

App 升级步骤

分布式、高可用升级集群监控功能操作

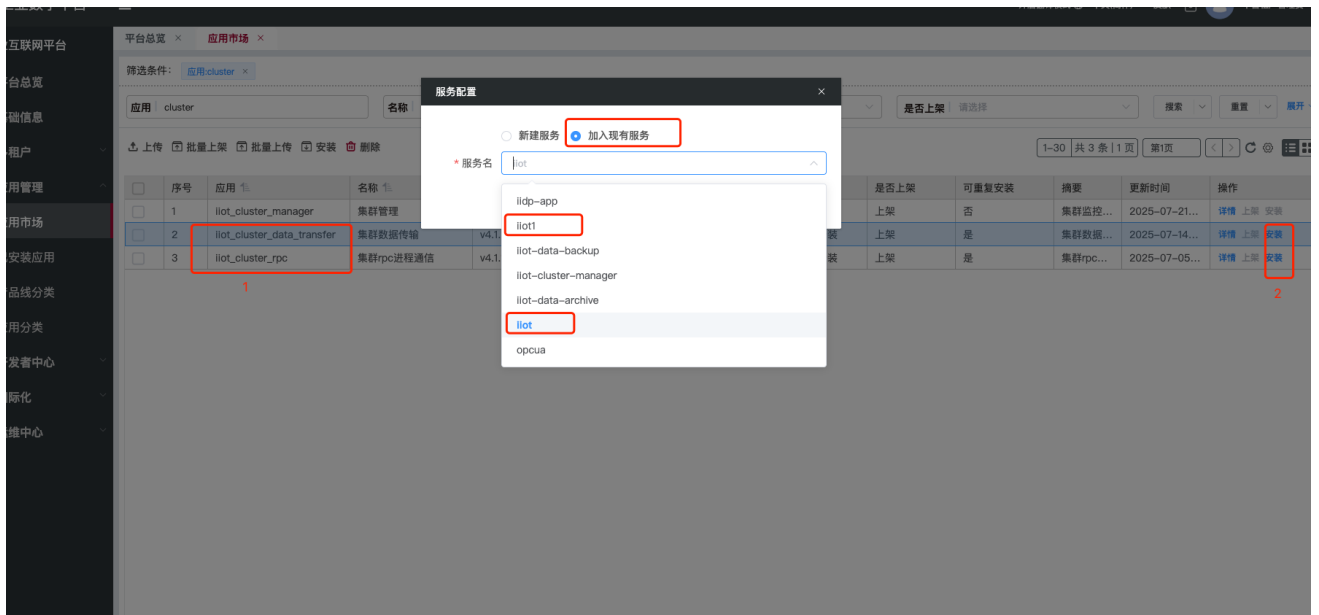
1. 应用服务

安装集群监控相关 3 个 App

聚合规则		iiot_aggregation	基础套餐包	标准	
行业模板库	-	iiot_industry_template_common		标准	
云边协同	工程协同	iiot_edge_project		标准	
	固件升级	iiot_edge_firmware_app、iiot_edge_sync		标准	
	驱动管理	iiot_edge_drive_app		标准	
数据调度	用户编程	iiot_program		标准	
	规则引擎	iiot_ruleengine		标准	
数据转发	协议配置	iiot_dataservice、iiot_dataservice_opcua (分布式部署, 需单独部署opcua容器)、iiot_dataservice_api、iiot_message-transfer、rabbitmq		标准	
	转发配置			标准	
	同步配置			标准	
	连接流管理			iiot_flow	标准
边缘节点	边缘管理	iiot_edge_node		标准	
数据存储管理	存储监控	iiot_store_monitor		基础	
	数据库管理	iiot_db_management		基础	
	数据查询	iiot_data_query	基础		
	存储管理	iiot_store_management	基础		
	系统配置	iiot_base	基础		
系统运维	关键日志	iiot_keylog	基础		
	集群监控	iiot_cluster_manager、iiot_cluster_data_transfer、iiot_cluster_rpc	基础	分布式部署必备	
	设备节点管理	iiot_device_scheduler	基础	分布式部署必备	
	工况回补	iiot_thing	基础		
数据编织	数据建模	iiot_data_fabric	高阶		
	数据管理		高阶		
数据仿真	HMI (IOT)	iiot_hmi、iiot_hmi_2D、iiot_hmi_3D、iiot_hmi_layout	高阶		
	MI设计器	iiot_datasource	高阶	IIOT数据源扩展APP需要与MI安装在同一容器 (需安装MI-Base/IDA)	
消息推送	消息渠道配置	iiot_messagepush、iiot_messagepolicy、iiot_messagepolicy_alarm、iiot_messagepolicy_ruleengine (暂不推荐)、iiot_messagepolicy_store_management	高阶		
	消息模板配置		高阶		
	消息接收组		高阶		
	发送策略		高阶		
	消息统计		高阶		
数据存储	请求记录		高阶		
	数据归档	iiot_data_archive	高阶		
	数据备份	iiot_data_backup	高阶		
主数据扩展	IOT扩展MBM主数据	iiot-thing-mbm-mdm	高阶		

