



包学包会

DataFlux 部署手册

摘要

详细介绍 DataFlux 的资源配置与部署升级

陆宏鸣

lhm@jiagouyun.com

1. 前言	3
1.1. 产品简介	3
1.2. 本文档说明	3
1.3. 关键词	3
1.4. 部署架构	4
2. 资源准备	5
2.1. 资源清单	5
2.2. 创建资源	5
2.2.1. 基础资源	5
2.2.2. ACK 服务创建	6
2.3. 资源配置	8
2.3.1. RDS	8
2.3.2. Redis	8
2.3.3. InfluxDB	8
3. kubectl 安装及配置	9
3.1. 安装 kubectl	9
3.2. 配置 kube config	9
4. 开始安装 DataFlux	10
4.1. 自动存储配置	10
4.1.1. NAS Controller	10
4.1.2. 存储类配置	10
4.1.3. 导入存储配置	10
4.2. Launcher 服务安装配置	10
4.2.1. Launcher 服务 YAML 模板	10
4.2.2. 导入 Launcher 服务	11
4.2.3. 解析 launcher 域名到 launcher 服务	11
4.3. 应用安装引导步骤	11
4.3.1. 数据库配置	11
4.3.2. Redis 配置	11
4.3.3. InfluxDB 配置	11
4.3.4. 其他设置	11
4.3.5. 安装信息	12
4.3.6. 应用配置文件	12

4.3.7.	应用镜像	12
4.3.8.	应用状态	12
4.4.	域名解析	12
4.5.	很重要的步骤!!!	13
5.	升级安装.....	14
5.1.	Launcher 服务升级	14
5.2.	升级应用	14
5.2.1.	新增应用配置	14
5.2.2.	升级应用配置	14
5.2.3.	升级数据库.....	15
5.2.4.	升级应用	16
5.2.5.	应用启动状态	16
5.3.	很重要的步骤!!!	17

1. 前言

1.1. 产品简介

DataFlux 是驻云 CloudCare 品牌下的实时大数据分析平台，它包含 DataKit 采集器、DataWay 数据网关、DataFlux Studio 实时数据洞察平台，DataFlux Admin Console 管理后台，DataFlux.f(x) 实时数据处理开发平台五大功能模块。DataFlux 面向企业提供全场景的数据洞察分析能力，具有实时性、灵活性、易扩展、易部署等特点，支持云端 SaaS 和本地部署模式。

1.2. 本文档说明

本文档主要介绍从资源规划、配置开始，到部署 DataFlux、运行的完整步骤。

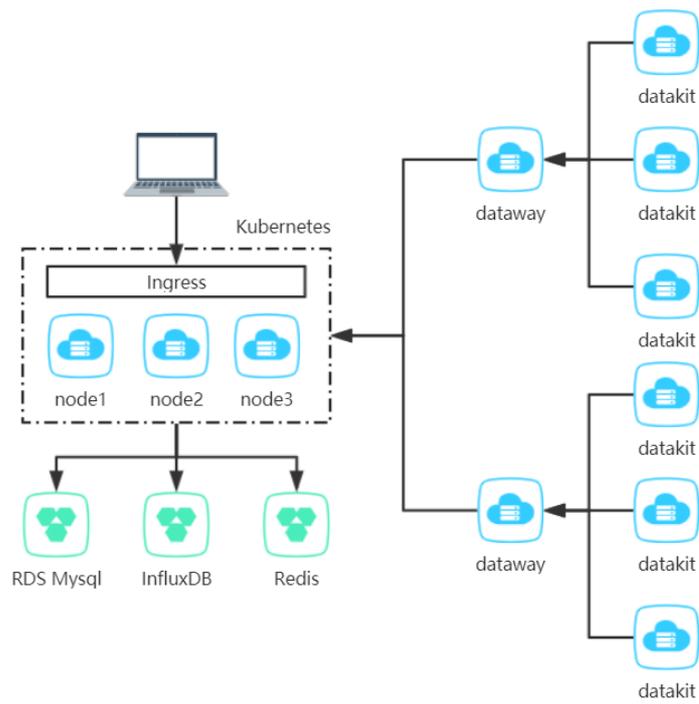
说明：

- 本文档以 `dataflux.cn` 为主域名示例，实际部署替换为相应的域名。

1.3. 关键词

词条	说明
Launcher	用于部署安装 DataFlux 的 WEB 应用，根据 Launcher 服务的引导步骤来完成 DataFlux 的安装与升级
运维操作机	安装了 kubectl，与目标 Kubernetes 集群在同一网络的运维机器
安装操作机	在浏览器访问 launcher 服务来完成 DataFlux 引导安装的机器
kubectl	Kubernetes 的命令行客户端工具，安装在 运维操作机 上

