链滴

使用 flagger 实现 automated canary analysis

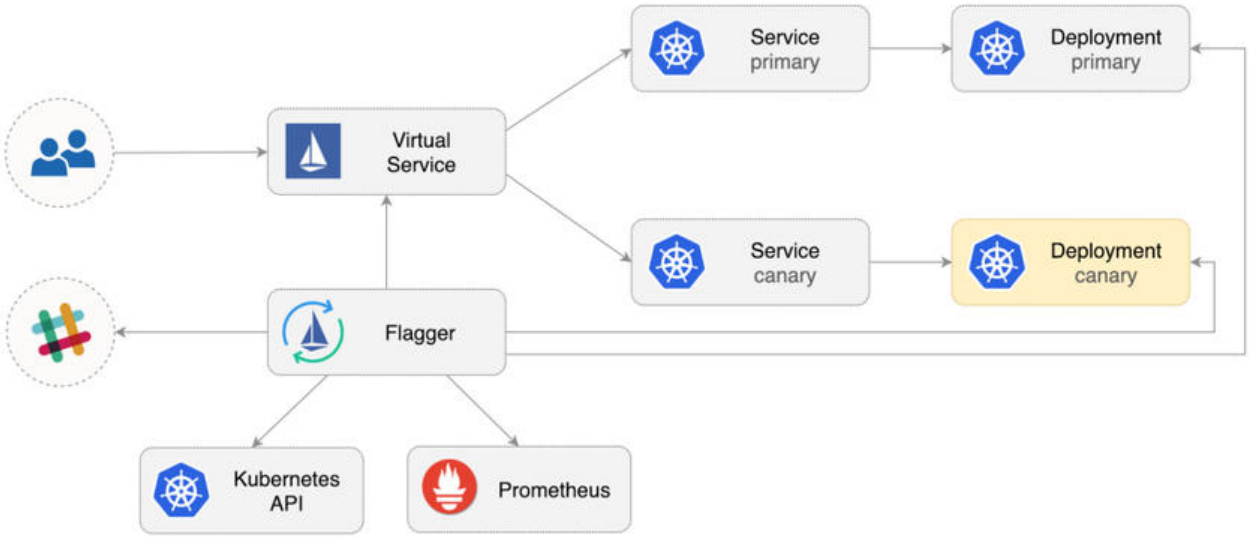
作者：fish2018

原文链接：https://ld246.com/article/1574058151661

来源网站：链滴

项目地址：https://github.com/weaveworks/flagger

flagger以prometheus的metrics作为依据，通过自动调整virtualservice的流量路由权重实现灰度发布



这里使用rancher部署，首先创建flagger名称空间，安装istio、kube-prometheus(prometheus operator)、helm

部署gateway资源

```
apiVersion: networking.istio.io/v1alpha3
kind: Gateway
metadata:
 name: flagger-gateway
 namespace: flagger
spec:
 selector:
 istio: ingressgateway
 servers:
 - hosts:
 - '*'
 port:
 name: http
 number: 80
 protocol: HTTP
```

部署flagger

添加charts仓库

```
helm repo add flagger https://flagger.app
```

部署flagger，指定istio，指定prometheus

```
helm upgrade -i flagger flagger/flagger \
--namespace=cattle-prometheus-p-2p8nx \ # prometheus所在的namespace
```