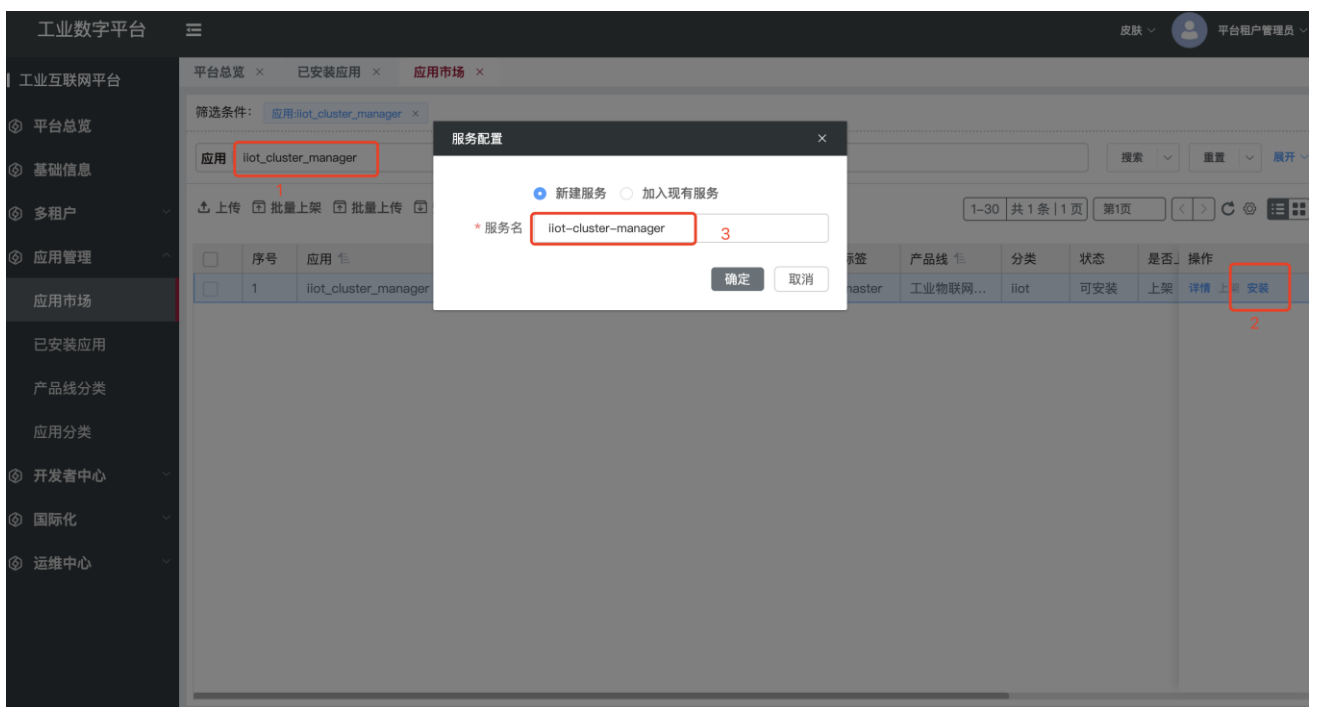
a. 加入已有服务

iiot_cluster_data_transfer 和 iiot_cluster_rpc 两个 app 要和已有 iiot 服务安装在一起, 有多个 iiot 服务时每个服务都要安装