1.4. 部署架构



2. 资源准备

2.1. 资源清单

资源	规格（最低配置）	规格（推荐配置）	数量	备注
ACK	标准托管集群版	标准托管集群版	1	-
NAS	500GB（容量型）	1TB（容量型）	1	ACK 集群数据持久化
NAT 网关	小型 NAT 网关	小型 NAT 网关	1	ACK 集群出网使用
SLB	性能保障型	性能保障型	2	在 Kubernetes Ingress 前
ECS	2C4G（单系统盘 80GB）	4C8G（单系统盘 120GB）	3	部署阿里云 ACK 托管版集群
	1C2G（单系统盘 80GB）	2C4G（单系统盘 120GB）	1	部署 Dataway
RDS	1C2G100GB	2C4G100GB（三节点企业版）	1	MySQL 5.7
Redis	1G（Redis4.0）	2G（Redis4.0 标准主从版双副本）	1	-
InfluxDB	2C8G200GB	8C32G500GB（高可用版）	1	-
云通信	-	-	1	开通邮件服务、短信服务
域名	-	-	1	主域名需备案，一个主域名下的 8 个子域名
SSL 证书	通配符域名证书	通配符域名证书	1	-

2.2. 创建资源

2.2.1. 基础资源

RDS、Redis、InfluxDB、NAS 存储 按配置要求创建，创建到同一地域的同一个 VPC 网络下。

ECS、SLB、NAT 网关，由 ACK 来自动创建，不需要单独创建。

2.2.2. ACK 服务创建

2.2.2.1. 集群配置

进入 **容器服务 kubernetes 版**，创建 **Kubernetes 集群**，选择 **标准托管集群版**，集群配置注意事项：

- 必须与前面创建的 RDS、InfluxDB 等资源同一地域
- 勾选“配置 SNAT”选项（ACK 自动创建和配置 NAT 网关，使集群有出网能力）
- 勾选“公网访问”选项（可以在公网访问集群 API，如果是在内网运维此集群，可以不勾选此项）

The screenshot shows the configuration page for an ACK cluster. The region is set to '华东1 可用区G'. The network plugin is 'Flannel'. The Pod network CIDR is '172.20.0.0/16' and the Service CIDR is '172.21.0.0/20'. The number of nodes is set to 64. The '配置 SNAT' option is checked, with a red arrow pointing to it and the text '勾选“配置 SNAT”选项 ACK 自动创建和配置 NAT 网关'. The '公网访问' option is also checked, with a red arrow pointing to it and a text box containing '勾选“公网访问”可以在公网访问集群 API 否则只能在内网访问'. The 'RDS白名单' section is currently empty.

2.2.2.2. Worker 配置

主要是选择 ECS 规格及数量，规格可以按配置清单要求来创建，或者按实际情况评估，但不可低于最低配置要求，数量至少 3 台，或 3 台以上。

Worker 实例 新增实例 添加已有实例

付费类型 按量付费 包年包月

查看两种计费方式区别 [详细对比](#)
您目前可以通过 ECS 管理控制台将按量付费实例转换成包年包月实例。 [查看详情](#)

实例规格 当前代 所有代

x86 计算 异构计算 GPU / FPGA / NPU 弹性裸金属服务器 (神龙) 超级计算集群

通用型 计算型 内存型 大数据型 本地 SSD 存储增强型 高主频型 共享型

规格族	实例规格	vCPU	内存	可用区	ENI 数量
<input type="radio"/> 计算型 (原独享) sn1	ecs.sn1.large	4 vCPU	8 GiB	G	3
<input type="radio"/> 计算型 (原独享) sn1	ecs.sn1.xlarge	8 vCPU	16 GiB	G	4
<input checked="" type="radio"/> 计算型 (原独享) sn1	ecs.sn1.medium	2 vCPU	4 GiB	G	2