原文链接：使用 flagger 实现 automated canary analysis

```
--set crd.create=true \
--set meshProvider=istio \
--set metricsServer=http://prometheus-operated:9090
```

podinfo的deployment、hpa资源

kubectl apply -f podinfo.yml --namespace=flagger

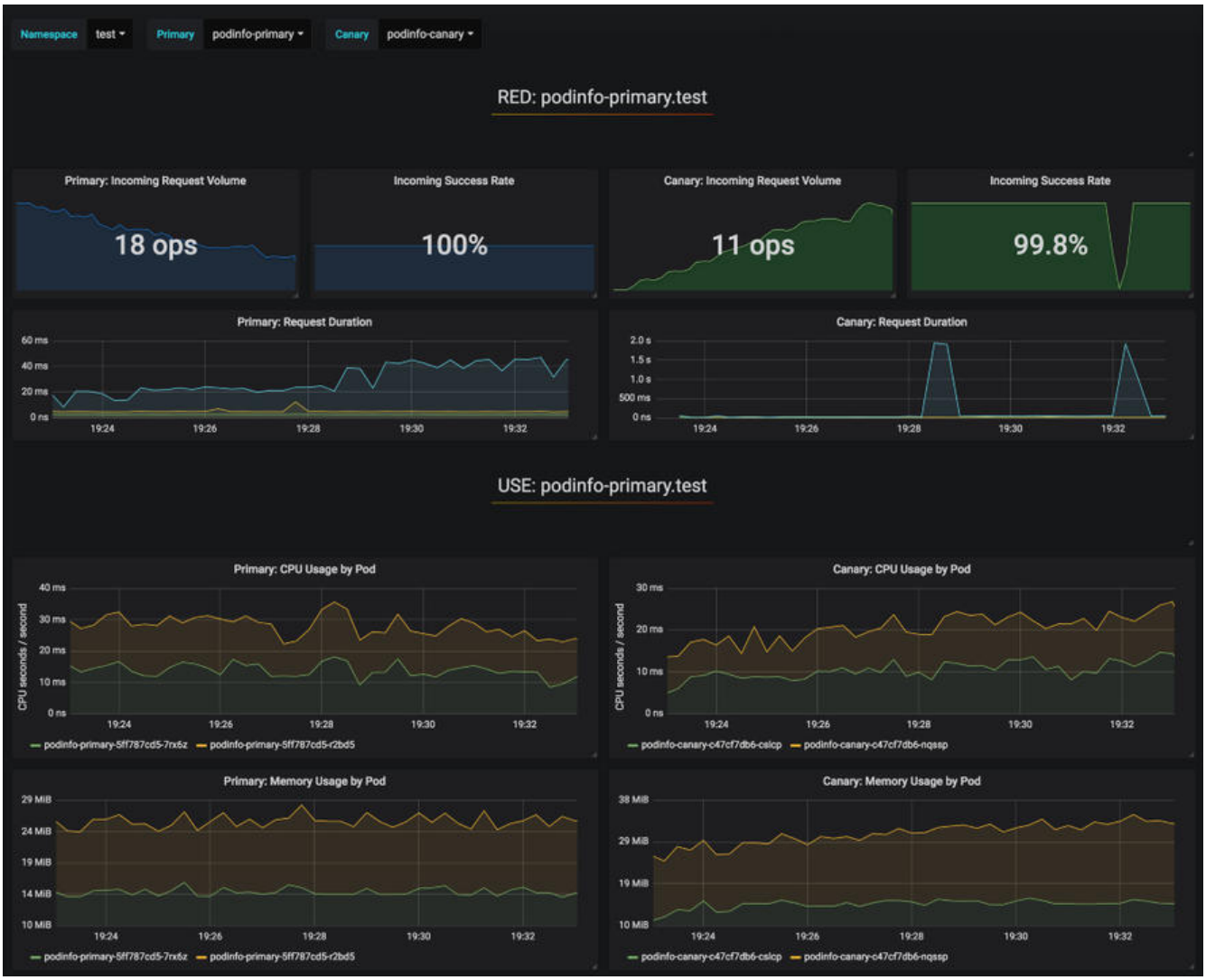
podinfo.yml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: podinfo
  labels:
    app: podinfo
spec:
  minReadySeconds: 5
  revisionHistoryLimit: 5
  progressDeadlineSeconds: 60
  strategy:
    rollingUpdate:
      maxUnavailable: 1
    type: RollingUpdate
  selector:
    matchLabels:
      app: podinfo
  template:
    metadata:
      annotations:
        prometheus.io/scrape: "true"
        prometheus.io/port: "9797"
      labels:
        app: podinfo
    spec:
      containers:
      - name: podinfod
        image: stefanprodan/podinfo:3.1.0
        imagePullPolicy: IfNotPresent
        ports:
        - name: http
          containerPort: 9898
          protocol: TCP
        - name: http-metrics
          containerPort: 9797
          protocol: TCP
        - name: grpc
          containerPort: 9999
          protocol: TCP
        command:
        - ./podinfo
        - --port=9898
        - --port-metrics=9797
        - --grpc-port=9999
```

