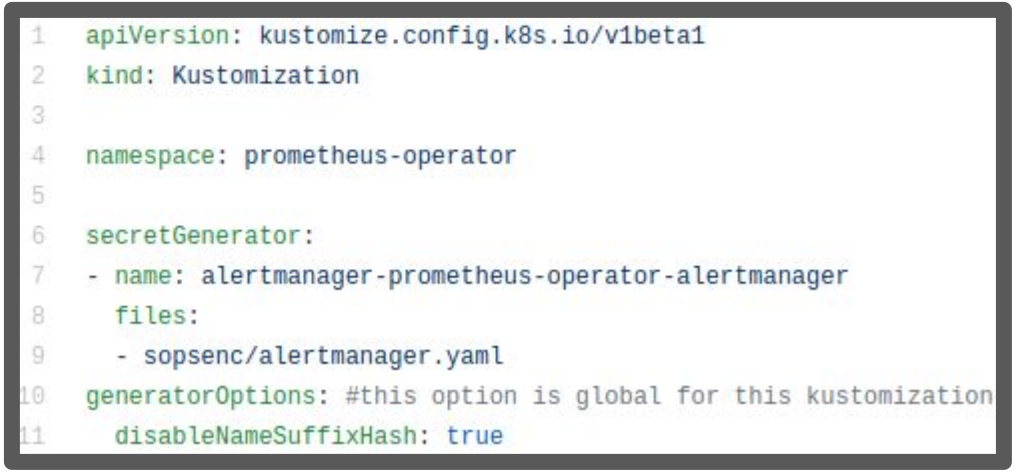
```yaml
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization

resources:
- ../../base/camunda-cloud
- alertmanager-secret

patchesStrategicMerge:
- grafana-ingress-patch.yaml
- prometheus-ingress-patch.yaml
- prometheus-crd-patch.yaml
- alertmanager-ingress-patch.yaml
- alertmanager-crd-patch.yaml

commonAnnotations:
  cluster: gke_camunda-cloud-240911_europe-west1-d_excitingdev
```



```
$ tree -L 3
.
├── alertmanager-crd-patch.yaml
├── alertmanager-ingress-patch.yaml
├── alertmanager-secret
│   ├── kustomization.yaml
│   └── sopsenc
│       └── alertmanager.yaml
├── grafana-ingress-patch.yaml
├── kustomization.yaml
├── prometheus-crd-patch.yaml
├── prometheus-ingress-patch.yaml
└── trigger.yaml
```

github.com/afirth/kceu2020

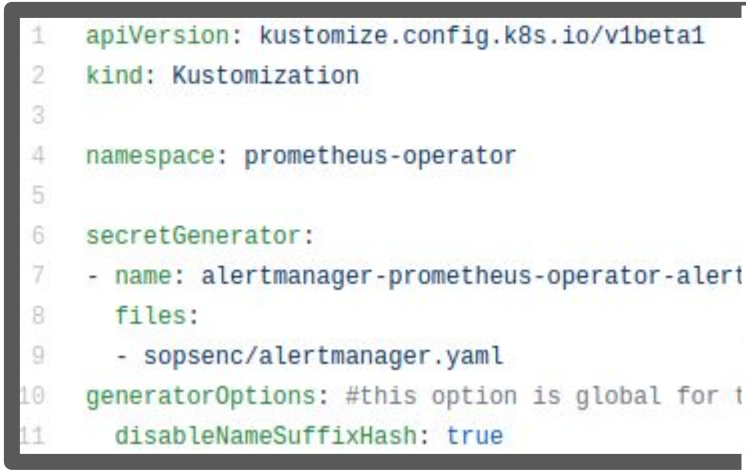
Deployment - secrets



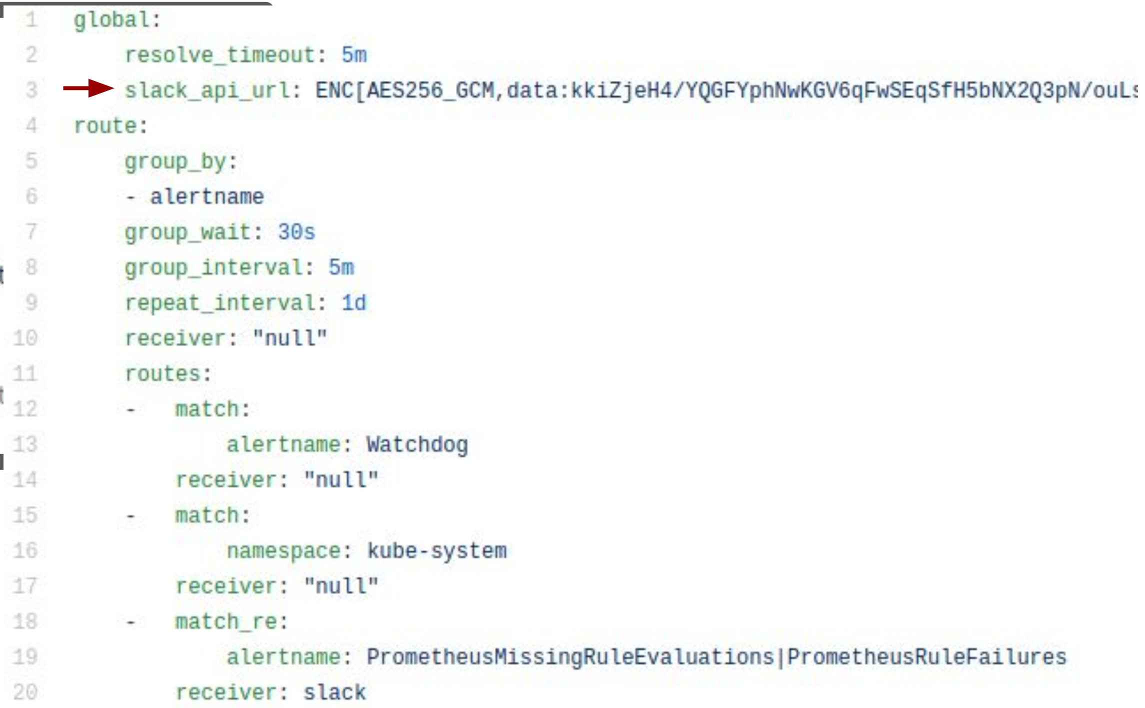
```yaml
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization

namespace: prometheus-operator

secretGenerator:
- name: alertmanager-prometheus-operator-alertmanager
  files:
  - sopsenc/alertmanager.yaml
generatorOptions: #this option is global for this kustomization
  disableNameSuffixHash: true
```

3 builtin secret generators:
https://github.com/kubernetes-sigs/kustomize/blob/master/examples/secretGeneratorPlugin.md#secret-values-from-local-files

Deployment - secrets



```yaml
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization

namespace: prometheus-operator

secretGenerator:
- name: alertmanager-prometheus-operator-alert
  files:
  - sopsenc/alertmanager.yaml
generatorOptions: #this option is global for t
  disableNameSuffixHash: true
```

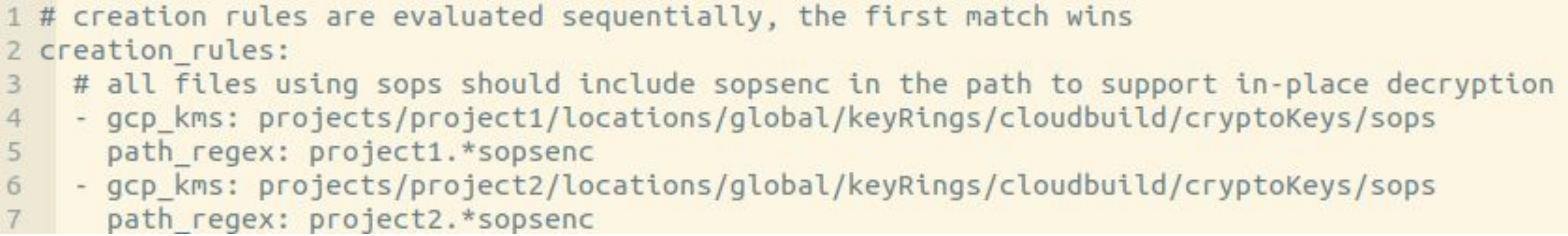


```yaml
global:
    resolve_timeout: 5m
    slack_api_url: ENC[AES256_GCM,data:kkiZjeH4/YQGFYphNwKGV6qFwSEqSfH5bNX2Q3pN/ouLs
route:
    group_by:
    - alertname
    group_wait: 30s
    group_interval: 5m
    repeat_interval: 1d
    receiver: "null"
    routes:
    -   match:
            alertname: Watchdog
        receiver: "null"
    -   match:
            namespace: kube-system
        receiver: "null"
    -   match_re:
            alertname: PrometheusMissingRuleEvaluations|PrometheusRuleFailures
        receiver: slack
```

```
sops -e -i --encrypted-suffix=api_url deploy/excitingdev/alertmanager-secret.sops.yaml
```

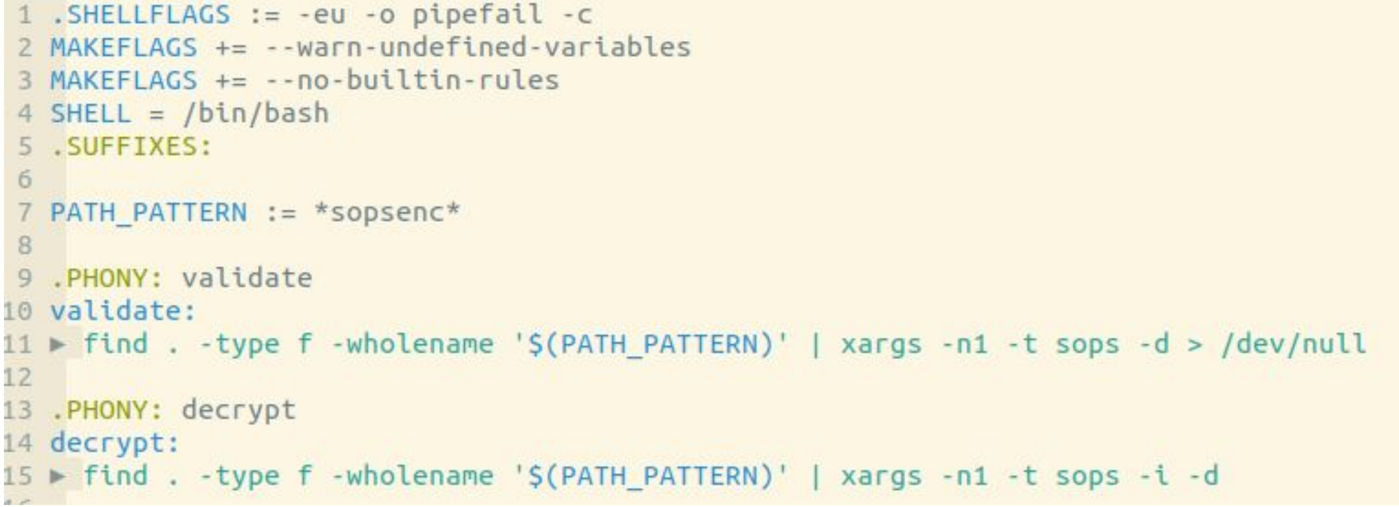
Deployment - secrets

.sops.yaml



```
# creation rules are evaluated sequentially, the first match wins
creation_rules:
  # all files using sops should include sopsenc in the path to support in-place decryption
  - gcp_kms: projects/project1/locations/global/keyRings/cloudbuild/cryptoKeys/sops
    path_regex: project1.*sopsenc
  - gcp_kms: projects/project2/locations/global/keyRings/cloudbuild/cryptoKeys/sops
    path_regex: project2.*sopsenc
```

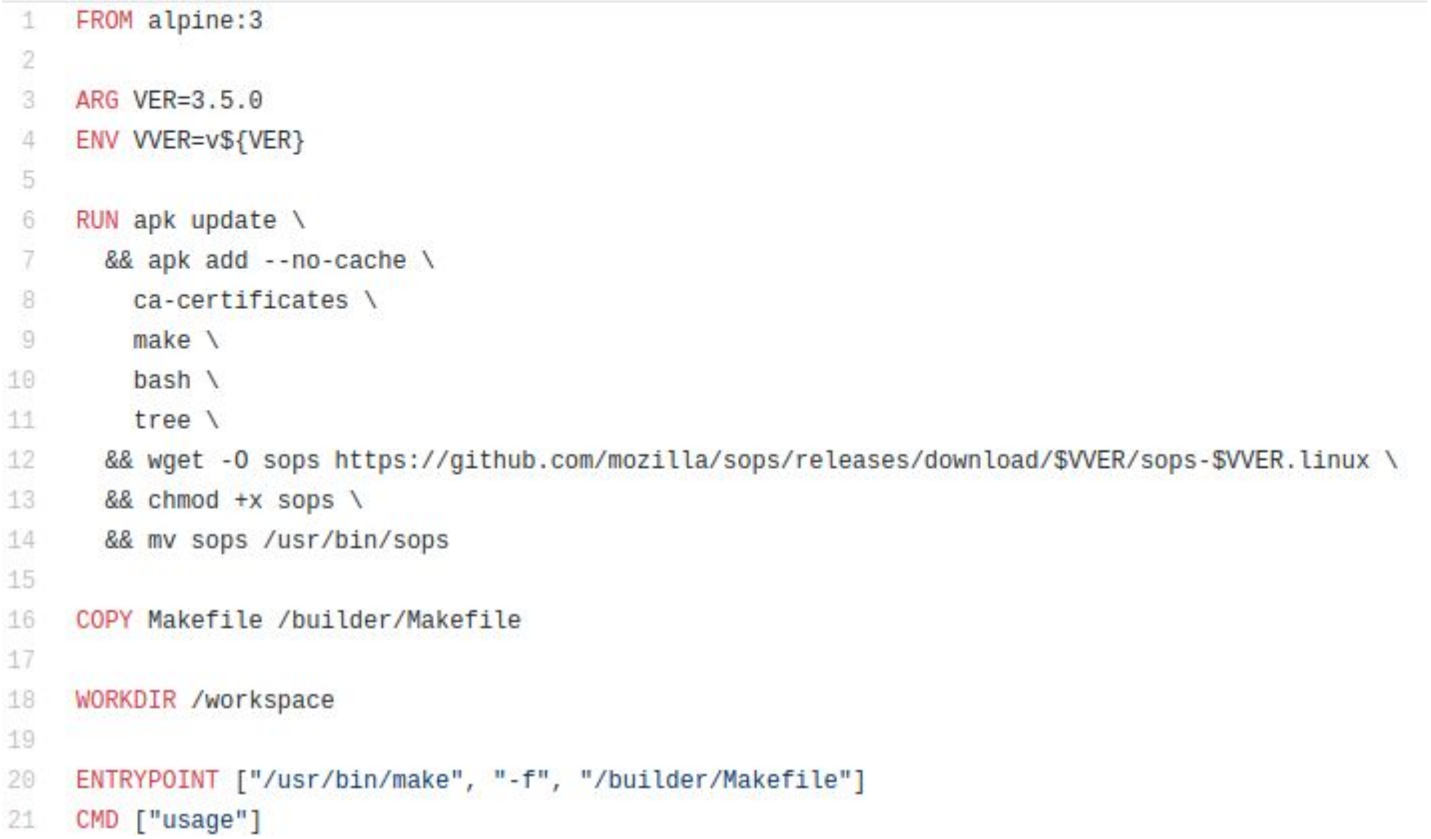
utility-images/gcloud-sops-slim/Makefile



```
.SHELLFLAGS := -eu -o pipefail -c
MAKEFLAGS += --warn-undefined-variables
MAKEFLAGS += --no-builtin-rules
SHELL = /bin/bash
.SUFFIXES:

PATH_PATTERN := *sopsenc*

.PHONY: validate
validate:
	find . -type f -wholename '$(PATH_PATTERN)' | xargs -n1 -t sops -d > /dev/null

.PHONY: decrypt
decrypt:
	find . -type f -wholename '$(PATH_PATTERN)' | xargs -n1 -t sops -i -d
```