已选规格 2 核 4 G (ecs.sn1.medium) ✕

您可以选择多个实例规格作为备选，每个节点创建时，将从第一个规格开始尝试购买，直到创建成功。最终购买的实例规格可能随库存变化而不同。

数量 3 台

节点将尽可能均匀分布在您所选择的多个虚拟交换机中

系统盘 高效云盘 120 GiB

挂载数据盘 你已选择 0 块盘，还可以选择 10 块盘

+ 增加一块数据盘 推荐

2.2.2.3. 组件配置

组件配置，必须勾选“安装 Ingress 组件”选项，选择“公网”类型，ACK 会自动创建一个公网类型的 SLB，安装完成后，将域名触到到此 SLB 的公网 IP。

创建 Kubernetes 集群 Kubernetes 专有版 Kubernetes 托管版 Serverless Kubernetes Kubernetes 边缘托管版 (公测)

集群配置 Worker 配置 3 组件配置

Ingress 安装 Ingress 组件

负载均衡类型 公网 内网

创建 Ingress Dashboard 需要同步安装日志服务组件

存储插件 Flexvolume CSI

云监控插件 在 ECS 节点上安装云监控插件 推荐

在节点上安装云监控插件，可以在云监控控制台查看所创建ECS实例的监控信息

日志服务 使用日志服务

使用已有 Project 创建新 Project

将自动创建名称为 k8s-log-(ClusterID) 的 Project

工作流引擎 AGS

勾选“安装 Ingress 组件”选项
选择公网类型的负载均衡
ACK 会自动创建一个公网类型的 SLB
安装完成后
域名解析到此 SLB 即可访问服务

2.3. 资源配置

2.3.1. RDS

- 创建管理员账号（必须是**管理员账号**，后续安装初始化需要用此账号去创建和初始化各应用 DB）
- 修改参数配置，将 `innodb_large_prefix` 设置为 ON
- 将 ACK 自动创建的 ECS 内网 IP，添加到 RDS 白名单

2.3.2. Redis

- 设置 Redis 密码
- 将 ACK 自动创建的 ECS 内网 IP，添加到 Redis 白名单

2.3.3. InfluxDB

- 创建管理员账号（必须是**管理员账号**，后续安装初始化需要用此账号去创建和初始化 DB 及 RP 等信息）
- 开通 VPC 双向访问功能
- 将 ACK 自动创建的 ECS 内网 IP，添加到 InfluxDB 白名单

3. kubectl 安装及配置

3.1. 安装 kubectl

kubectl 是一个 kubernetes 的一个命令行客户端工具，可以通过此命令行工具去部署应用、检查和管理集群资源等。

我们的 Launcher 就是基于此命令行工具，去部署应用的，具体安装方式可以看官方文档：

<https://kubernetes.io/docs/tasks/tools/install-kubectl/>

3.2. 配置 kube config

kubectl 要获得管理集群的能力，需要将集群的 kubeconfig 内容放入 `$HOME/.kube/config` 文件内，kubeconfig 内容可以在集群 **基本信息** 中查看到。

选择 公网访问还是内网访问的 kubeconfig，取决于你的运维操作机是否与集群内网联通。

The screenshot shows the 'Cluster Information' (集群基本信息) page in the Kubernetes console. It displays various cluster details in a table:

Service CIDR	172.21.0.0/20
测试域名	*.cc3054adb0144735a1160f474877dfe3.cn-hangzhou.alicontainer.com
kube-proxy 代理模式	ipvs
节点 IP 数量	64
网络插件	Flannel

Below this is the 'Cluster Resources' (集群资源) section with a table of resources:

资源编排 ROS	k8s-for-cc-3054adb0144735a1160f474877dfe3
虚拟专有网络 VPC	vpc-3054adb0144735a1160f474877dfe3
节点虚拟交换机	vsw-bp18o-3054adb0144735a1160f474877dfe3
安全组	sg-bp18o-3054adb0144735a1160f474877dfe3
Worker RAM 角色	KubernetesWorkerRole-3054adb0144735a1160f474877dfe3
日志服务 Project	k8s-log-cc3054adb0144735a1160f474877dfe3
Nginx Ingress SLB	lb-bp19s-3054adb0144735a1160f474877dfe3

At the bottom, there are instructions for connecting to the cluster via kubectl. A red arrow points to the 'KubeConfig (内网访问)' option, with a red text box containing the instruction: **内网运维机器使用“内网访问 KubeConfig”**.