```
        - --grpc-service-name=podinfo
        - --level=info
        - --random-delay=false
        - --random-error=false
        livenessProbe:
          exec:
            command:
            - podcli
            - check
            - http
            - localhost:9898/healthz
          initialDelaySeconds: 5
          timeoutSeconds: 5
        readinessProbe:
          exec:
            command:
            - podcli
            - check
            - http
            - localhost:9898/readyz
          initialDelaySeconds: 5
          timeoutSeconds: 5
        resources:
          limits:
            cpu: 2000m
            memory: 512Mi
          requests:
            cpu: 100m
            memory: 64Mi
---
apiVersion: autoscaling/v2beta1
kind: HorizontalPodAutoscaler
metadata:
  name: podinfo
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: podinfo
  minReplicas: 2
  maxReplicas: 4
  metrics:
  - type: Resource
    resource:
      name: cpu
      # scale up if usage is above
      # 99% of the requested CPU (100m)
      targetAverageUtilization: 99
```

部署grafana

自带一个istio-canary面板，也可以导出面板模板导入其他grafana中




```
helm upgrade -i flagger-grafana flagger/grafana \
--namespace=cattle-prometheus-p-2p8nx \
--set url=http://prometheus-operated:9090
```

flagger-loadtester的deployment、service资源

```
kubectl apply -f flagger-loadtester.yml --namespace=flagger
```

flagger-loadtester.yml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: flagger-loadtester
  labels:
    app: flagger-loadtester
spec:
  selector:
    matchLabels:
      app: flagger-loadtester
  template:
    metadata:
```

原文链接：使用 flagger 实现 automated canary analysis

```
    labels:
      app: flagger-loadtester
    annotations:
      prometheus.io/scrape: "true"
      prometheus.io/port: "8080"
  spec:
    containers:
      - name: loadtester
        image: weaveworks/flagger-loadtester:0.11.0
        imagePullPolicy: IfNotPresent
        ports:
          - name: http
            containerPort: 8080
        command:
          - ./loadtester
          - -port=8080
          - -log-level=info
          - -timeout=1h
        livenessProbe:
          exec:
            command:
              - wget
              - --quiet
              - --tries=1
              - --timeout=4
              - --spider
              - http://localhost:8080/healthz
          timeoutSeconds: 5
        readinessProbe:
          exec:
            command:
              - wget
              - --quiet
              - --tries=1
              - --timeout=4
              - --spider
              - http://localhost:8080/healthz
          timeoutSeconds: 5
        resources:
          limits:
            memory: "512Mi"
            cpu: "1000m"
          requests:
            memory: "32Mi"
            cpu: "10m"
        securityContext:
          readOnlyRootFilesystem: true
          runAsUser: 10001
---
apiVersion: v1
kind: Service
metadata:
  name: flagger-loadtester
  labels:
```