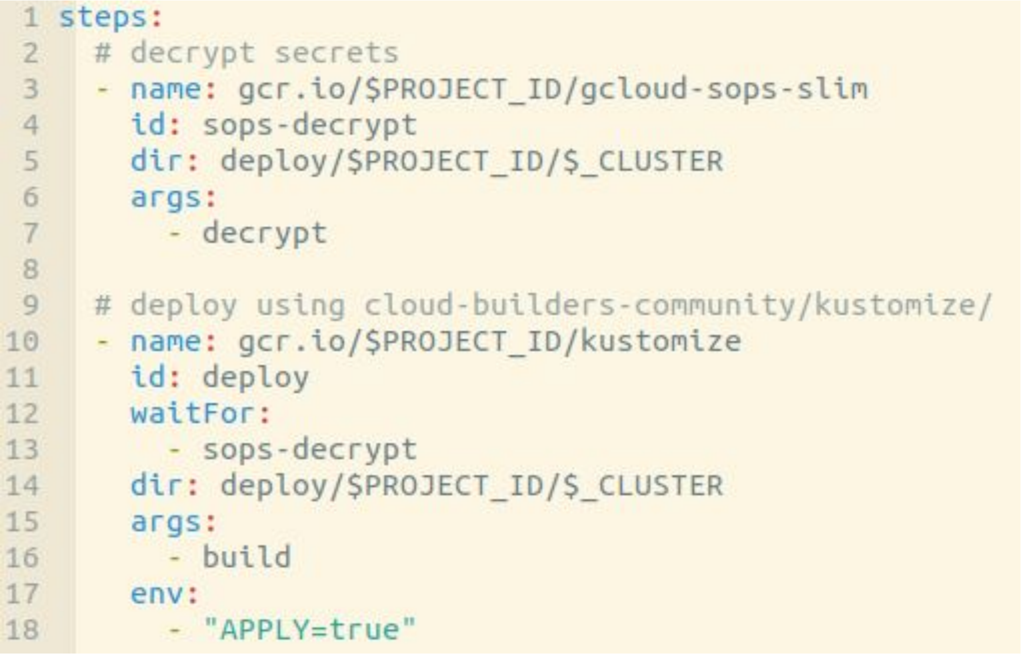
Deployment - secrets

utility-images/gcloud-sops-slim/Dockerfile



```
FROM alpine:3

ARG VER=3.5.0
ENV VVER=v${VER}

RUN apk update \
  && apk add --no-cache \
    ca-certificates \
    make \
    bash \
    tree \
  && wget -O sops https://github.com/mozilla/sops/releases/download/$VVER/sops-$VVER.linux \
  && chmod +x sops \
  && mv sops /usr/bin/sops

COPY Makefile /builder/Makefile

WORKDIR /workspace

ENTRYPOINT ["/usr/bin/make", "-f", "/builder/Makefile"]
CMD ["usage"]
```

Deployment - pipelines



```yaml
steps:
  # decrypt secrets
  - name: gcr.io/$PROJECT_ID/gcloud-sops-slim
    id: sops-decrypt
    dir: deploy/$PROJECT_ID/$_CLUSTER
    args:
      - decrypt

  # deploy using cloud-builders-community/kustomize/
  - name: gcr.io/$PROJECT_ID/kustomize
    id: deploy
    waitFor:
      - sops-decrypt
    dir: deploy/$PROJECT_ID/$_CLUSTER
    args:
      - build
    env:
      - "APPLY=true"
```

github.com/afirth/kceu2020

Deployment - secrets

- HashiCorp Vault
  - Secrets remain encrypted until sidecar starts
  - Secrets not in git
  - Plugin available for kustomize (removes most benefits vs sops)
  - No partial encryption
- Bitnami sealed-secrets
  - Decrypted in cluster by operator
  - No suffix hashing
  - No partial encryption
- Manage secrets separately
  - Cloud providers or manually

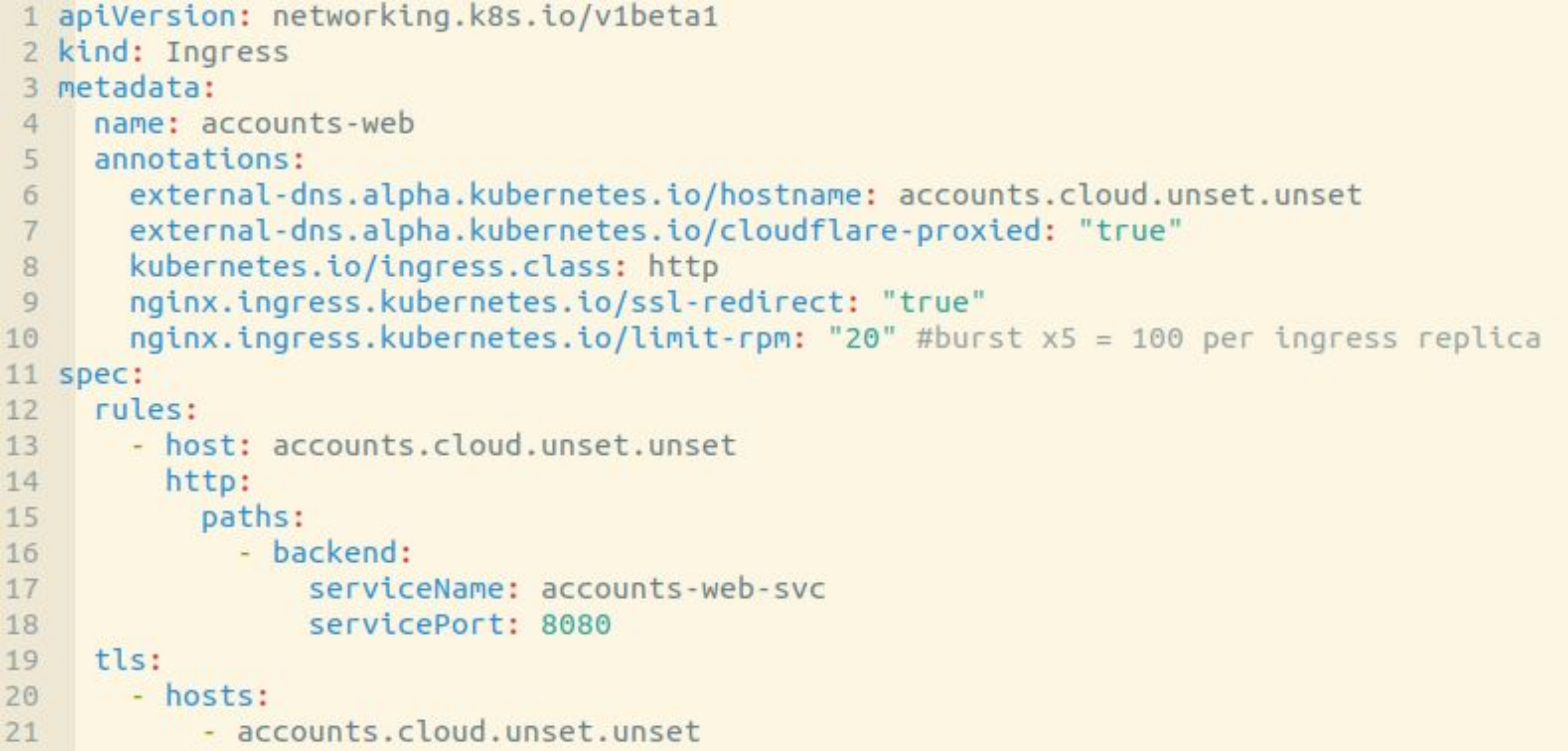
Painful bits SOLVED!

- Deployment
  - Helm template + kustomize = plain manifests
  - ++ patches and secrets with kustomize
- Following upstream changes
  - make fetch && make generate

Fun bits

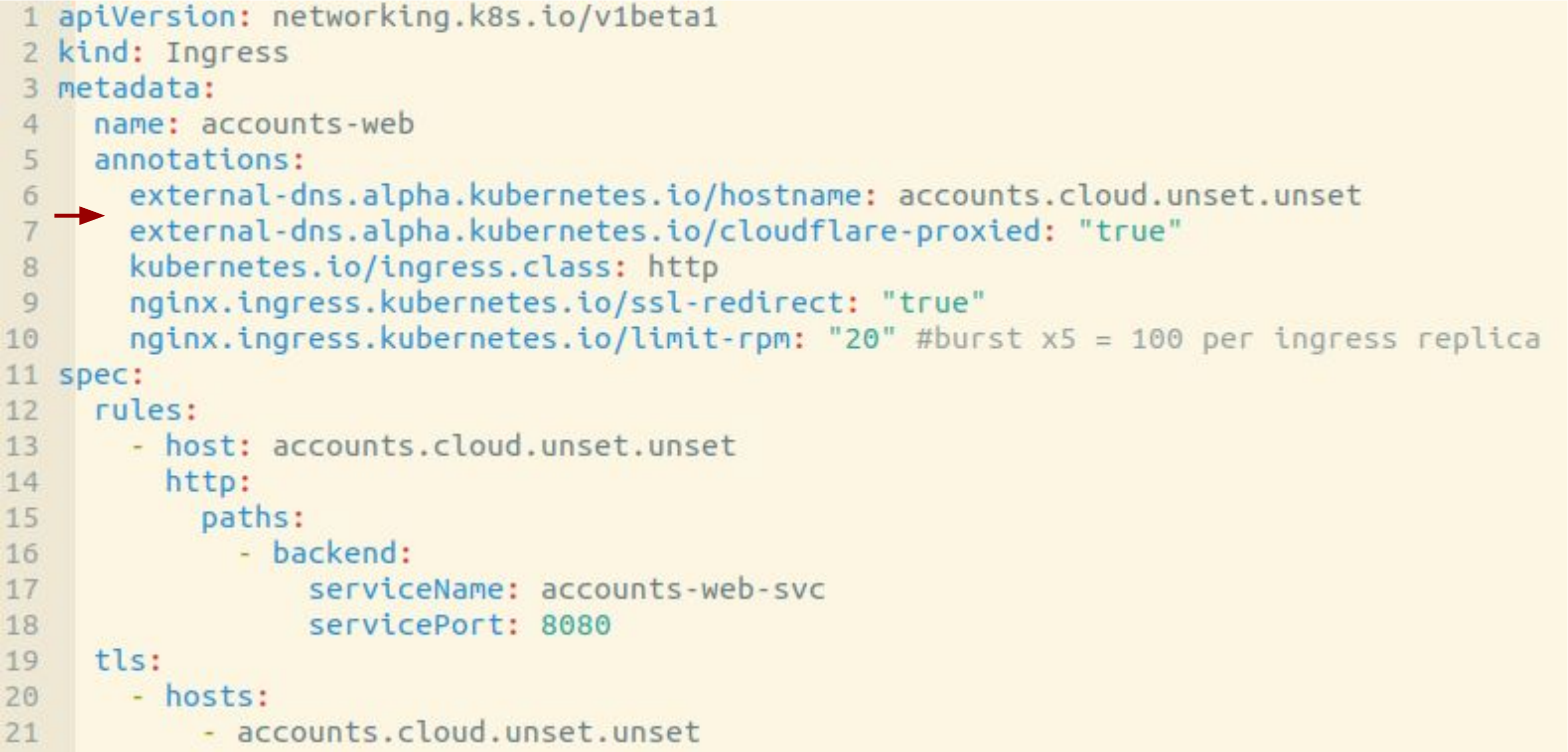
- DNS records
  - external-dns
- TLS certs
  - cert-manager
- Monitoring
  - prometheus-operator
- AuthN/AuthZ
  - oauth2-proxy/oauth2-proxy
- Reverse proxy
  - kubernetes/ingress-nginx

Anatomy of an ingress



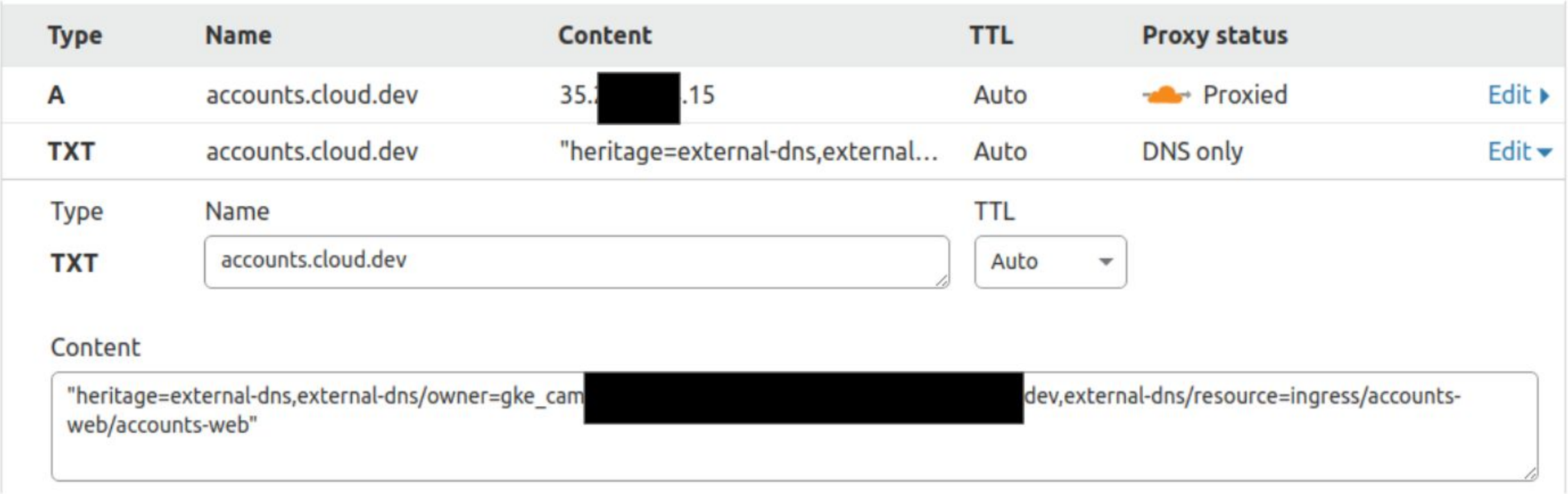
```yaml
apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
  name: accounts-web
  annotations:
    external-dns.alpha.kubernetes.io/hostname: accounts.cloud.unset.unset
    external-dns.alpha.kubernetes.io/cloudflare-proxied: "true"
    kubernetes.io/ingress.class: http
    nginx.ingress.kubernetes.io/ssl-redirect: "true"
    nginx.ingress.kubernetes.io/limit-rpm: "20" #burst x5 = 100 per ingress replica
spec:
  rules:
    - host: accounts.cloud.unset.unset
      http:
        paths:
          - backend:
              serviceName: accounts-web-svc
              servicePort: 8080
  tls:
    - hosts:
      - accounts.cloud.unset.unset
```