b. 新建服务

iiot_cluster_manage 安装要新建服务，修改服务名 iiot-cluster-manage 点击安装

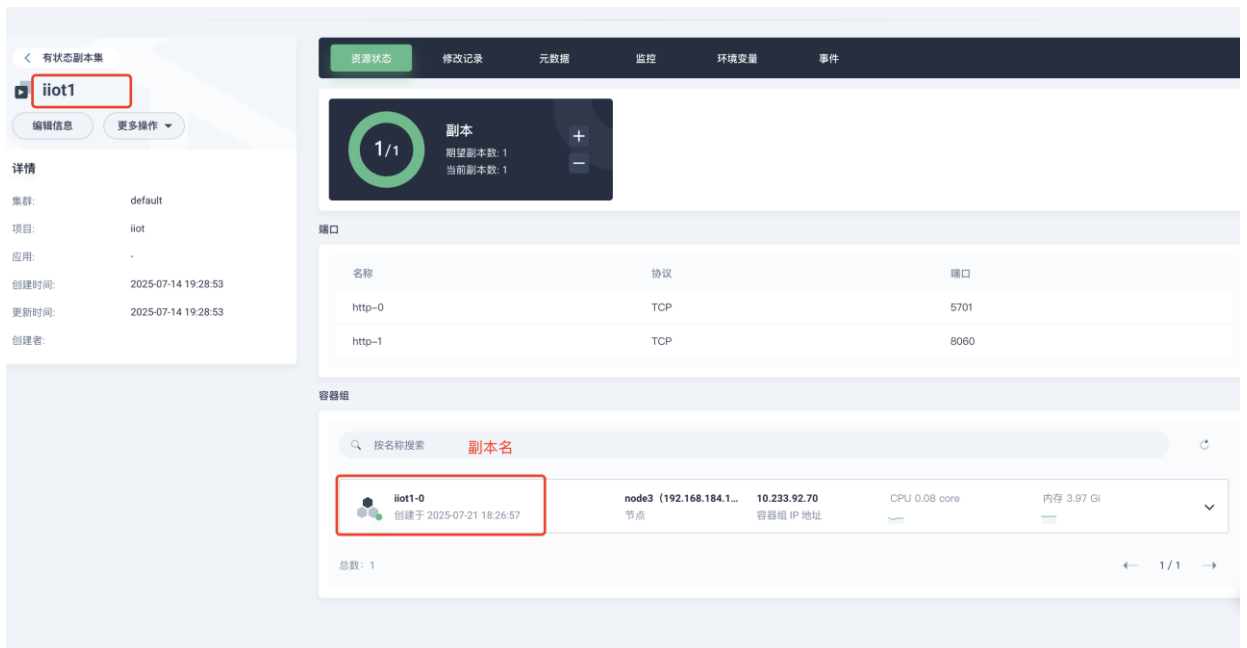


2. 配置副本服务