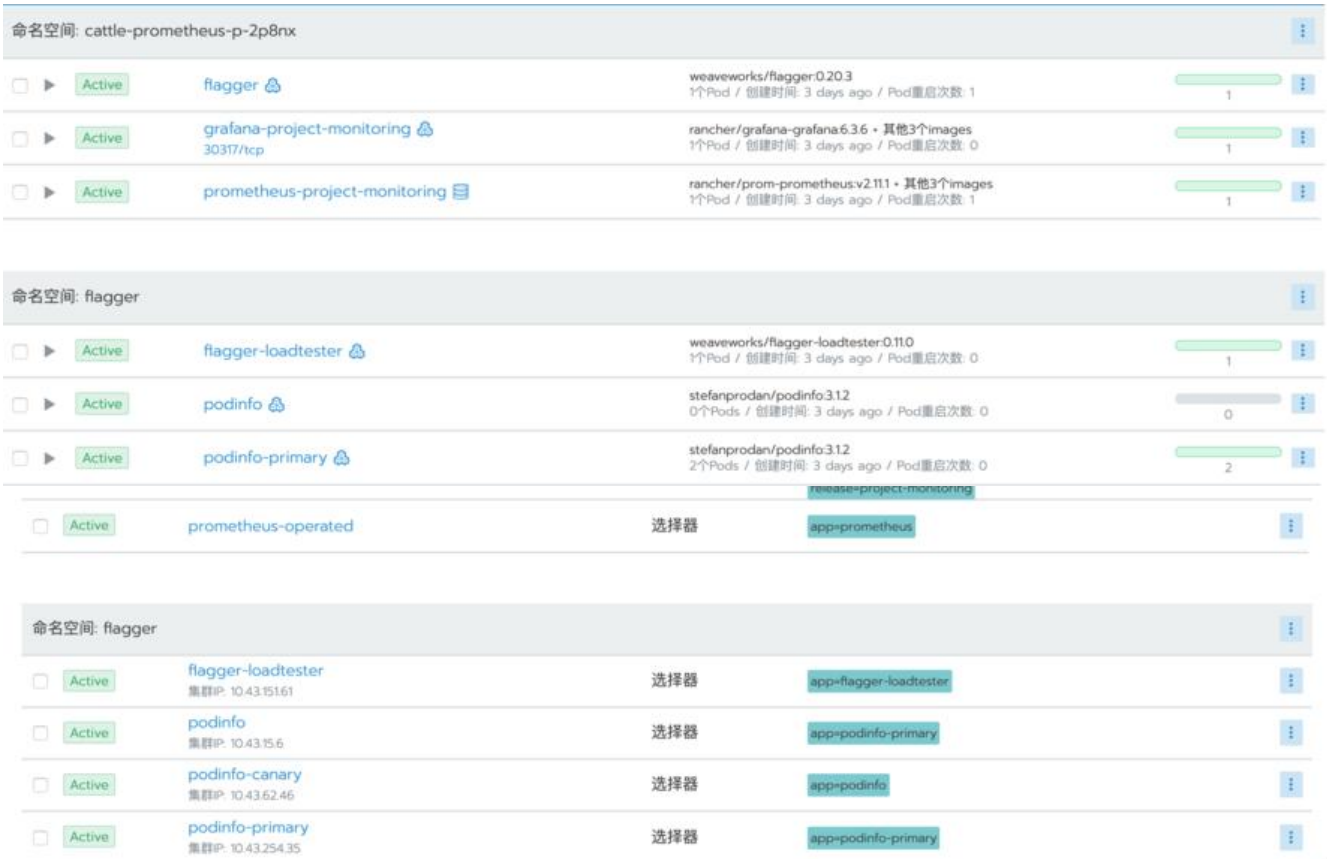
```
    app: flagger-loadtester
spec:
  type: ClusterIP
  selector:
    app: flagger-loadtester
  ports:
    - name: http
      port: 80
      protocol: TCP
      targetPort: http
```

podinfo的canary资源

rancher中istio的ingressgateway默认http2端口31380，在slb添加转发80=>31380

在hosts中添加app.istio.example.com映射slb的解析

canary资源创建podinfo-primary的deployment和service，podinfo-canary的service，istio的destinationrules、virtualservice等