Anatomy of an ingress



```yaml
apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
  name: accounts-web
  annotations:
    external-dns.alpha.kubernetes.io/hostname: accounts.cloud.unset.unset
    external-dns.alpha.kubernetes.io/cloudflare-proxied: "true"
    kubernetes.io/ingress.class: http
    nginx.ingress.kubernetes.io/ssl-redirect: "true"
    nginx.ingress.kubernetes.io/limit-rpm: "20" #burst x5 = 100 per ingress replica
spec:
  rules:
    - host: accounts.cloud.unset.unset
      http:
        paths:
          - backend:
              serviceName: accounts-web-svc
              servicePort: 8080
  tls:
    - hosts:
        - accounts.cloud.unset.unset
```

Anatomy of an ingress



Type	Name	Content	TTL	Proxy status	
A	accounts.cloud.dev	35.[redacted].15	Auto	Proxied	Edit ▸
TXT	accounts.cloud.dev	"heritage=external-dns,external...	Auto	DNS only	Edit ▾

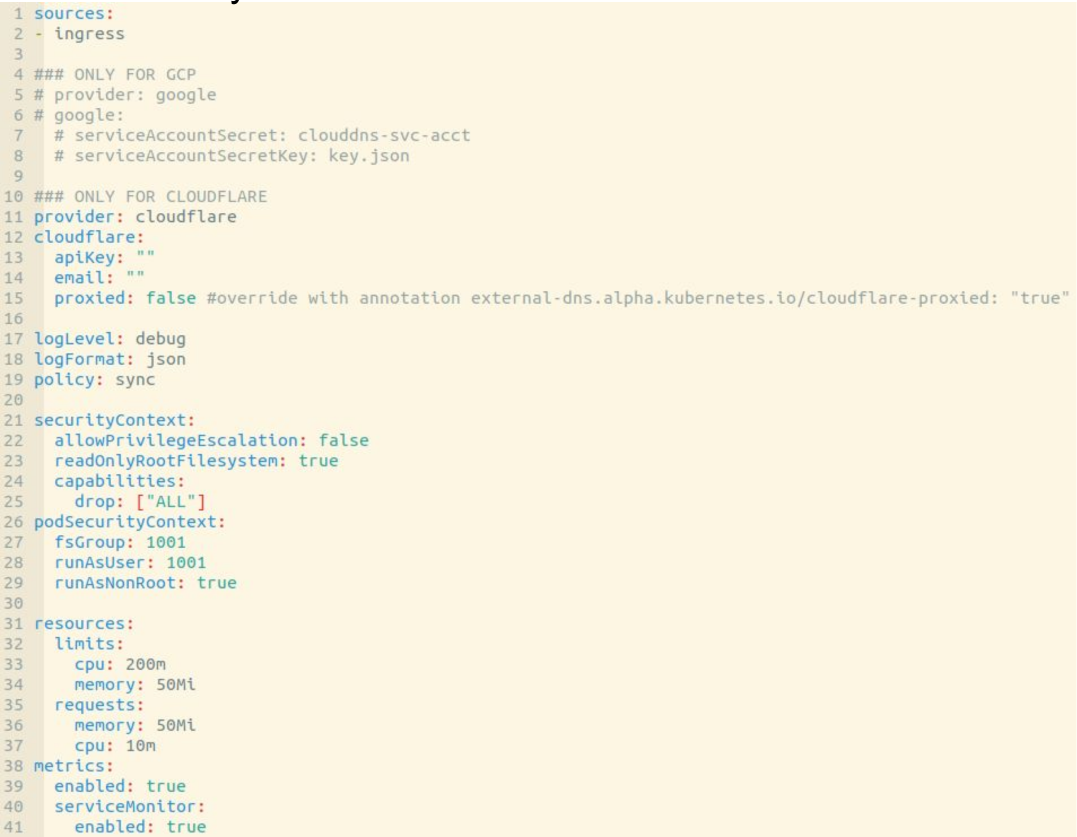
Type	Name	TTL
TXT	accounts.cloud.dev	Auto ▾

Content

"heritage=external-dns,external-dns/owner=gke_cam[redacted]dev,external-dns/resource=ingress/accounts-web/accounts-web"

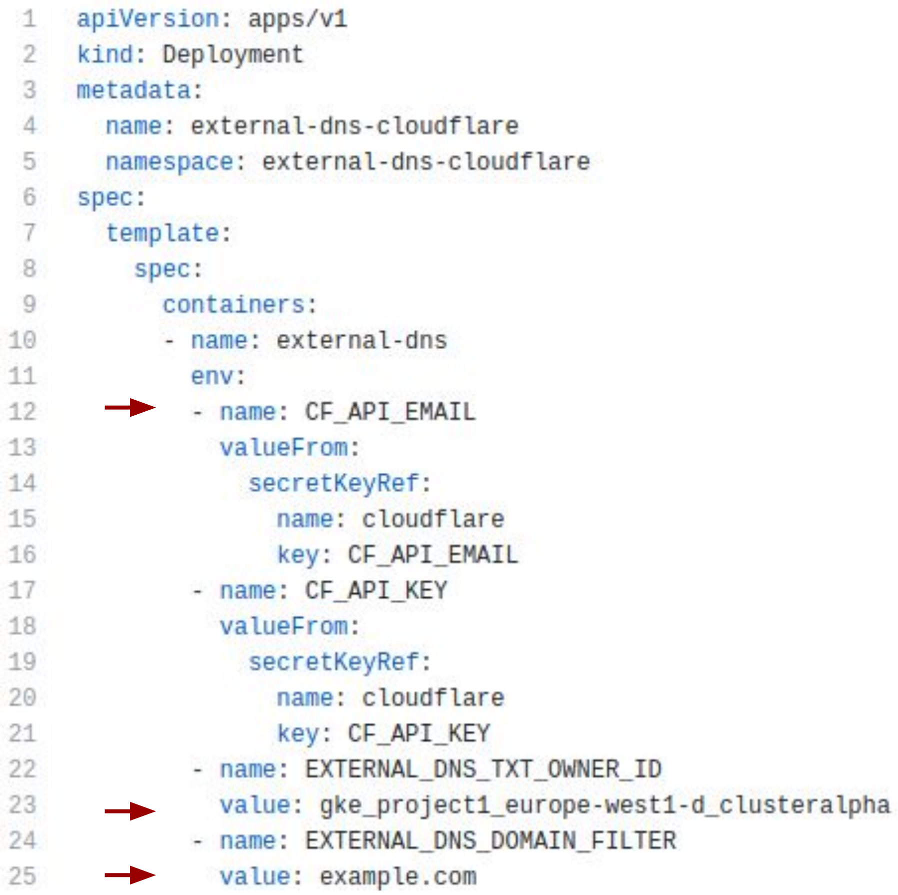
External-dns

helm-values.yaml



```yaml
sources:
- ingress

### ONLY FOR GCP
# provider: google
# google:
  # serviceAccountSecret: clouddns-svc-acct
  # serviceAccountSecretKey: key.json

### ONLY FOR CLOUDFLARE
provider: cloudflare
cloudflare:
  apiKey: ""
  email: ""
  proxied: false #override with annotation external-dns.alpha.kubernetes.io/cloudflare-proxied: "true"

logLevel: debug
logFormat: json
policy: sync

securityContext:
  allowPrivilegeEscalation: false
  readOnlyRootFilesystem: true
  capabilities:
    drop: ["ALL"]
podSecurityContext:
  fsGroup: 1001
  runAsUser: 1001
  runAsNonRoot: true

resources:
  limits:
    cpu: 200m
    memory: 50Mi
  requests:
    memory: 50Mi
    cpu: 10m
metrics:
  enabled: true
  serviceMonitor:
    enabled: true
```

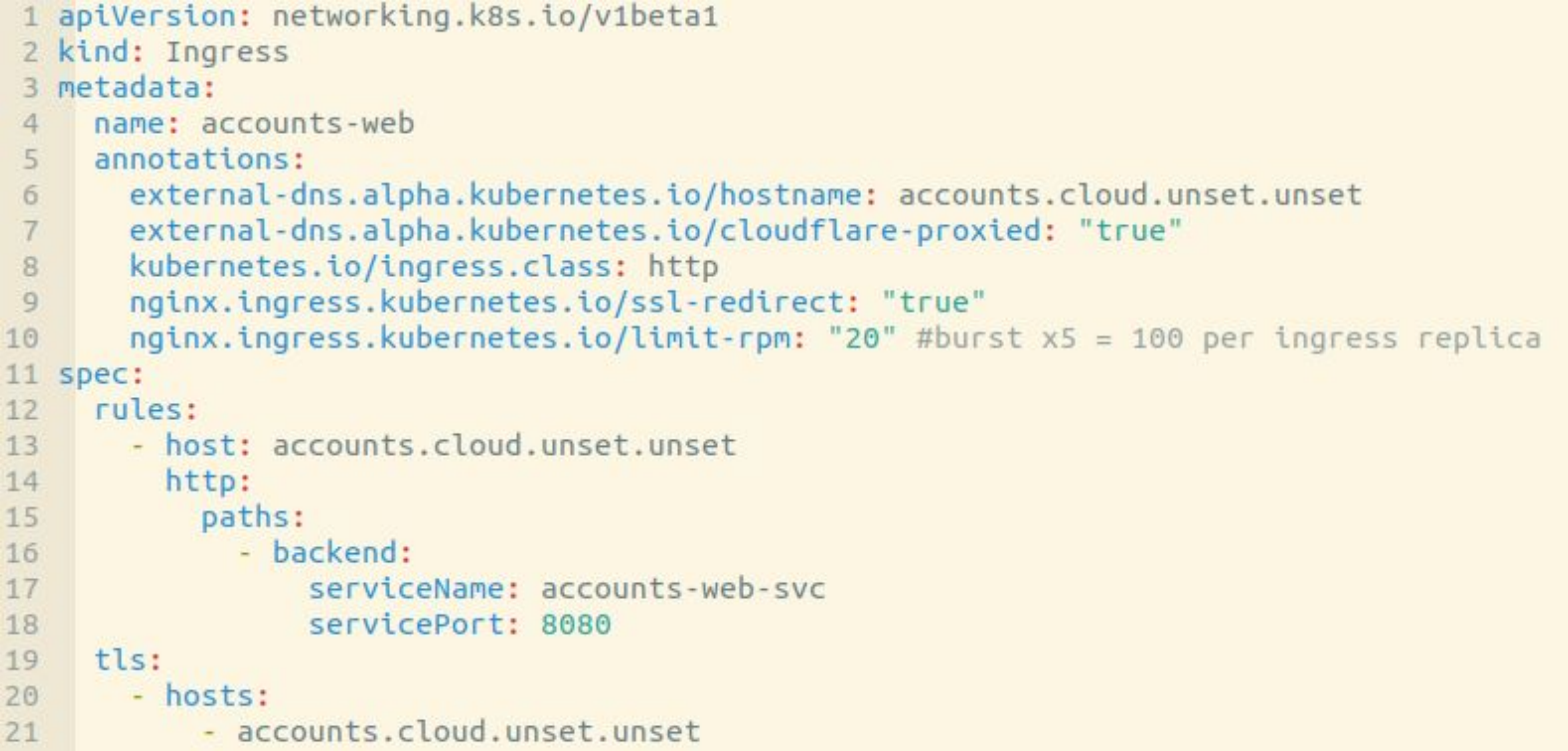
External-dns



```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: external-dns-cloudflare
  namespace: external-dns-cloudflare
spec:
  template:
    spec:
      containers:
      - name: external-dns
        env:
        - name: CF_API_EMAIL
          valueFrom:
            secretKeyRef:
              name: cloudflare
              key: CF_API_EMAIL
        - name: CF_API_KEY
          valueFrom:
            secretKeyRef:
              name: cloudflare
              key: CF_API_KEY
        - name: EXTERNAL_DNS_TXT_OWNER_ID
          value: gke_project1_europe-west1-d_clusteralpha
        - name: EXTERNAL_DNS_DOMAIN_FILTER
          value: example.com
```

GKE scope
https://www.googleapis.com/auth/ndev.clouddns.readwrite
Similar for other providers

