

k8s 创建新的容器环境

前置知识

如果之前没有弄过 k8s 会有权限报错 点击这个链接配置一下权限 => [k8s 服务集群常用命令汇总](#)

常见 缩写 ns => namespace svc => service

单独起一个 pod 的配置文件 例如下面的配置文件叫做 mvp-frontend-nodejs.yaml

注意事项

1. 建议复制下面配置
2. 整体替换 mvp 三个字母 为你想要的服务
3. 非上线应用替换 frontend-nodejs 在第四行
4. 替换对外端口 在倒数第二行

```
1  apiVersion: v1
2  kind: Namespace
3  metadata:
4    name: frontend-nodejs # 非上线的pod 建议自己搞一个 ns
5  ---
6  apiVersion: apps/v1
7  kind: Deployment
8  metadata:
9    name: mvp-application # 这个名字自己修改
10 namespace: frontend-nodejs
11 spec:
12   replicas: 1
13   selector:
14     matchLabels:
15       app: mvp-node-application # 这个名字自己修改
16   template:
17     metadata:
18       labels:
19         app: mvp-node-application # 这个名字自己修改
20     spec:
21       imagePullSecrets:
22         - name: image-pull-secret # 这个 secret 需要自己配置一下 默认先不弄 看看能不能成功 如果成功了就先不用处理 不成功看评论
23       containers:
24         - name: mvp-node-application
```

```

25     image: registry.gitlab.dipeak.com/dipeak/generic-repository/deploy-
frontend:feat_public_mvp.v23.1106.c710b90f
26     ports:
27     - containerPort: 80
28     env:
29     - name: PM2_WATCH # docker启动的环境变量
30       value: "true"
31     - name: BACKEND_PROXY_URI
32       value: 'http://192.168.110.25:32290'
33     - name: DEVOPS_PROXY_URI
34       value: 'http://devops-backend-service.dipeak:8848'
35     - name: DITEST_PROXY_URI
36       value: 'http://192.168.110.27:18000'
37     resources:
38       limits:
39         cpu: "4"
40         memory: "8Gi"
41       requests:
42         cpu: "2"
43         memory: "4Gi"
44     nodeSelector:
45       env: master
46     tolerations:
47     - key: node-role.kubernetes.io/master
48       operator: Equal
49       effect: NoSchedule
50 ---
51 apiVersion: v1
52 kind: Service
53 metadata:
54   name: mvp-node-app-service # service 用来对外 处理服务到镜像内
55   namespace: frontend-nodejs
56 spec:
57   selector:
58     app: mvp-node-application # 对应 app 的名字和 上面进行对应
59   ports:
60   - protocol: TCP
61     port: 3000 # service port 不用动
62     targetPort: 80 # Application port 不用动
63     nodePort: 32336 # cluster port 这个是访问端口 访问示例
http://192.168.110.27:32336/xengine/engine 形式进行访问
64   type: NodePort

```

使用上面的配置

- 1 `kubectl apply -f mvp-frontend-nodejs.yaml`
- 2 `kubectl get pods -n frontend-nodejs -o wide` 查看你的 pod 的状态
- 3 `kubectl exec -it [pod名字 上面的指令会出现] bash -n frontend-nodejs`