kubectl apply -f podinfo-canary.yml --namespace=flagger

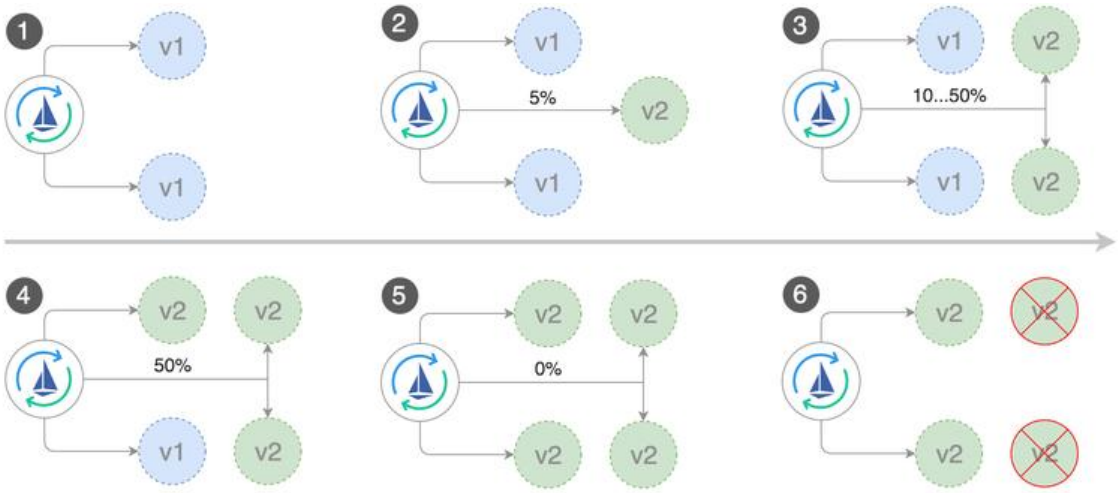
podinfo-canary.yml

```
apiVersion: flagger.app/v1alpha3
kind: Canary
metadata:
  name: podinfo
  namespace: flagger
spec:
```

```
# deployment reference
targetRef:
  apiVersion: apps/v1
  kind: Deployment
  name: podinfo
# the maximum time in seconds for the canary deployment
# to make progress before it is rollback (default 600s)
progressDeadlineSeconds: 60
# HPA reference (optional)
autoscalerRef:
  apiVersion: autoscaling/v2beta1
  kind: HorizontalPodAutoscaler
  name: podinfo
service:
  # service port number
  port: 9898
  # container port number or name (optional)
  targetPort: 9898
  # Istio gateways (optional)
  gateways:
  - mesh
  - flagger-gateway
  # Istio virtual service host names (optional)
  hosts:
  - podinfo.flagger
  - app.istio.example.com
  # Istio traffic policy (optional)
  trafficPolicy:
    tls:
      # use ISTIO_MUTUAL when mTLS is enabled
      mode: DISABLE
  # Istio retry policy (optional)
  retries:
    attempts: 3
    perTryTimeout: 1s
    retryOn: "gateway-error,connect-failure,refused-stream"
canaryAnalysis:
  # schedule interval (default 60s)
  interval: 1m
  # max number of failed metric checks before rollback
  threshold: 5
  # max traffic percentage routed to canary
  # percentage (0-100)
  maxWeight: 50
  # canary increment step
  # percentage (0-100)
  stepWeight: 10
  metrics:
  - name: istio_requests_total
    # minimum req success rate (non 5xx responses)
    # percentage (0-100)
    threshold: 99
    interval: 30s
  - name: istio_request_duration_seconds_bucket
```

```
    # maximum req duration P99
    # milliseconds
    threshold: 500
    interval: 30s
  # testing (optional)
  webhooks:
    - name: acceptance-test
      type: pre-rollout
      url: http://flagger-loadtester.flagger/
      timeout: 30s
      metadata:
        type: bash
        cmd: "curl -sd 'test' http://podinfo-canary:9898/token | grep token"
    - name: load-test
      url: http://flagger-loadtester.flagger/
      timeout: 5s
      metadata:
        cmd: "hey -z 1m -q 10 -c 2 http://podinfo-canary.flagger:9898/"
```