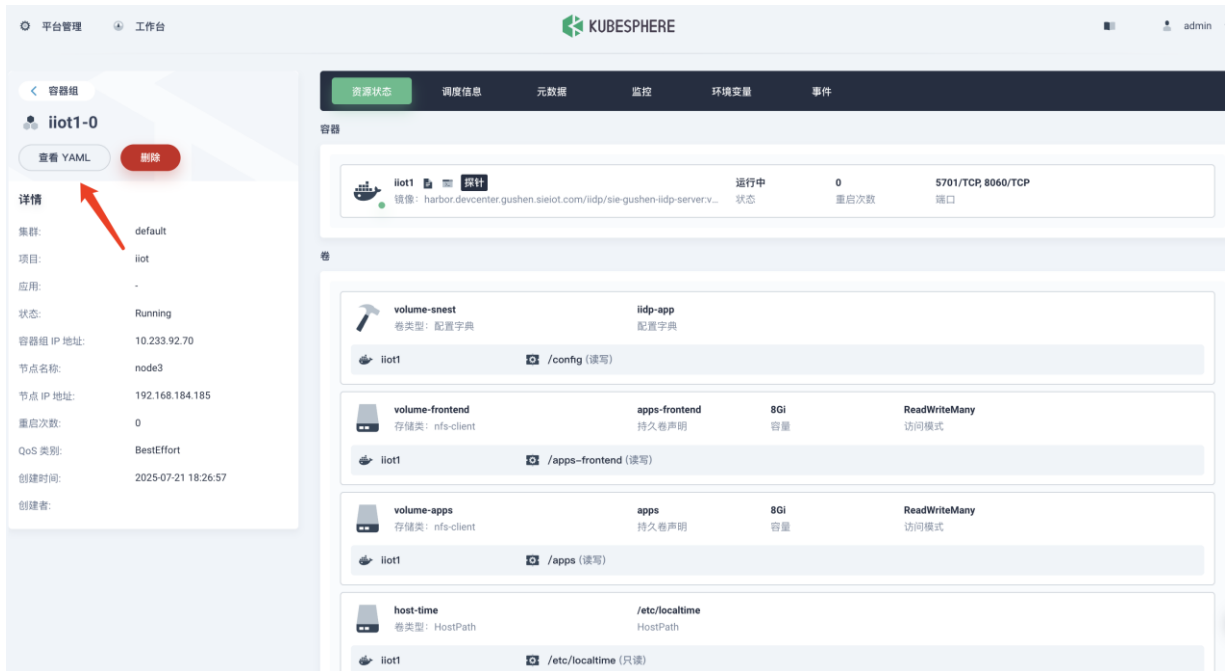
登录 kubernetes 管理页面，按照下面操作配置副本服务

1. 在工作负载->有状态副本->iiot1 下面查看当前负载有几个副本，有多个副本后续的操作就要有多个服务配置，每个服务配置对应一个副本（当前演示的场景只有一个副本）