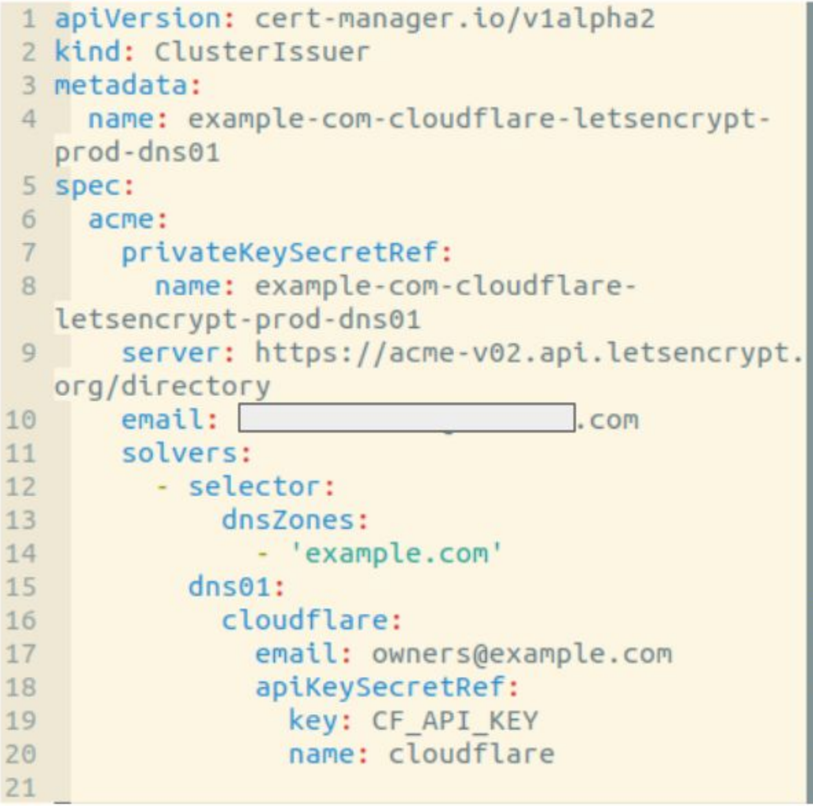
Anatomy of an ingress



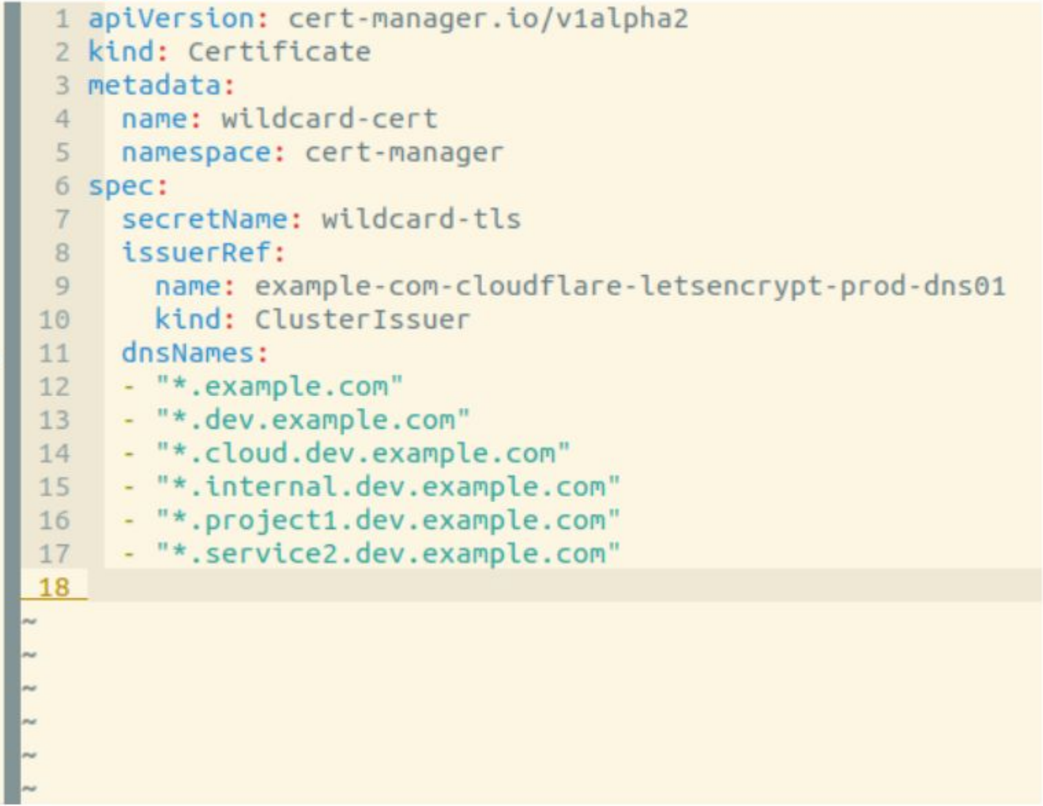
```yaml
apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
  name: accounts-web
  annotations:
    external-dns.alpha.kubernetes.io/hostname: accounts.cloud.unset.unset
    external-dns.alpha.kubernetes.io/cloudflare-proxied: "true"
    kubernetes.io/ingress.class: http
    nginx.ingress.kubernetes.io/ssl-redirect: "true"
    nginx.ingress.kubernetes.io/limit-rpm: "20" #burst x5 = 100 per ingress replica
spec:
  rules:
    - host: accounts.cloud.unset.unset
      http:
        paths:
          - backend:
              serviceName: accounts-web-svc
              servicePort: 8080
  tls:
    - hosts:
      - accounts.cloud.unset.unset
```

→ secretName: ????

Cert-manager - CRDs



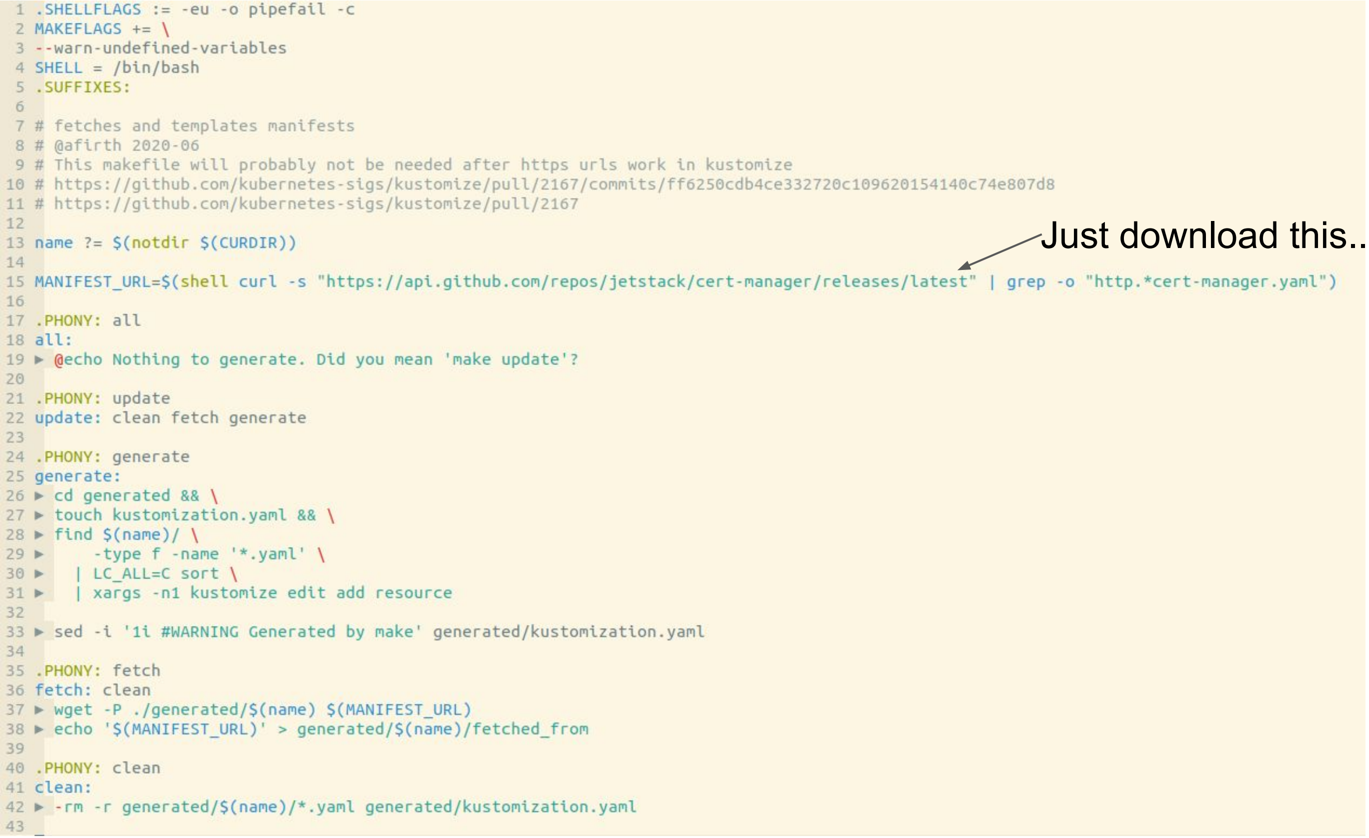
```yaml
apiVersion: cert-manager.io/v1alpha2
kind: ClusterIssuer
metadata:
  name: example-com-cloudflare-letsencrypt-prod-dns01
spec:
  acme:
    privateKeySecretRef:
      name: example-com-cloudflare-letsencrypt-prod-dns01
    server: https://acme-v02.api.letsencrypt.org/directory
    email: [redacted].com
    solvers:
      - selector:
          dnsZones:
            - 'example.com'
        dns01:
          cloudflare:
            email: owners@example.com
            apiKeySecretRef:
              key: CF_API_KEY
              name: cloudflare

```



```yaml
apiVersion: cert-manager.io/v1alpha2
kind: Certificate
metadata:
  name: wildcard-cert
  namespace: cert-manager
spec:
  secretName: wildcard-tls
  issuerRef:
    name: example-com-cloudflare-letsencrypt-prod-dns01
    kind: ClusterIssuer
  dnsNames:
  - "*.example.com"
  - "*.dev.example.com"
  - "*.cloud.dev.example.com"
  - "*.internal.dev.example.com"
  - "*.project1.dev.example.com"
  - "*.service2.dev.example.com"

```

github.com/afirth/kceu2020

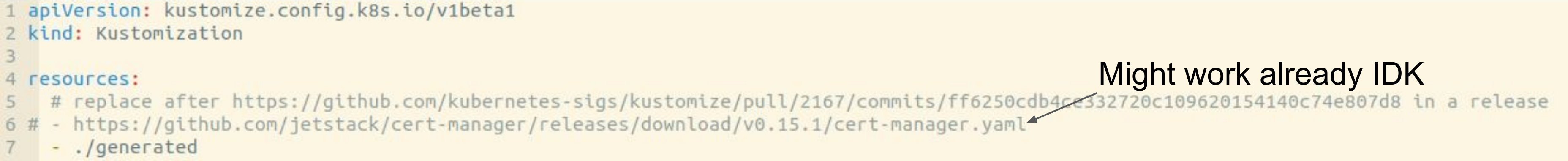
Cert-manager - base



```
.SHELLFLAGS := -eu -o pipefail -c
MAKEFLAGS += \
--warn-undefined-variables
SHELL = /bin/bash
.SUFFIXES:

# fetches and templates manifests
# @afirth 2020-06
# This makefile will probably not be needed after https urls work in kustomize
# https://github.com/kubernetes-sigs/kustomize/pull/2167/commits/ff6250cdb4ce332720c109620154140c74e807d8
# https://github.com/kubernetes-sigs/kustomize/pull/2167

name ?= $(notdir $(CURDIR))

MANIFEST_URL=$(shell curl -s "https://api.github.com/repos/jetstack/cert-manager/releases/latest" | grep -o "http.*cert-manager.yaml")

.PHONY: all
all:
▶ @echo Nothing to generate. Did you mean 'make update'?

.PHONY: update
update: clean fetch generate

.PHONY: generate
generate:
▶ cd generated && \
▶ touch kustomization.yaml && \
▶ find $(name)/ \
▶     -type f -name '*.yaml' \
▶   | LC_ALL=C sort \
▶   | xargs -n1 kustomize edit add resource

▶ sed -i '1i #WARNING Generated by make' generated/kustomization.yaml

.PHONY: fetch
fetch: clean
▶ wget -P ./generated/$(name) $(MANIFEST_URL)
▶ echo '$(MANIFEST_URL)' > generated/$(name)/fetched_from

.PHONY: clean
clean:
▶ -rm -r generated/$(name)/*.yaml generated/kustomization.yaml
```

Just download this...

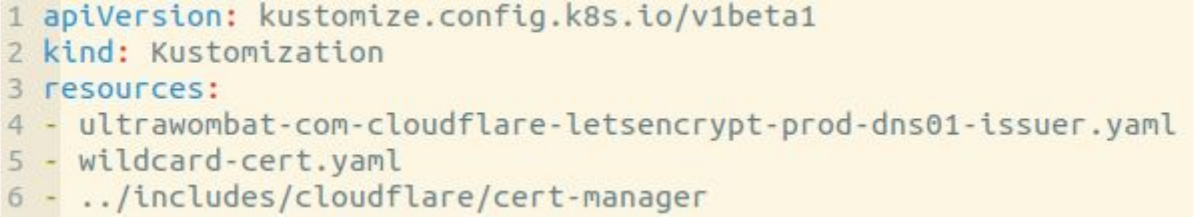
Cert-manager - base



```yaml
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization

resources:
  # replace after https://github.com/kubernetes-sigs/kustomize/pull/2167/commits/ff6250cdb4ce332720c109620154140c74e807d8 in a release
# - https://github.com/jetstack/cert-manager/releases/download/v0.15.1/cert-manager.yaml
  - ./generated
```

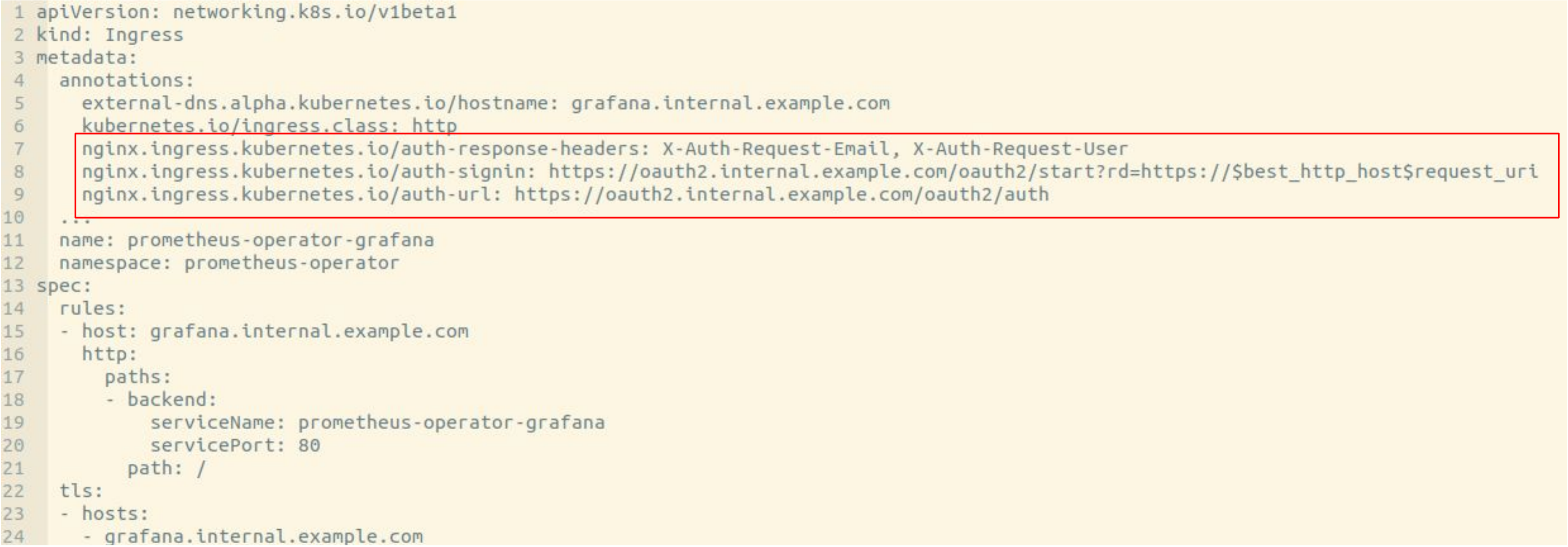
Might work already IDK

Cert-manager - overlay



```yaml
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization
resources:
- ultrawombat-com-cloudflare-letsencrypt-prod-dns01-issuer.yaml
- wildcard-cert.yaml
- ../includes/cloudflare/cert-manager
```

