Отчет по kubernetes и программы на python (подходит для любого ЯП)

1: Загрузить конфиг kubernetes политеха

```
mkdir ~/.kube
copy C:\Users\имя\kubeconfig.yaml ~/.kube\config
```

проверка:

```
kubectl get pods -A
```

```
NAMESPACE     NAME                                   READY   STATUS    RESTARTS   AGE
kube-system   coredns-668d6bf9bc-cz5pb               1/1     Running   0          9d
kube-system   coredns-668d6bf9bc-zs62t               1/1     Running   0          9d
kube-system   etcd-kube-master2                      1/1     Running   0          9d
kube-system   kube-apiserver-kube-master2            1/1     Running   0          9d
kube-system   kube-controller-manager-kube-master2   1/1     Running   0          9d
kube-system   kube-proxy-bqvl7                       1/1     Running   0          9d
kube-system   kube-proxy-dcs65                       1/1     Running   0          9d
kube-system   kube-proxy-jvbtf                       1/1     Running   0          9d
kube-system   kube-proxy-ph6pv                       1/1     Running   0          9d
kube-system   kube-scheduler-kube-master2            1/1     Running   0          9d
kube-system   weave-net-55n5j                        2/2     Running   0          9d
kube-system   weave-net-bwgtg                        2/2     Running   1          9d
kube-system   weave-net-pvfkn                        2/2     Running   0          9d
kube-system   weave-net-rq58j                        2/2     Running   0          9d
```

```
kubectl get nodes
```

```
NAME           STATUS   ROLES           AGE   VERSION
kube-master2   Ready    control-plane   9d    v1.32.3
kube-worker1   Ready    <none>          9d    v1.32.3
kube-worker2   Ready    <none>          9d    v1.32.3
kube-worker3   Ready    <none>          9d    v1.32.3
```

2:
Составить dockerfile и собрать образ из приложения python:

dockerfile:

```
FROM python:3.13.3-slim
RUN apt-get update && apt-get install -y --no-install-recommends \
```

```
    build-essential \
    libssl-dev \
    && rm -rf /var/lib/apt/lists/*

WORKDIR /app
COPY src/ /app/
COPY requirements.txt /app/
RUN pip install --no-cache-dir -r requirements.txt
CMD ["python", "main.py"]
```

Собираем:

```
docker build -t hs_python_opcua:latest
```

3:
Загрузка образа в kubernetes:

через **Docker Registry** :
логин в Docker hub

```
docker login
```

Для приватных репозиториев:

```
kubectl create secret docker-registry regcred `
  --docker-username="DOCKER_HUB_USERNAME" `
  --docker-password="DOCKER_HUB_PASSWORD" `
  --docker-email="youremail@example.com"
```

пушим в репу докерХаба:

```
docker tag hs_python_opcua:latest login_dockerHub/hs_python_opcua:latest
docker push login_dockerHub/hs_python_opcua:latest
```

Манифест hs_python_deployment.yaml:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: plc-app
```

```yaml
spec:
  replicas: 1
  selector:
    matchLabels:
      app: plc-opcua
  template:
    metadata:
      labels:
        app: plc-opcua
    spec:
      imagePullSecrets:
        - name: regcred
      containers:
        - name: plc-opcua-container
          image: Name-docker-hub/hs_python_opcua:latest
          ports:
            - containerPort: 4840
```

4:
запуск:

```
kubectl apply -f plc-app-deployment.yaml
```

результат:

```
PS C:\Users\user\Desktop\hs_python_docker> kubectl get pods
NAME                          READY   STATUS    RESTARTS   AGE
plc-app-59dc8dcd64-g2vtt      1/1     Running   0          19s
```

происходит запуск упаковочной линии.

5: Взаимодействие с API Kubernetes

можно например использовать библиотеку `kubernetes-client` для Python

```
pip install kubernetes
```

Программа которая выводит доступные деплойменты и позволяют включать и отключать программу, посредством изменения количества реплик 0/1:

```python
from kubernetes import client, config
config.load_kube_config()
v1 = client.AppsV1Api()
deployments = v1.list_namespaced_deployment(namespace='default')
```

```
for deployment in deployments.items:
    print(f"Deployment: {deployment.metadata.name}")
def scale_deployment(deployment_name, replicas):
    body = {
        "spec": {
            "replicas": replicas
        }
    }
 v1.patch_namespaced_deployment(name=deployment_name, namespace="default",
body=body)
#scale_deployment("plc-app", 1)  # запуск программы
scale_deployment("plc-app", 0) #выключить программу
```