注意：如果安装多个 iiot 服务，每个 iiot 服务都要配置下面副本对应的服务



2. 点击副本名跳转到副本容器组页面，点击查看 YAML



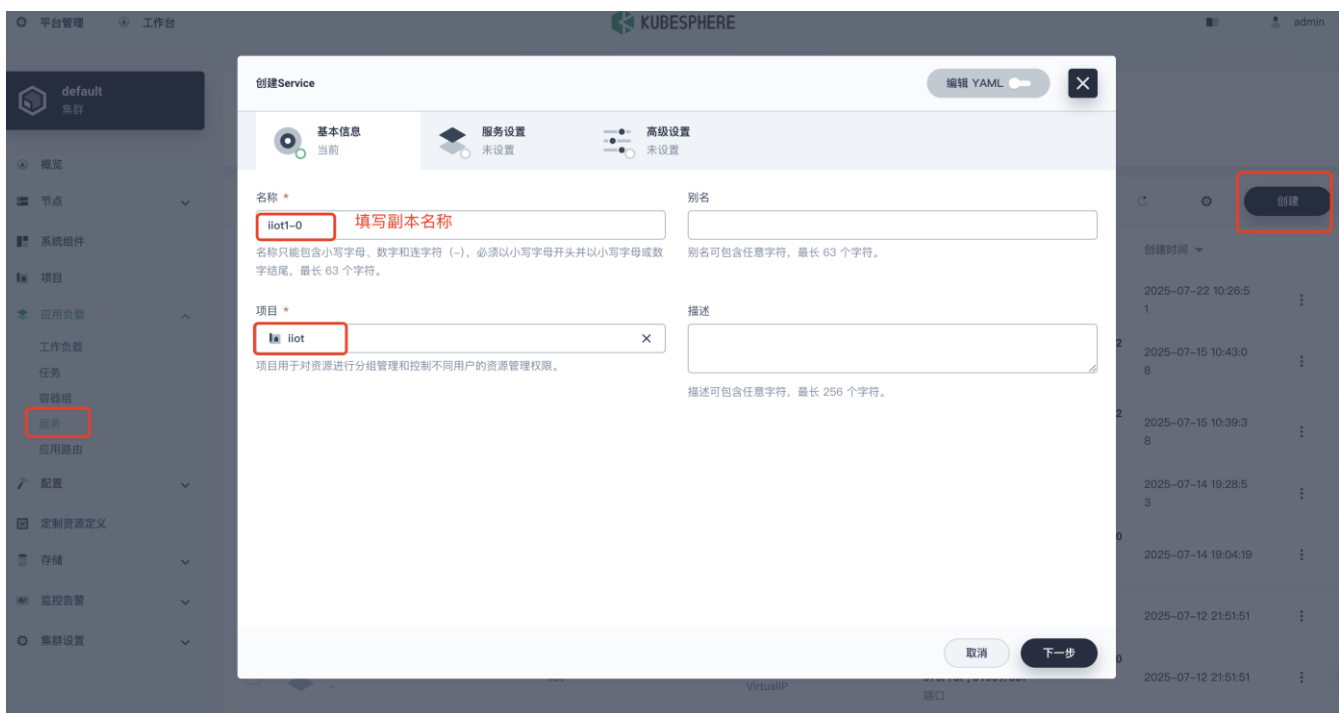
3. 复制 yaml 文件中这三行之后要

```
查看 YAML
```

```
1 kind: Pod
2 apiVersion: v1
3 metadata:
4   name: iiot1-0
5   generateName: iiot1-
6   namespace: iiot
7   labels:
8     app: iiot1
9     controller-revision-hash: iiot1-795d8d5555
10    iidp: iidp
11    statefulset.kubernetes.io/pod-name: iiot1-0
12  annotations:
13    cni.projectcalico.org/containerID: 4bc17144986846cbe95e2ab4d48735504b4e1f93406ad0282643f1a3e257316f
14    cni.projectcalico.org/podIP: 10.233.92.70/32
15    cni.projectcalico.org/podIPs: 10.233.92.70/32
16    iidp.com/restartedAt: '2025-07-21T10:21:28.024Z'
17    kubescape.io/imagepullsecrets: '[]'
18  spec:
19    volumes:
20      - name: volume-snest
21        configMap:
22          name: iidp-app
23          defaultMode: 420
24      - name: volume-frontend
25        persistentVolumeClaim:
26          claimName: apps-frontend
27      - name: volume-apps
28        persistentVolumeClaim:
29          claimName: apps
30      - name: host-time
31        hostPath:
32          path: /etc/localtime
33          type: ''
34      - name: kube-api-access-xb57n
```

复制这三行，之后要用

4. 进入当前项目的服务列表，点击”创建“，填写和副本名一样的名称，选择 iiot 项目 (如果没有 iiot 项目则去工作负载中查看副本所在项目，选择一样的) (如果已有此副本的服务则直接进入查看并记录端口，参考第 7 步)



5. 下一步，按照图片方式填写并配置端口，

编辑服务 ✕

内部访问模式

虚拟 IP 地址
为服务分配虚拟 IP 地址，可通过虚拟 IP 地址在集群内部访问服务。

工作负载选择器 *

没有工作负载匹配当前选择器。 刚刚上面记录的 3 个信息按如下格式填写

app	iio1	✕
liidp	liidp	✕
statefulset.kubernetes.io/pod-name	iio1-0	✕

指定工作负载 添加

端口 配置这两个端口

设置容器端口和服务端口。

协议 ?	HTTP	名称	http-8888	容器端口	8888	服务端口	8888	✕
协议 ?	TCP	名称	tcp-1883	容器端口	1883	服务端口	1883	✕