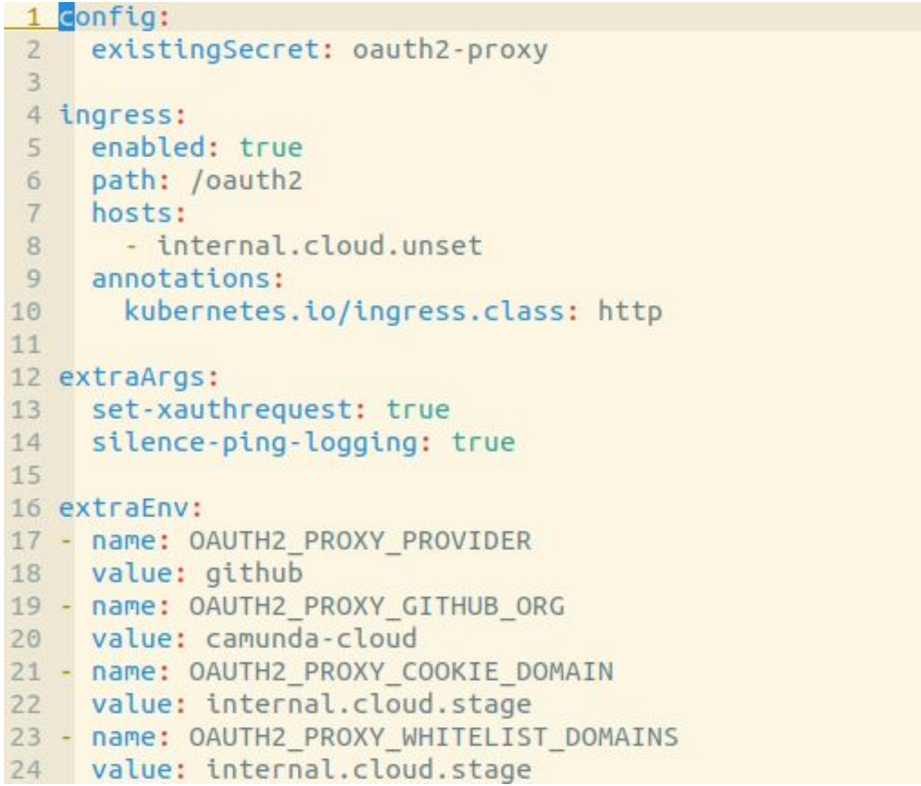
Oauth2-proxy - sample ingress



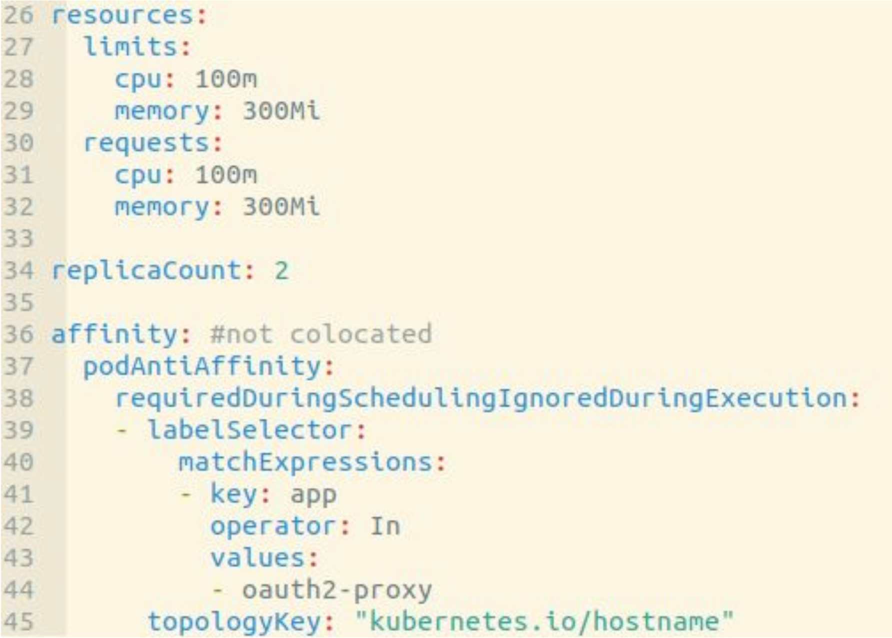
```yaml
apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
  annotations:
    external-dns.alpha.kubernetes.io/hostname: grafana.internal.example.com
    kubernetes.io/ingress.class: http
    nginx.ingress.kubernetes.io/auth-response-headers: X-Auth-Request-Email, X-Auth-Request-User
    nginx.ingress.kubernetes.io/auth-signin: https://oauth2.internal.example.com/oauth2/start?rd=https://$best_http_host$request_uri
    nginx.ingress.kubernetes.io/auth-url: https://oauth2.internal.example.com/oauth2/auth
  ...
  name: prometheus-operator-grafana
  namespace: prometheus-operator
spec:
  rules:
  - host: grafana.internal.example.com
    http:
      paths:
      - backend:
          serviceName: prometheus-operator-grafana
          servicePort: 80
        path: /
  tls:
  - hosts:
    - grafana.internal.example.com
```

github.com/afirth/kceu2020

Oauth2-proxy - helm values



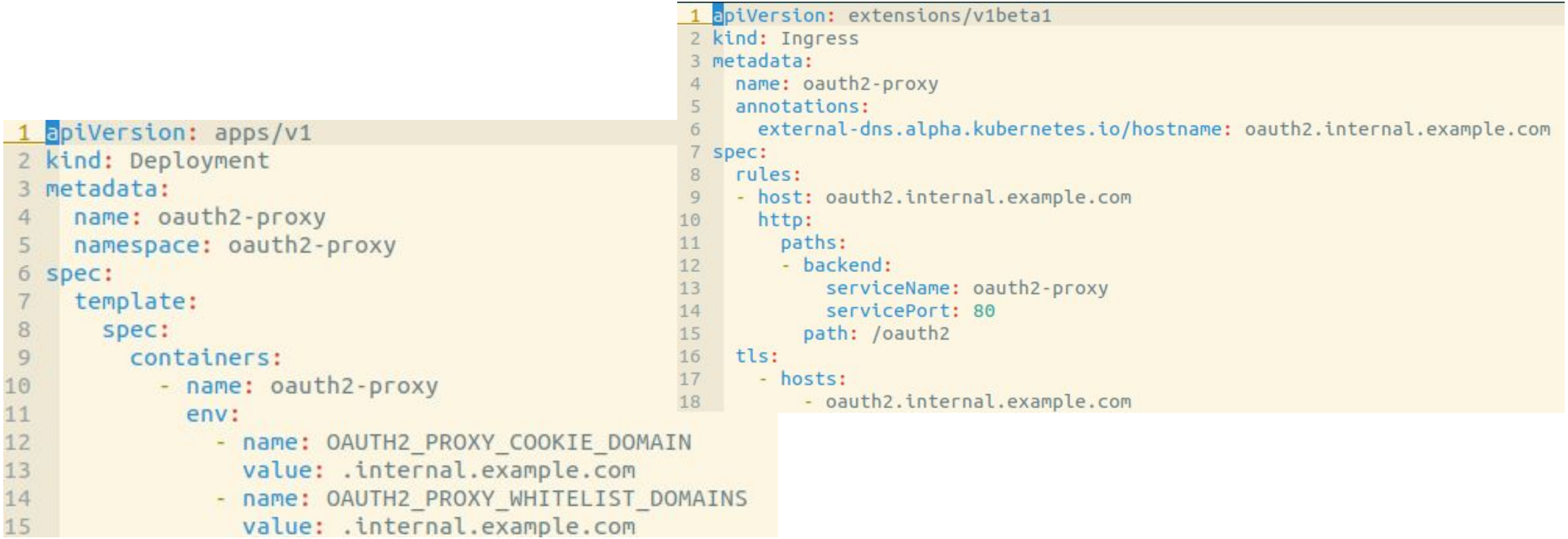
```yaml
config:
  existingSecret: oauth2-proxy

ingress:
  enabled: true
  path: /oauth2
  hosts:
    - internal.cloud.unset
  annotations:
    kubernetes.io/ingress.class: http

extraArgs:
  set-xauthrequest: true
  silence-ping-logging: true

extraEnv:
- name: OAUTH2_PROXY_PROVIDER
  value: github
- name: OAUTH2_PROXY_GITHUB_ORG
  value: camunda-cloud
- name: OAUTH2_PROXY_COOKIE_DOMAIN
  value: internal.cloud.stage
- name: OAUTH2_PROXY_WHITELIST_DOMAINS
  value: internal.cloud.stage
```



```yaml
resources:
  limits:
    cpu: 100m
    memory: 300Mi
  requests:
    cpu: 100m
    memory: 300Mi

replicaCount: 2

affinity: #not colocated
  podAntiAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
    - labelSelector:
        matchExpressions:
        - key: app
          operator: In
          values:
          - oauth2-proxy
      topologyKey: "kubernetes.io/hostname"
```

github.com/afirth/kceu2020

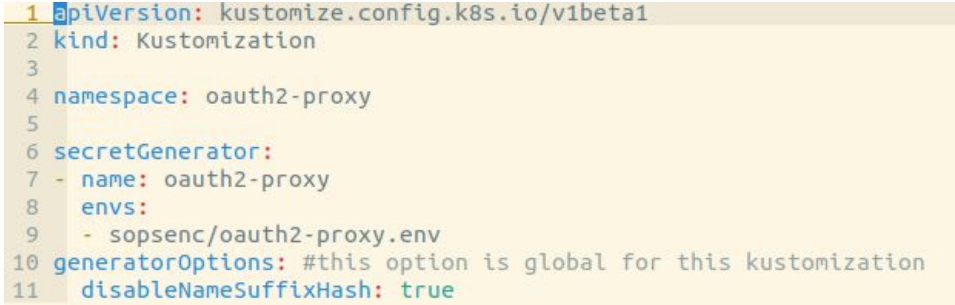
Oauth2-proxy - patches



```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: oauth2-proxy
  namespace: oauth2-proxy
spec:
  template:
    spec:
      containers:
        - name: oauth2-proxy
          env:
            - name: OAUTH2_PROXY_COOKIE_DOMAIN
              value: .internal.example.com
            - name: OAUTH2_PROXY_WHITELIST_DOMAINS
              value: .internal.example.com
```

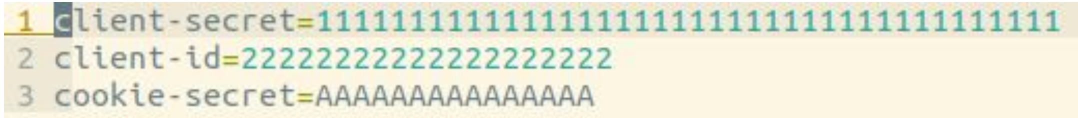
```yaml
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: oauth2-proxy
  annotations:
    external-dns.alpha.kubernetes.io/hostname: oauth2.internal.example.com
spec:
  rules:
  - host: oauth2.internal.example.com
    http:
      paths:
      - backend:
          serviceName: oauth2-proxy
          servicePort: 80
        path: /oauth2
  tls:
    - hosts:
      - oauth2.internal.example.com
```

Oauth2-proxy - secret



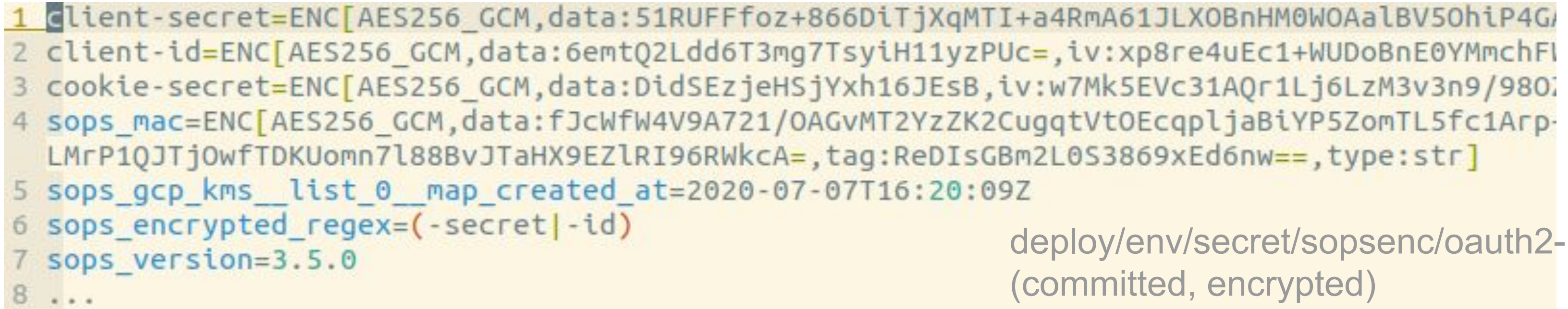
```yaml
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization

namespace: oauth2-proxy

secretGenerator:
- name: oauth2-proxy
  envs:
  - sopsenc/oauth2-proxy.env
generatorOptions: #this option is global for this kustomization
  disableNameSuffixHash: true
```

deploy/env/secret/kustomization.yaml



```
client-secret=1111111111111111111111111111111111111111
client-id=22222222222222222222
cookie-secret=AAAAAAAAAAAAAAA
```

deploy/env/secret/sopsenc/oauth2-proxy.env (plaintext)