金丝雀发布



部署新版本镜像

```
kubectl -n book set image deployment/podinfo \
podinfod=stefanprodan/podinfo:3.1.1
```

查看变化

```
watch 'kubectl -n flagger describe canary/podinfo | tail -n 5'
```

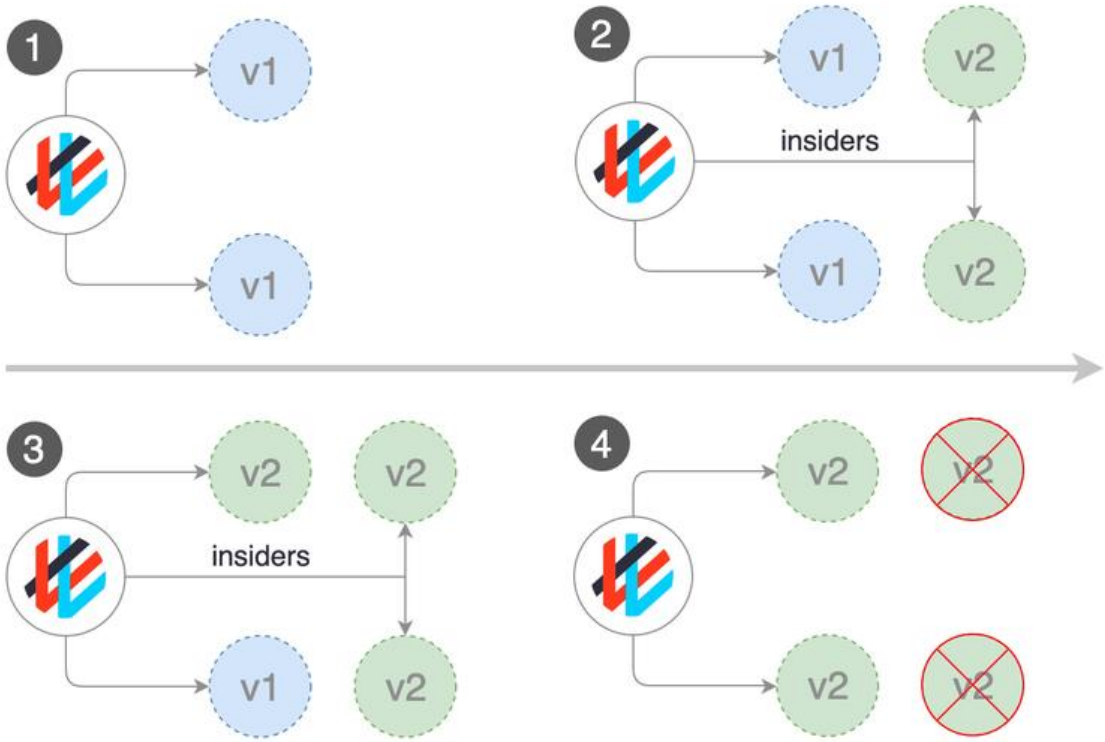
```
Events:

New revision detected podinfo.flagger
Scaling up podinfo.flagger
Waiting for podinfo.flagger rollout to finish: 0 of 1 updated replicas are available
Advance podinfo.flagger canary weight 5
Advance podinfo.flagger canary weight 10
Advance podinfo.flagger canary weight 15
Advance podinfo.flagger canary weight 20
```

