一、启动flask 。提供metrics 页面（监控类型程序实现，内容为标红字段） push_metrics.py

```
#------------------
#!/usr/bin/env python
# coding: utf-8
import commands
import prometheus_client
from flask import Response, Flask
from prometheus_client import Gauge
from prometheus_client.core import CollectorRegistry
import os

app = Flask(__name__)

registry = CollectorRegistry()

kafka_ok = Gauge("kafka_ok", "kafka_ok", registry=registry)
kafka_notok = Gauge("kafka_not_ok", "kafka_not_ok", registry=registry)
uic_ok = Gauge("uic_ok", "uic_ok", registry=registry)
uic_notok = Gauge("uic_not_ok", "uic_not_ok", registry=registry)
notice_ok = Gauge("notice_ok", "notice_ok", registry=registry)
notice_notok = Gauge("notice_not_ok", "notice_not_ok", registry=registry)

key_list = {
    "kafka_ok": "HLEN act:present:mobile_phone:add_vip_success",
    "kafka_notok": "GET act:present:mobile_phone:add_vip_failure",
    "uic_ok": "GET act:present:mobile_phone:update_uic_success",
    "uic_notok": "GET act:present:mobile_phone:update_uic_failure",
    "notice_ok": "GET act:present:mobile_phone:notice_success",
    "notice_notok": "GET act:present:mobile_phone:notice_failure",
}
```

```
def get_value(comm):
    command = "/app/scripts/redis-cli -h  主机ip -p 8080 -a 密码 -c %s" % comm
    #resp = commands.getoutput(command)
    a=os.popen(command)
    resp=a.read()
    if not resp:
        resp = 0
    return resp


@app.route("/metrics")
def requests_count():
    for name, key in key_list.items():
        globals().get(name).set(get_value(key))
    return Response(prometheus_client.generate_latest(registry),
                    mimetype="text/plain")


if __name__ == "__main__":
    app.run(host="0.0.0.0",port="8090")

#-----------------------
```

二、配置k8s ep（监控主要依赖ep） push_count.yaml

```
apiVersion: v1
kind: Endpoints
metadata:
  name: push-count
  labels:
    k8s-app: push-count   #用于关联serviceMonitor
  namespace: kube-system
subsets:
- addresses:
  - ip: python falsk服务IP：端口
    nodeName: push-count
```

```
  ports:
  - name: port
    port: 8090
```

三、配置serviceMonitor（selector ep 的matchLabels）push_count-serviceMonitor.yaml

```
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  name: push-count
  labels:
    k8s-app: push-count
  namespace: kube-system
spec:
  endpoints:
  - port: port
    interval: 5s   #采集频率
  jobLabel: k8s-app
  namespaceSelector:
    matchNames:
    - kube-system
  selector:
    matchLabels:
      k8s-app: push-count
```