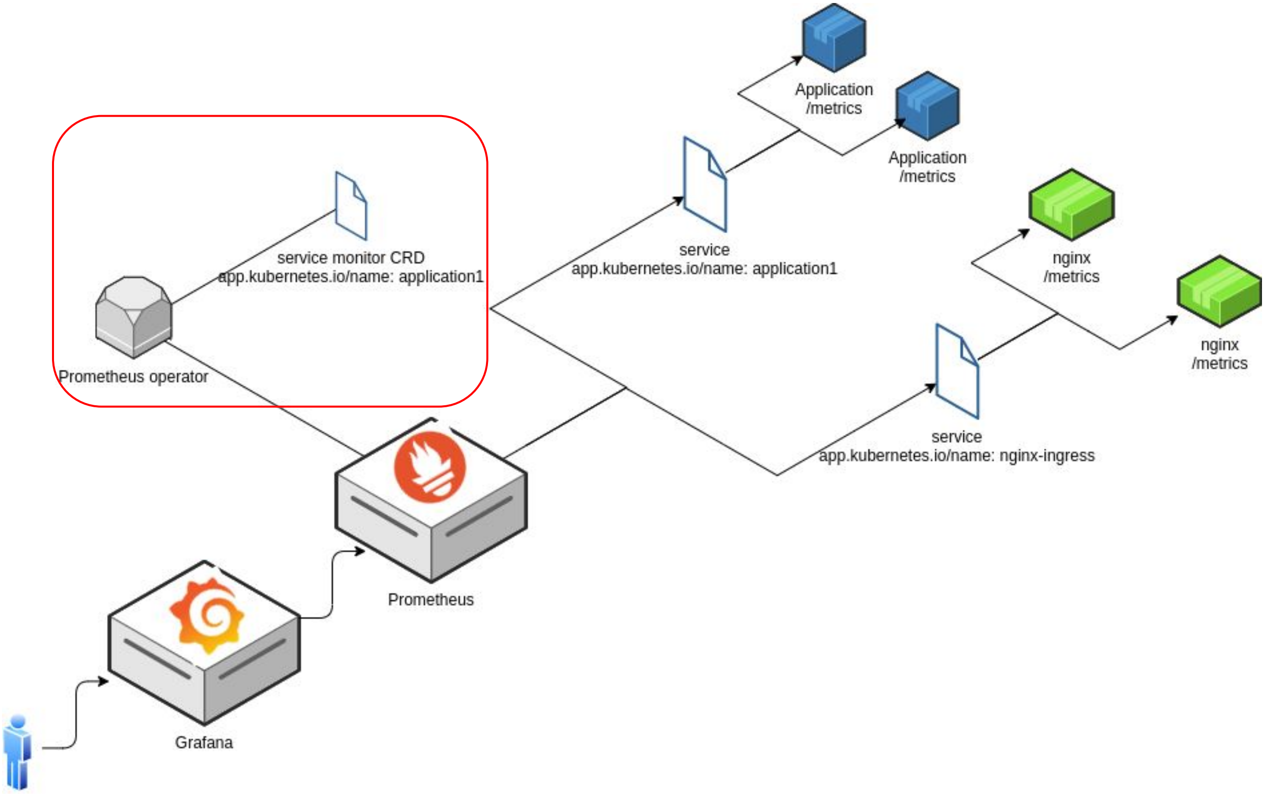
```
client-secret=ENC[AES256_GCM,data:51RUFFfoz+866DiTjXqMTI+a4RmA61JLXOBnHM0WOAalBV5OhiP4G
client-id=ENC[AES256_GCM,data:6emtQ2Ldd6T3mg7TsyiH11yzPUc=,iv:xp8re4uEc1+WUDoBnE0YMmchF
cookie-secret=ENC[AES256_GCM,data:DidSEzjeHSjYxh16JEsB,iv:w7Mk5EVc31AQr1Lj6LzM3v3n9/98O
sops_mac=ENC[AES256_GCM,data:fJcWfW4V9A721/OAGvMT2YzZK2CugqtVtOEcqpljaBiYP5ZomTL5fc1Arp-
LMrP1QJTjOwfTDKUomn7l88BvJTaHX9EZlRI96RWkcA=,tag:ReDIsGBm2L0S3869xEd6nw==,type:str]
sops_gcp_kms__list_0__map_created_at=2020-07-07T16:20:09Z
sops_encrypted_regex=(-secret|-id)
sops_version=3.5.0
...
```

deploy/env/secret/sopsenc/oauth2-proxy.env (committed, encrypted)

github.com/afirth/kceu2020

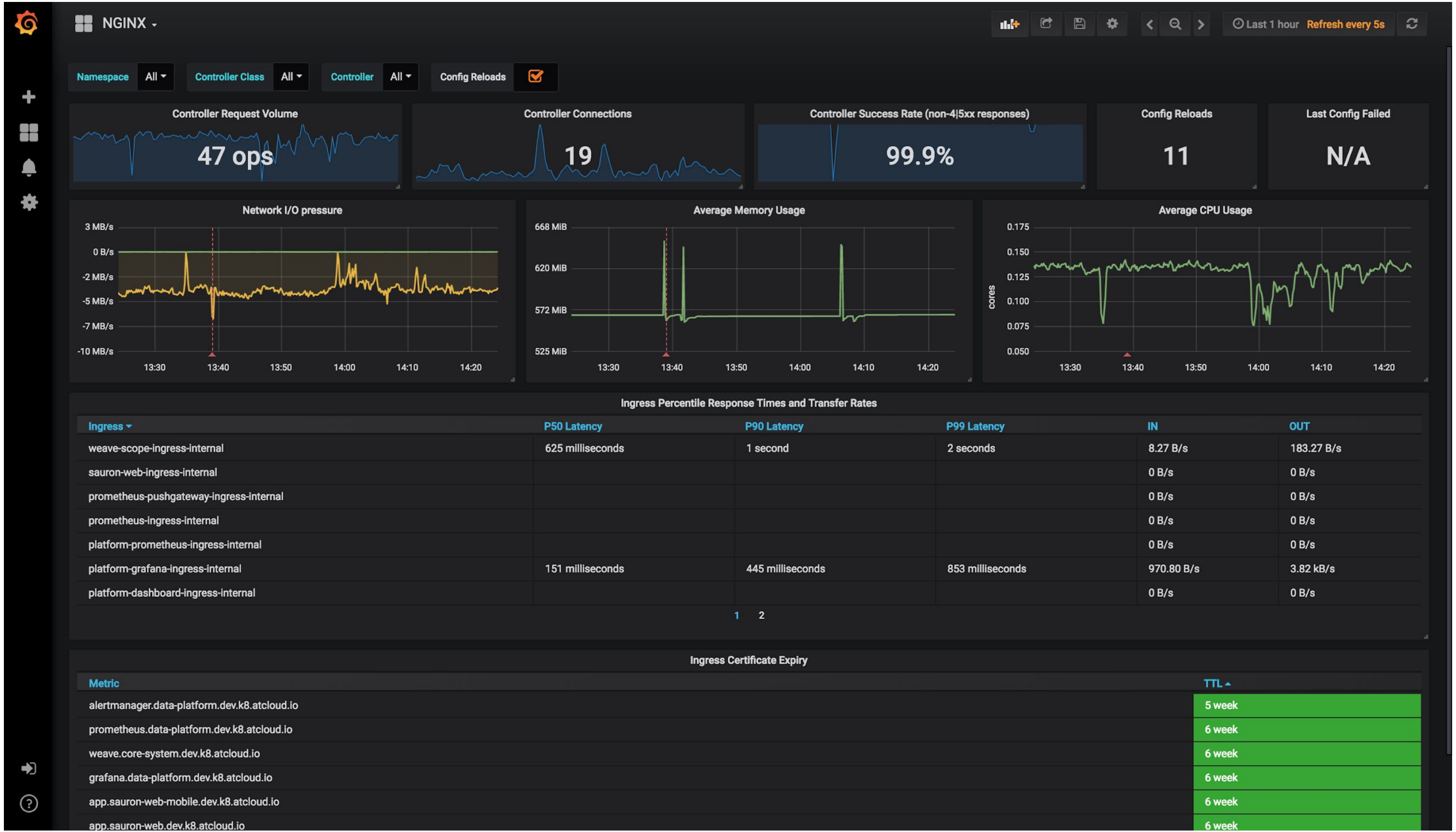
Monitoring - overview

https://github.com/helm/charts/tree/master/stable/prometheus-operator



Monitoring - overview

https://github.com/kubernetes/ingress-nginx/tree/master/deploy/grafana/dashboards



Namespace: All
Controller Class: All
Controller: All
Config Reloads ☑

Controller Request Volume: 47 ops

Controller Connections: 19

Controller Success Rate (non-4|5xx responses): 99.9%

Config Reloads: 11

Last Config Failed: N/A

Ingress Percentile Response Times and Transfer Rates

Ingress	P50 Latency	P90 Latency	P99 Latency	IN	OUT
weave-scope-ingress-internal	625 milliseconds	1 second	2 seconds	8.27 B/s	183.27 B/s
sauron-web-ingress-internal				0 B/s	0 B/s
prometheus-pushgateway-ingress-internal				0 B/s	0 B/s
prometheus-ingress-internal				0 B/s	0 B/s
platform-prometheus-ingress-internal				0 B/s	0 B/s
platform-grafana-ingress-internal	151 milliseconds	445 milliseconds	853 milliseconds	970.80 B/s	3.82 kB/s
platform-dashboard-ingress-internal				0 B/s	0 B/s

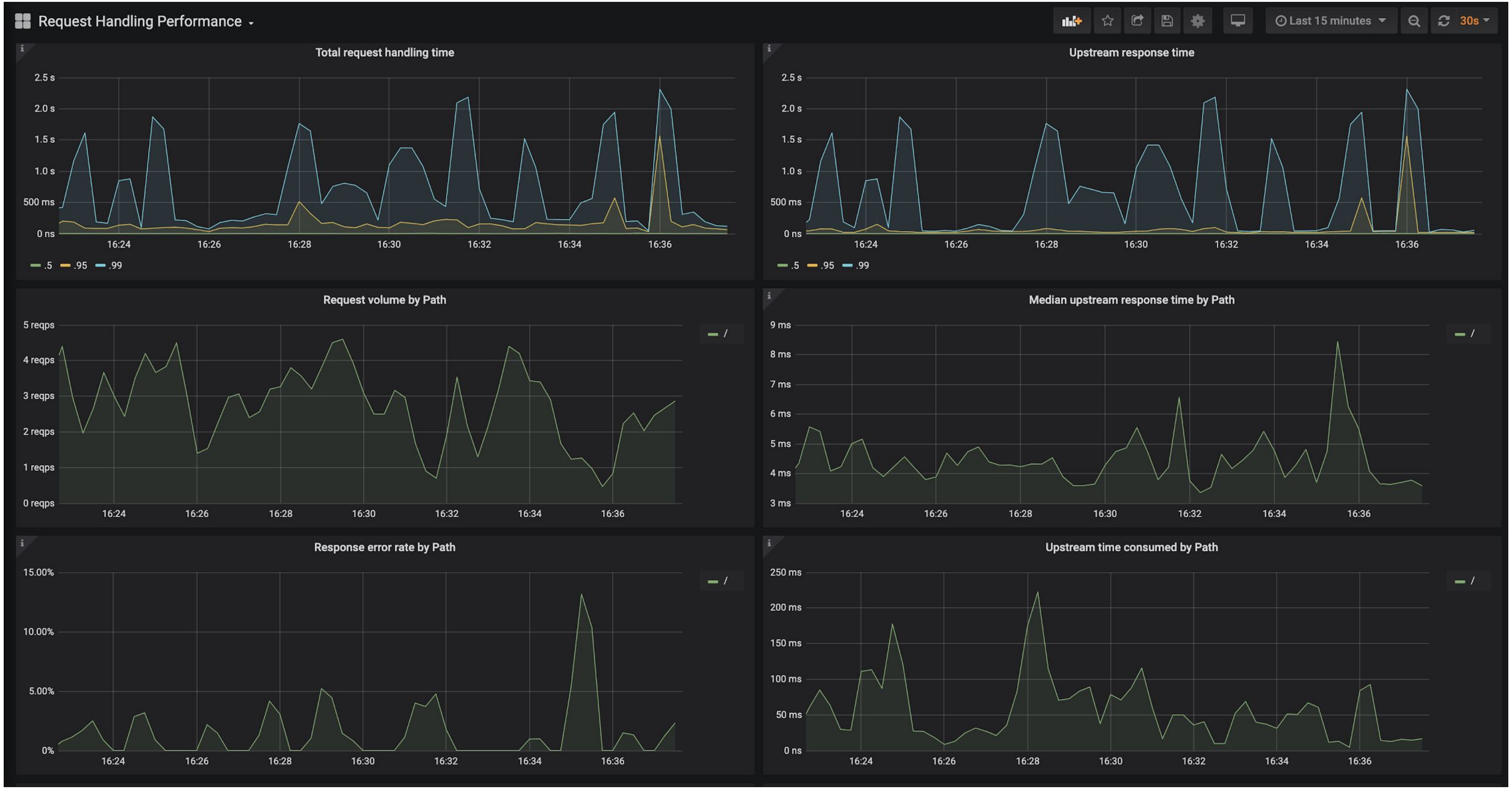
1 2

Ingress Certificate Expiry

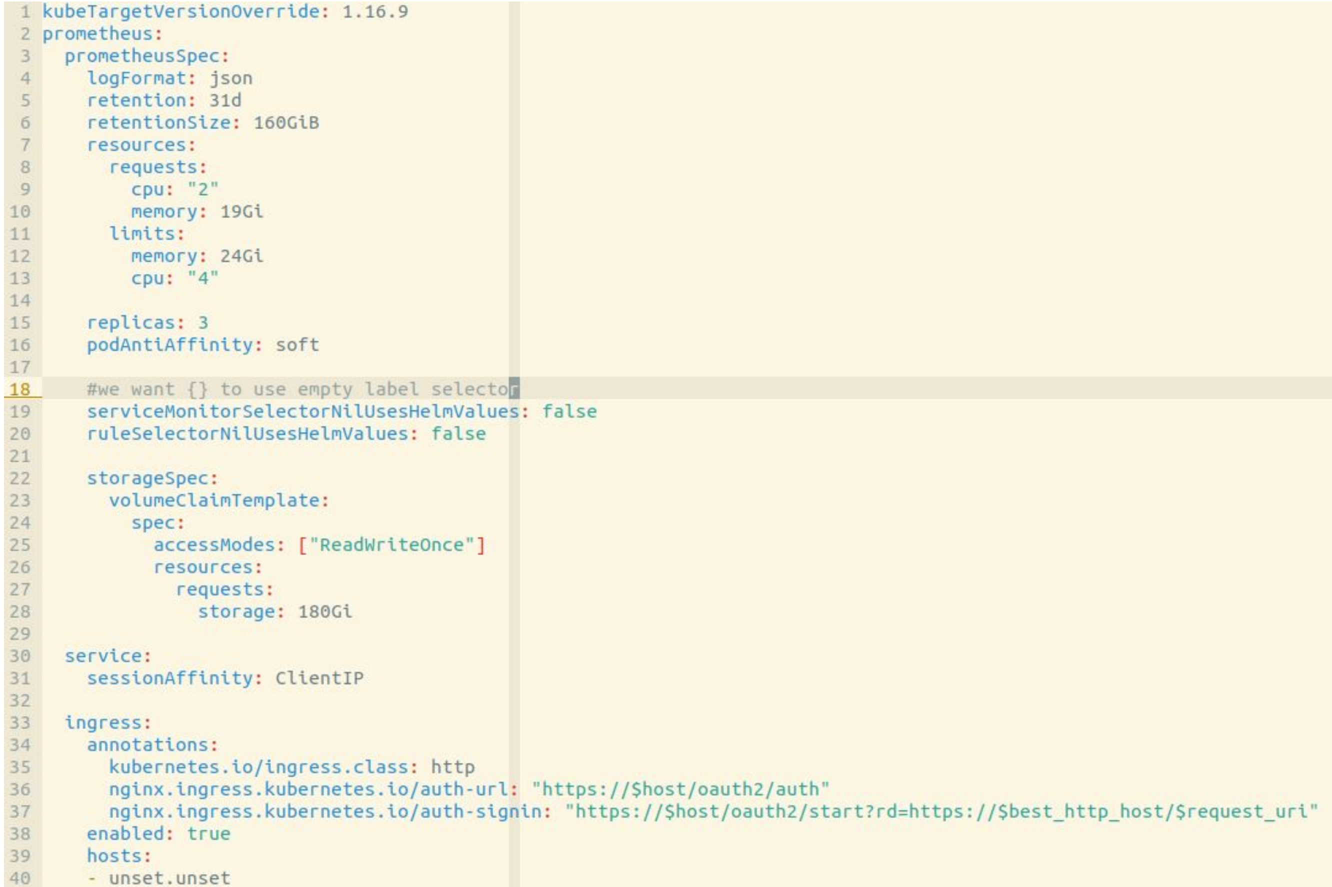
Metric	TTL
alertmanager.data-platform.dev.k8.atcloud.io	5 week
prometheus.data-platform.dev.k8.atcloud.io	6 week
weave.core-system.dev.k8.atcloud.io	6 week
grafana.data-platform.dev.k8.atcloud.io	6 week
app.sauron-web-mobile.dev.k8.atcloud.io	6 week
app.sauron-web.dev.k8.atcloud.io	6 week