原文链接：使用 flagger 实现 automated canary analysis

```
Advance podinfo.flagger canary weight 25
Advance podinfo.flagger canary weight 30
Advance podinfo.flagger canary weight 35
Advance podinfo.flagger canary weight 40
Advance podinfo.flagger canary weight 45
Advance podinfo.flagger canary weight 50
Copying podinfo.flagger template spec to podinfo-primary.flagger
Waiting for podinfo-primary.flagger rollout to finish: 1 of 2 updated replicas are available
Promotion completed! Scaling down podinfo.flagger
```

A/B测试

通过header匹配来路由流量




```
apiVersion: flagger.app/v1alpha3
kind: Canary
metadata:
  name: podinfo
  namespace: flagger
spec:
  # deployment reference
  targetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: podinfo
  # the maximum time in seconds for the canary deployment
  # to make progress before it is rollback (default 600s)
  progressDeadlineSeconds: 60
  # HPA reference (optional)
  autoscalerRef:
    apiVersion: autoscaling/v2beta1
```

原文链接：使用 flagger 实现 automated canary analysis

```
    kind: HorizontalPodAutoscaler
    name: podinfo
  service:
    # container port
    port: 9898
    # Istio gateways (optional)
    gateways:
    - flagger-gateway
    # Istio virtual service host names (optional)
    hosts:
    - app.istio.example.com
    # Istio traffic policy (optional)
    trafficPolicy:
      tls:
        # use ISTIO_MUTUAL when mTLS is enabled
        mode: DISABLE
  canaryAnalysis:
    # schedule interval (default 60s)
    interval: 1m
    # total number of iterations
    iterations: 10
    # max number of failed iterations before rollback
    threshold: 2
    # canary match condition
    match:
      - headers:
          user-agent:
            regex: "^(?!.*Chrome).*Safari.*"
      - headers:
          cookie:
            regex: "^(.*?;)?(type=insider)(;.*)?$"
    metrics:
    - name: request-success-rate
      # minimum req success rate (non 5xx responses)
      # percentage (0-100)
      threshold: 99
      interval: 1m
    - name: request-duration
      # maximum req duration P99
      # milliseconds
      threshold: 500
      interval: 30s
    # generate traffic during analysis
    webhooks:
      - name: load-test
        url: http://flagger-loadtester.flagger/
        timeout: 5s
        metadata:
          cmd: "hey -z 1m -q 10 -c 2 -H 'Cookie: type=insider' http://podinfo.flagger:9898/"
```

自动回滚

在金丝雀分析期间，可以生成HTTP 500错误和高响应延迟，以测试Flagger是否暂停升级。

在loadtest中执行命令，生成HTTP 500错误返回：

```
watch curl http://podinfo-canary:9898/status/500
```

生成延迟

```
watch curl http://podinfo-canary:9898/delay/1
```

当失败检查的数量达到金丝雀分析阈值时，流量被路由回主版本，金丝雀版本被缩放为0，并且升级标记为失败。

金丝雀报错和延迟峰值被记录为Kubernetes事件

```
kubectl -n cattle-prometheus-p-2p8nx logs deployment/flagger -f | jq .msg
```

```
Starting canary deployment for podinfo.flagger
Advance podinfo.flagger canary weight 5
Advance podinfo.flagger canary weight 10
Advance podinfo.flagger canary weight 15
Halt podinfo.flagger advancement success rate 69.17% < 99%
Halt podinfo.flagger advancement success rate 61.39% < 99%
Halt podinfo.flagger advancement success rate 55.06% < 99%
Halt podinfo.flagger advancement success rate 47.00% < 99%
Halt podinfo.flagger advancement success rate 37.00% < 99%
Halt podinfo.flagger advancement request duration 1.515s > 500ms
Halt podinfo.flagger advancement request duration 1.600s > 500ms
Halt podinfo.flagger advancement request duration 1.915s > 500ms
Halt podinfo.flagger advancement request duration 2.050s > 500ms
Halt podinfo.flagger advancement request duration 2.515s > 500ms
Rolling back podinfo.flagger failed checks threshold reached 10
Canary failed! Scaling down podinfo.flagger
```