添加

取消 确定

6. 下一步，选择 NodePort 访问模式，点击创建

创建Service
编辑 YAML
✕

基本信息
已设置

服务设置
已设置

高级设置
当前

✓ **外部访问**
 设置从集群外访问服务的方式。 1

访问模式

NodePort

选择 NodePort 方式

2

会话保持
 设置系统在指定的时间内将同一个会话中来自同一个客户端的请求全部转发给同一个容器组。

添加元数据
为服务添加元数据。

标签 *

app

test-iiot1-0

✕

添加

取消
上一步
创建

7. 再次点击进入 iiot1-0 服务，记录服务分配的 2 个类型的外部访问端口，之后在系统配置时要使用

服务

iiot1-0

编辑信息 更多操作

详情

集群: default

项目: iiot

类型: 自定义服务 (VirtualIP)

应用: iiot1-0

虚拟 IP 地址: 10.233.30.156

Y 部 IP 地址:

Y 语象和性: 已禁用

选择器: app=iiot1,uidp=iiotp,statefulset.kubernetes.io/pod-name=iiot1-0

Y NS: iiot1-0.iiot

节点: 10.233.92.70:1883
10.233.92.70:8888

Y 建时间: 2025-07-15 10:43:08

Y 建者: admin

资源状态 元数据 事件

端口

8888 容器端口	→ TCP →	8888 服务端口	→ TCP →	30691 NodePort	websocket	记录这两个外部访问端口 后面会使用
1883 容器端口	→ TCP →	1883 服务端口	→ TCP →	30271 NodePort	mqtt	

工作负载

未发现可用工作负载

容器组

iiot1-0 创建于 2025-07-21 18:26:57

node3 (192.168.184.1... 节点) 10.233.92.70 容器组 IP 地址 CPU 0.08 core 内存 4.47 Gi

总数: 1

到此一个副本已配置完成，如果有多个 iiot 服务和副本需要对不同服务、副本重复以上配置。

3. 修改系统配置

- 通过菜单系统运维-系统配置进入页面
- 修改以下配置信息

配置键	说明	示例
iio <i>t</i> .iio <i>t</i> -0.mqt <i>t</i>	服务 iio <i>t</i> 的 iio <i>t</i> -0 副本向外暴露的 mqt <i>t</i> 地址 对应 4.2.3.3 章节配置副本服务记录的 mqt <i>t</i> 端口， 不同副本的服务端口不一样	192.168.174.169:30460
iio <i>t</i> 1.iio <i>t</i> 1-0.mqt <i>t</i>	服务 iio <i>t</i> 1 的 iio <i>t</i> 1-0 副本向外暴露的 mqt <i>t</i> 地址 对应 4.2.3.3 章节配置副本服务记录的 mqt <i>t</i> 端口， 不同副本的服务端口不一样	192.168.174.169:30271
iio <i>t</i> .iio <i>t</i> -0.websocke <i>t</i>	服务 iio <i>t</i> 的 iio <i>t</i> -0 副本向外暴露的 websocke <i>t</i> 地址 对应 4.2.3.3 章节配置副本服务记录的 websocke <i>t</i> 端口，不同副本的服务端口不一样	ws://192.168.174.169:32248/w s
iio <i>t</i> .iio <i>t</i> 1- 0.websocke <i>t</i>	服务 iio <i>t</i> 1 的 iio <i>t</i> 1-0 副本向外暴露的 websocke <i>t</i> 地 址 对应 4.2.3.3 章节配置副本服务记录的 websocke <i>t</i> 端口，不同副本的服务端口不一样	ws://192.168.174.169:30691/w s