Monitoring - overview

https://github.com/kubernetes/ingress-nginx/tree/master/deploy/grafana/dashboards

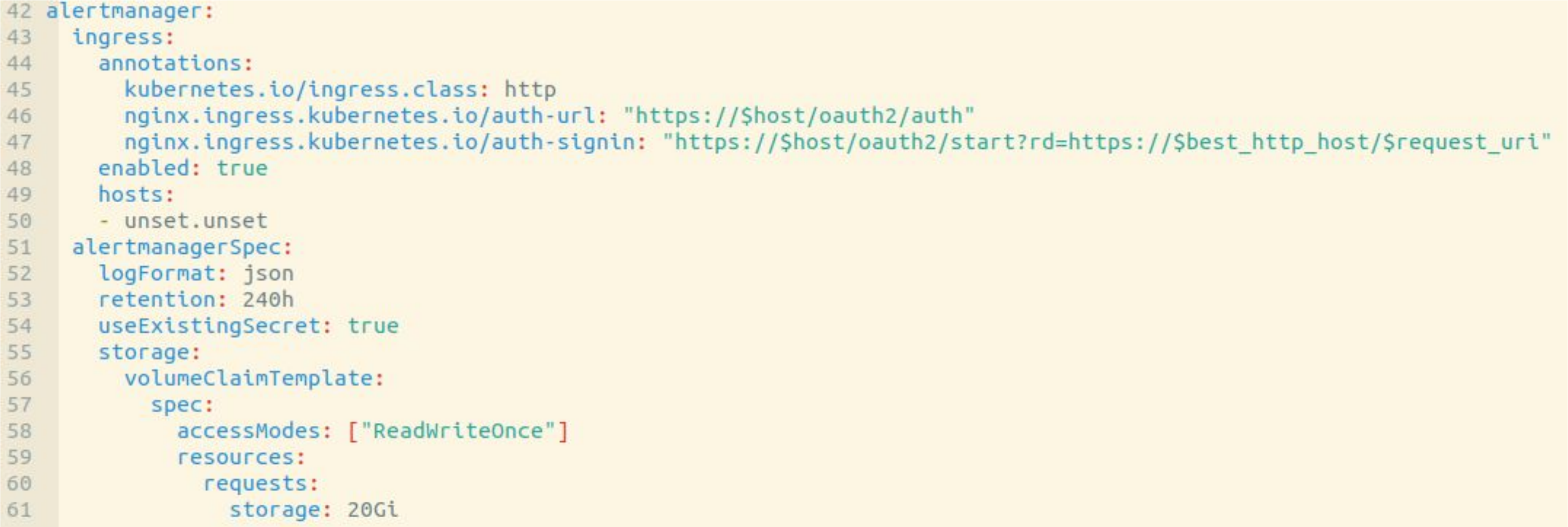


Monitoring - helm-values



```yaml
kubeTargetVersionOverride: 1.16.9
prometheus:
  prometheusSpec:
    logFormat: json
    retention: 31d
    retentionSize: 160GiB
    resources:
      requests:
        cpu: "2"
        memory: 19Gi
      limits:
        memory: 24Gi
        cpu: "4"

    replicas: 3
    podAntiAffinity: soft

    #we want {} to use empty label selector
    serviceMonitorSelectorNilUsesHelmValues: false
    ruleSelectorNilUsesHelmValues: false

    storageSpec:
      volumeClaimTemplate:
        spec:
          accessModes: ["ReadWriteOnce"]
          resources:
            requests:
              storage: 180Gi

  service:
    sessionAffinity: ClientIP

  ingress:
    annotations:
      kubernetes.io/ingress.class: http
      nginx.ingress.kubernetes.io/auth-url: "https://$host/oauth2/auth"
      nginx.ingress.kubernetes.io/auth-signin: "https://$host/oauth2/start?rd=https://$best_http_host/$request_uri"
    enabled: true
    hosts:
    - unset.unset
```

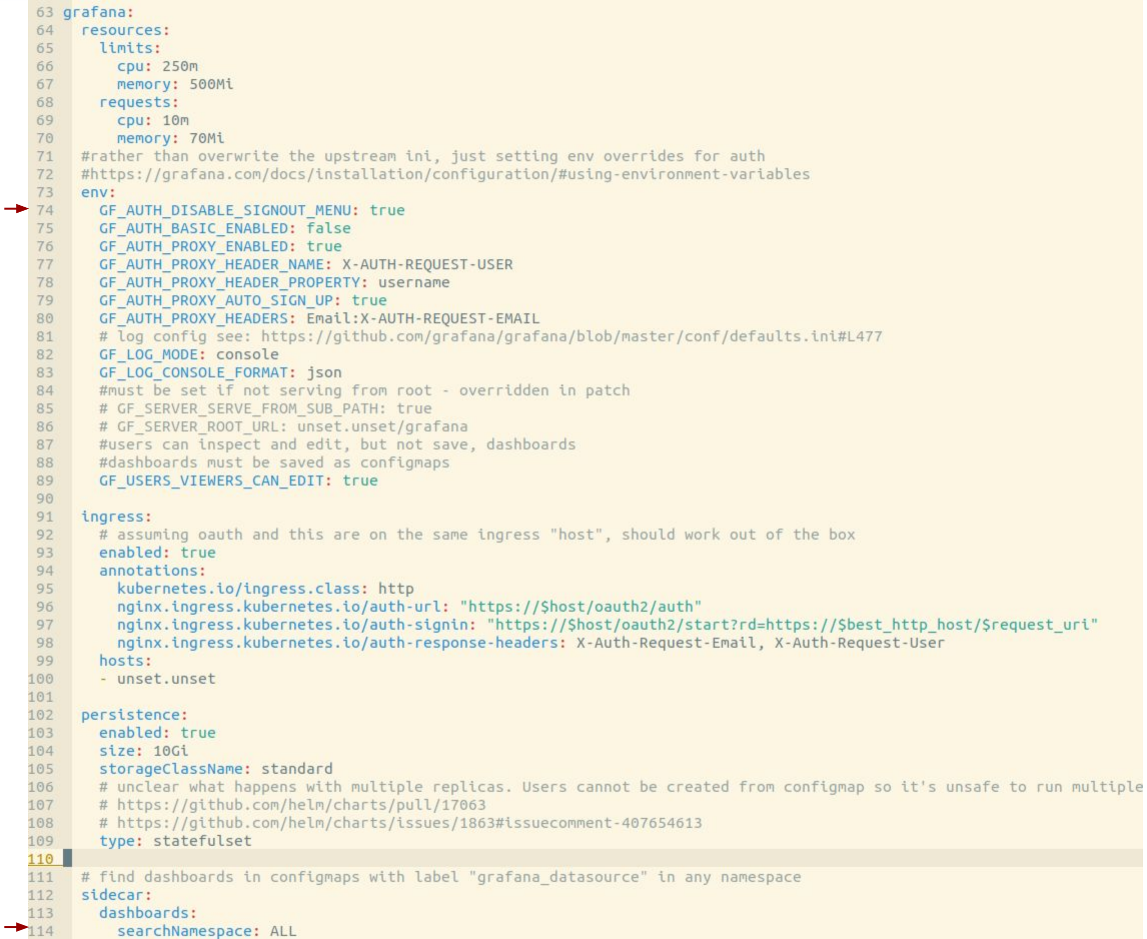
Monitoring - helm-values



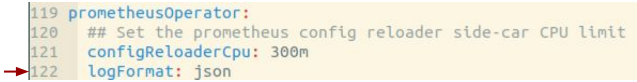
```yaml
alertmanager:
  ingress:
    annotations:
      kubernetes.io/ingress.class: http
      nginx.ingress.kubernetes.io/auth-url: "https://$host/oauth2/auth"
      nginx.ingress.kubernetes.io/auth-signin: "https://$host/oauth2/start?rd=https://$best_http_host/$request_uri"
    enabled: true
    hosts:
    - unset.unset
  alertmanagerSpec:
    logFormat: json
    retention: 240h
    useExistingSecret: true
    storage:
      volumeClaimTemplate:
        spec:
          accessModes: ["ReadWriteOnce"]
          resources:
            requests:
              storage: 20Gi
```

github.com/afirth/kceu2020

Monitoring - helm-values

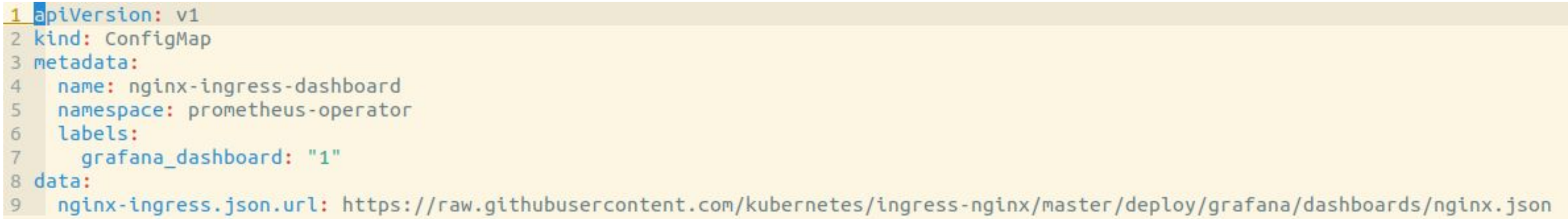


```yaml
grafana:
  resources:
    limits:
      cpu: 250m
      memory: 500Mi
    requests:
      cpu: 10m
      memory: 70Mi
  #rather than overwrite the upstream ini, just setting env overrides for auth
  #https://grafana.com/docs/installation/configuration/#using-environment-variables
  env:
    GF_AUTH_DISABLE_SIGNOUT_MENU: true
    GF_AUTH_BASIC_ENABLED: false
    GF_AUTH_PROXY_ENABLED: true
    GF_AUTH_PROXY_HEADER_NAME: X-AUTH-REQUEST-USER
    GF_AUTH_PROXY_HEADER_PROPERTY: username
    GF_AUTH_PROXY_AUTO_SIGN_UP: true
    GF_AUTH_PROXY_HEADERS: Email:X-AUTH-REQUEST-EMAIL
    # log config see: https://github.com/grafana/grafana/blob/master/conf/defaults.ini#L477
    GF_LOG_MODE: console
    GF_LOG_CONSOLE_FORMAT: json
    #must be set if not serving from root - overridden in patch
    # GF_SERVER_SERVE_FROM_SUB_PATH: true
    # GF_SERVER_ROOT_URL: unset.unset/grafana
    #users can inspect and edit, but not save, dashboards
    #dashboards must be saved as configmaps
    GF_USERS_VIEWERS_CAN_EDIT: true

  ingress:
    # assuming oauth and this are on the same ingress "host", should work out of the box
    enabled: true
    annotations:
      kubernetes.io/ingress.class: http
      nginx.ingress.kubernetes.io/auth-url: "https://$host/oauth2/auth"
      nginx.ingress.kubernetes.io/auth-signin: "https://$host/oauth2/start?rd=https://$best_http_host/$request_uri"
      nginx.ingress.kubernetes.io/auth-response-headers: X-Auth-Request-Email, X-Auth-Request-User
    hosts:
    - unset.unset

  persistence:
    enabled: true
    size: 10Gi
    storageClassName: standard
    # unclear what happens with multiple replicas. Users cannot be created from configmap so it's unsafe to run multiple
    # https://github.com/helm/charts/pull/17063
    # https://github.com/helm/charts/issues/1863#issuecomment-407654613
    type: statefulset

  # find dashboards in configmaps with label "grafana_datasource" in any namespace
  sidecar:
    dashboards:
      searchNamespace: ALL
```



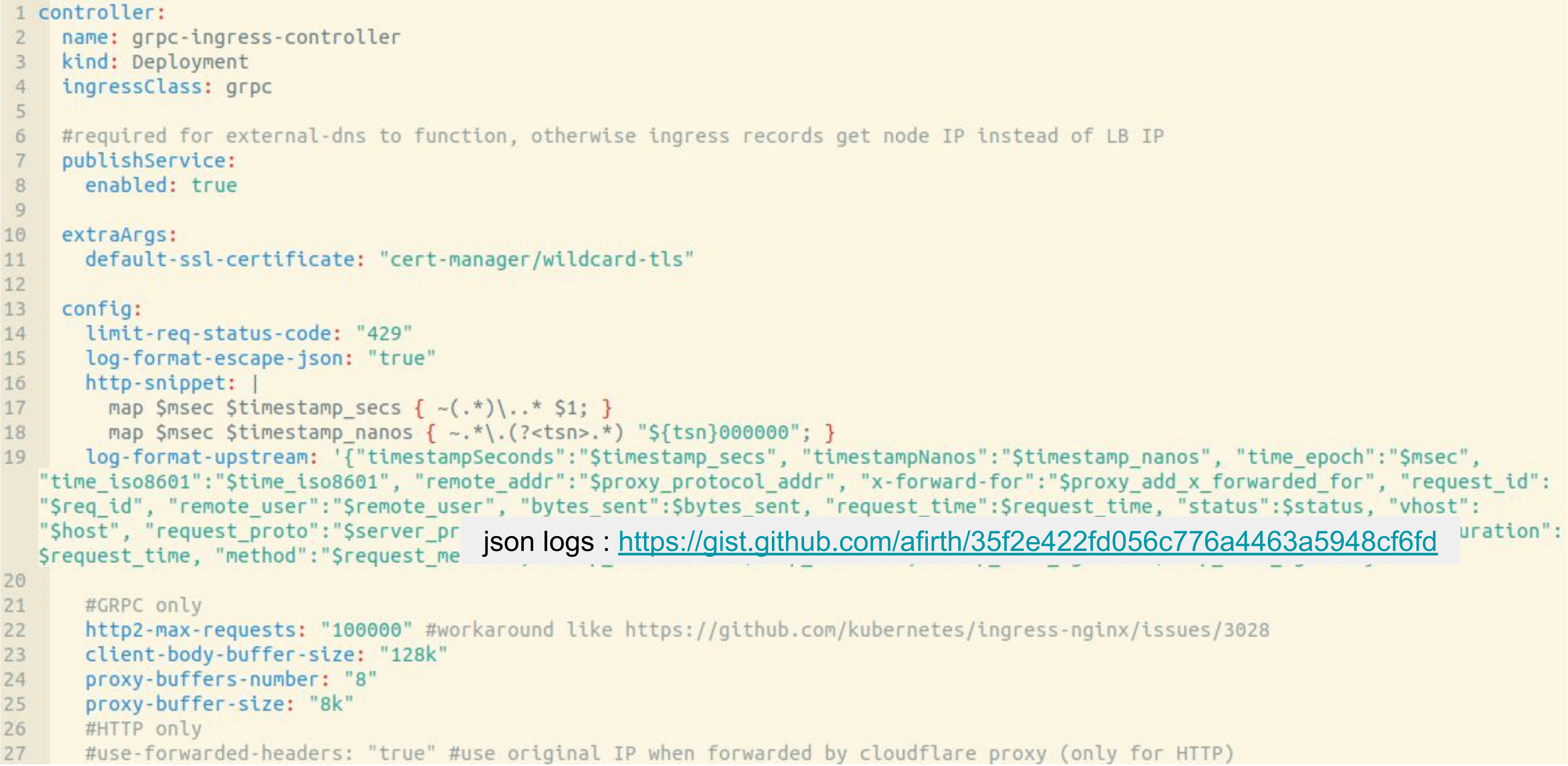
```yaml
prometheusOperator:
  ## Set the prometheus config reloader side-car CPU limit
  configReloaderCpu: 300m
  logFormat: json
```

Monitoring - dashboards



```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: nginx-ingress-dashboard
  namespace: prometheus-operator
  labels:
    grafana_dashboard: "1"
data:
  nginx-ingress.json.url: https://raw.githubusercontent.com/kubernetes/ingress-nginx/master/deploy/grafana/dashboards/nginx.json
```

NGINX - helm-values

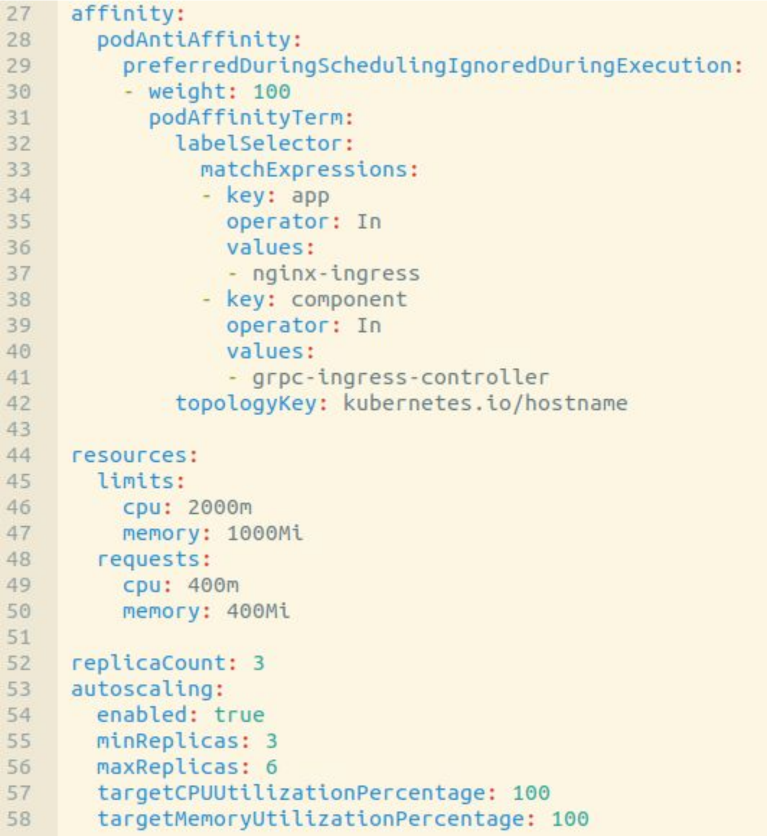


```yaml
controller:
  name: grpc-ingress-controller
  kind: Deployment
  ingressClass: grpc

  #required for external-dns to function, otherwise ingress records get node IP instead of LB IP
  publishService:
    enabled: true

  extraArgs:
    default-ssl-certificate: "cert-manager/wildcard-tls"

  config:
    limit-req-status-code: "429"
    log-format-escape-json: "true"
    http-snippet: |
      map $msec $timestamp_secs { ~(.*)\..* $1; }
      map $msec $timestamp_nanos { ~.*\.(?<tsn>.*) "${tsn}000000"; }
    log-format-upstream: '{"timestampSeconds":"$timestamp_secs", "timestampNanos":"$timestamp_nanos", "time_epoch":"$msec",
"time_iso8601":"$time_iso8601", "remote_addr":"$proxy_protocol_addr", "x-forward-for":"$proxy_add_x_forwarded_for", "request_id":
"$req_id", "remote_user":"$remote_user", "bytes_sent":$bytes_sent, "request_time":$request_time, "status":$status, "vhost":
"$host", "request_proto":"$server_pr                                                                                    uration":
$request_time, "method":"$request_me

    #GRPC only
    http2-max-requests: "100000" #workaround like https://github.com/kubernetes/ingress-nginx/issues/3028
    client-body-buffer-size: "128k"
    proxy-buffers-number: "8"
    proxy-buffer-size: "8k"
    #HTTP only
    #use-forwarded-headers: "true" #use original IP when forwarded by cloudflare proxy (only for HTTP)
```

json logs : https://gist.github.com/afirth/35f2e422fd056c776a4463a5948cf6fd

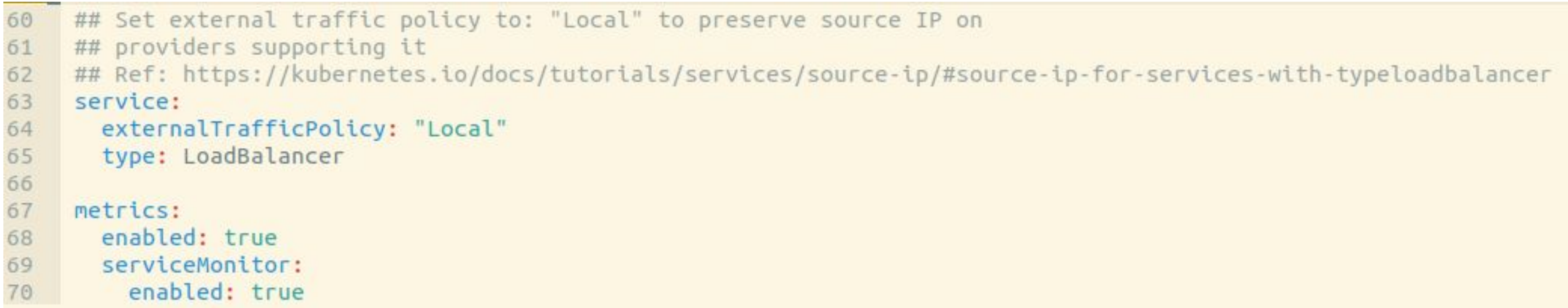
github.com/afirth/kceu2020

NGINX - helm-values



```yaml
affinity:
  podAntiAffinity:
    preferredDuringSchedulingIgnoredDuringExecution:
    - weight: 100
      podAffinityTerm:
        labelSelector:
          matchExpressions:
          - key: app
            operator: In
            values:
            - nginx-ingress
          - key: component
            operator: In
            values:
            - grpc-ingress-controller
        topologyKey: kubernetes.io/hostname

resources:
  limits:
    cpu: 2000m
    memory: 1000Mi
  requests:
    cpu: 400m
    memory: 400Mi

replicaCount: 3
autoscaling:
  enabled: true
  minReplicas: 3
  maxReplicas: 6
  targetCPUUtilizationPercentage: 100
  targetMemoryUtilizationPercentage: 100
```

NGINX - helm-values



```yaml
## Set external traffic policy to: "Local" to preserve source IP on
## providers supporting it
## Ref: https://kubernetes.io/docs/tutorials/services/source-ip/#source-ip-for-services-with-typeloadbalancer
service:
  externalTrafficPolicy: "Local"
  type: LoadBalancer

metrics:
  enabled: true
  serviceMonitor:
    enabled: true
```

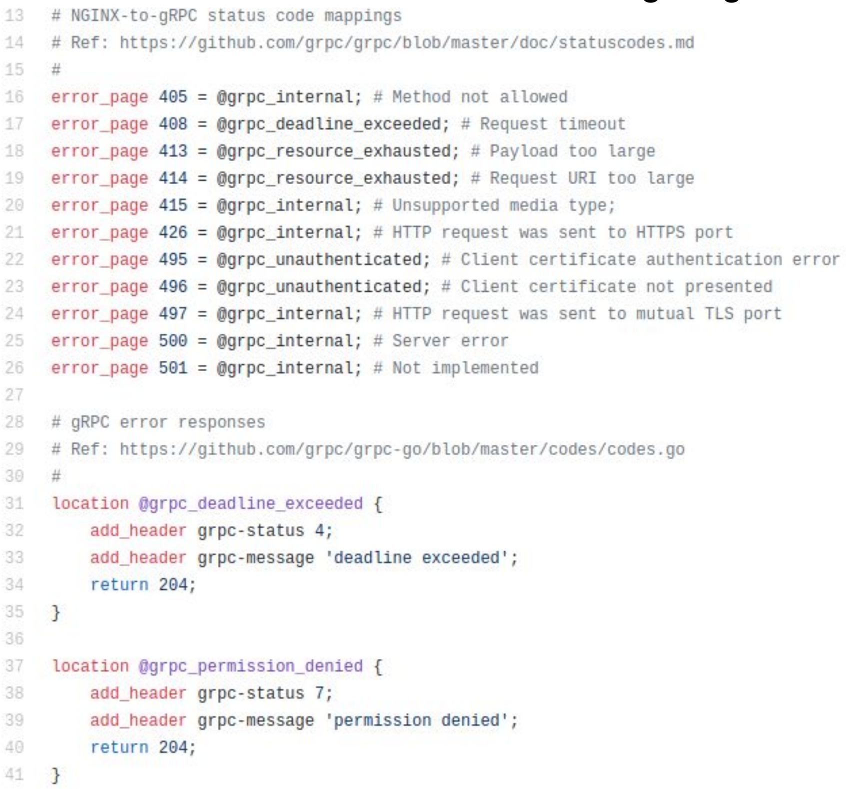
NGINX - patches

- Horizontal Pod Autoscaler
- Resources
- Maybe configmap tweaks

NGINX - gRPC specific

https://gist.github.com/nginx-gists/87ed942d4ee9f7e7ebb2ccf757ed90be#file-errors-grpc_conf-L4

nginx-gists/errors.grpc_conf

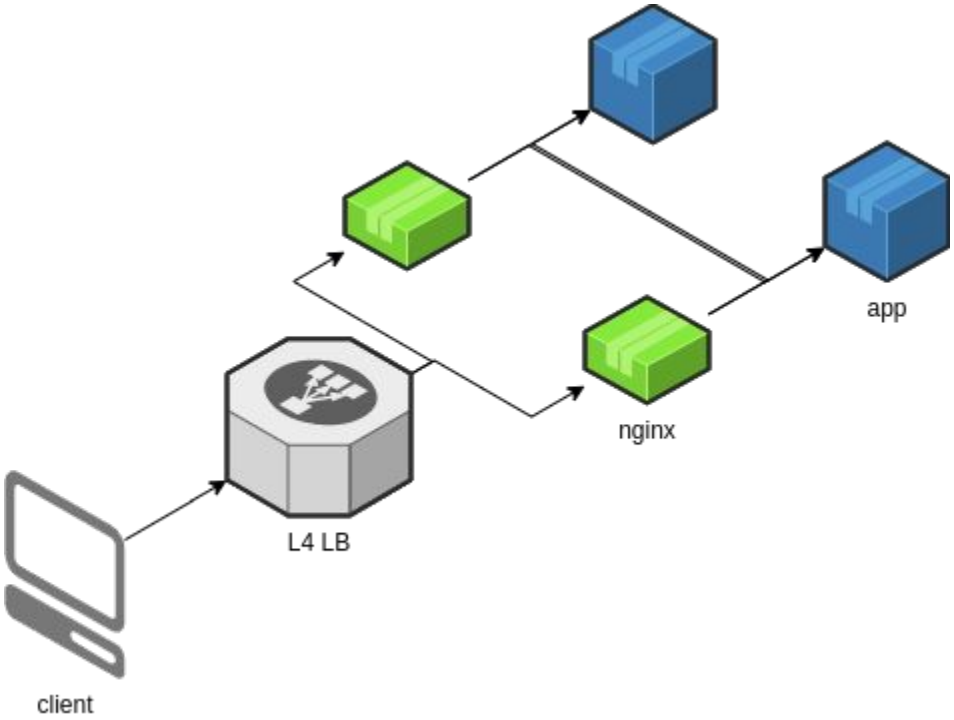


```
# NGINX-to-gRPC status code mappings
# Ref: https://github.com/grpc/grpc/blob/master/doc/statuscodes.md
#
error_page 405 = @grpc_internal; # Method not allowed
error_page 408 = @grpc_deadline_exceeded; # Request timeout
error_page 413 = @grpc_resource_exhausted; # Payload too large
error_page 414 = @grpc_resource_exhausted; # Request URI too large
error_page 415 = @grpc_internal; # Unsupported media type;
error_page 426 = @grpc_internal; # HTTP request was sent to HTTPS port
error_page 495 = @grpc_unauthenticated; # Client certificate authentication error
error_page 496 = @grpc_unauthenticated; # Client certificate not presented
error_page 497 = @grpc_internal; # HTTP request was sent to mutual TLS port
error_page 500 = @grpc_internal; # Server error
error_page 501 = @grpc_internal; # Not implemented

# gRPC error responses
# Ref: https://github.com/grpc/grpc-go/blob/master/codes/codes.go
#
location @grpc_deadline_exceeded {
    add_header grpc-status 4;
    add_header grpc-message 'deadline exceeded';
    return 204;
}

location @grpc_permission_denied {
    add_header grpc-status 7;
    add_header grpc-message 'permission denied';
    return 204;
}
```

NGINX - gRPC specific

- ++ Fast, stable, widespread
- -- Long lived connections
- -- HTTP2 error codes
- -- Routing on anything besides hostname
- -- Request level load balancing

Recap



++

- DNS records
- TLS certs
- Monitoring
- AuthN/AuthZ
- Multiple K8s
- Multiple apps
- Multiple stages
- Multiple nginxs