将以下内容复制到计算机 `$HOME/.kube/config`

```
apiVersion: v1
clusters:
- cluster:
  server: https://116.62.116.62:443
  certificate-authority-data:
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0tLS0tCk1JSURHakNDQWdLZ0F3SUJBZ01CQURBTKJna3Foa21lHOXcwQkFRc0ZBREErTVNjdD
iM1V4RXpBUkFnTlZCQU1UQ210MVltVnlibVYwW1hNdwPIAGNOTWpBd016RXpNRGN5TkrRFeFdoY05NekF3TXpFeE1EY3l0REV4V2pB
VaM3BvYjNVeEV6QVJCZ05WQkFNVENTdDFZbvZ5Ym1WMPpYTXcKZ2dFaU1BMEdDU3FHU01lM0RRRUJBUVVBQTRJQkR3QXdnZ0VlQW93
05SSDBtLZ2ZicUc3em9rMF15dGv2cG5ITG5UNEM4SDNsVU5XUFVlSFVtCmIzaklyK0tkVEovUXBI2VF30StYMzAxNjQ1dzMjVORTN
```

4. 开始安装 DataFlux

4.1. 自动存储配置

4.1.1. NAS Controller

NAS Controller YAML 文件下载地址:

http://static.dataflux.cn/launcher/nas_controller.yaml

将上面的 YAML 内容保存为 `nas_controller.yaml` 文件，放到运维操作机上。

4.1.2. 存储类配置

Storage Class YAML 下载:

http://static.dataflux.cn/launcher/storage_class.yaml

将上面的 YAML 内容保存为 `storage_class.yaml` 文件，放到运维操作机上，然后替换文档内的变量部分:

- `{{ nas_server_id }}` 替换为前面创建的 NAS 存储的 Server ID。

4.1.3. 导入存储配置

- 导入 `controller.yaml`，在运维操作机上执行命令：
`kubectl apply -f ./nas_controller.yaml`
- 导入 `storage_class.yaml`，在运维操作机上执行命令：
`kubectl apply -f ./storage_class.yaml`

4.2. Launcher 服务安装配置

4.2.1. Launcher 服务 YAML 模板

Launcher YAML 下载:

<http://static.dataflux.cn/launcher/launcher.yaml>

将上面的 YAML 内容保存为 `launcher.yaml` 文件，放到运维操作机上，然后替换文档内的变量部分:

- `{{ domain }}` 替换为主域名，如使用 `dataflux.cn`
- `{{ kube_config }}` 替换为 `kube config`，`launcher` 需要获取到集群权限，去自动部署应用，注意缩进
- `{{ launcher_image }}` 替换为 Launcher 镜像，在公共镜像仓库，取对应的版本目录内的 `launcher` 镜像地址

- `{{ storageClassName }}` 替换为 storage class name, 如前一步的 `storage_class.yaml` 中配置的 `name: alicloud-nas`

4.2.2. 导入 Launcher 服务

在**运维操作机**上执行以下 `kubectl` 命令, 在导入 **Launcher** 服务:

```
kubectl apply -f ./launcher.yaml
```

4.2.3. 解析 launcher 域名到 launcher 服务

因为 launcher 服务为部署和升级 DataFlux 使用, 不需要对用户开放访问, 所以域名不要在公网解析, 可以在**安装操作机**上, 绑定 host 的方式, 模拟域名解析, 在 `/etc/hosts` 中添加 `launcher.dataflux.cn` 的域名绑定:

```
192.168.0.1 launcher.dataflux.cn
```

`192.168.0.1` 实际为前面步骤中创建 ACK 时自动创建的 SLB 实例的公网 IP 地址。

4.3. 应用安装引导步骤

在**安装操作机**的浏览器中访问 `launcher.dataflux.cn`, 根据引导步骤一步一步完成安装配置。

4.3.1. 数据库配置

- 数据库连接地址必须使用内网地址
- 账号必须使用管理员账号, 因为需要此账号去初始化多个子应用的数据库及数据库访问账号

4.3.2. Redis 配置

- Redis 连接地址必须使用内网地址

4.3.3. InfluxDB 配置

- InfluxDB 链接地址必须使用内网地址
- 账号必须使用管理员账号, 因为需要使用此账号去初始化 DB 以及 RP 待信息
- 可添加多个 InfluxDB 实例

4.3.4. 其他设置

- DataFlux 后台管理员的初始账号信息
- 集群节点内网 IP (会自动获取, 需要确认是否正确)
- 主域名及各子应用的子域名配置, 默认子域名如下, 可根据需要修改:
 - `dataflux`
 - `df-api`
 - `df-management`
 - `df-management-api`

- df-websocket
- df-func
- df-integration
- df-kodo
- TLS 域名证书填写

4.3.5. 安装信息

汇总显示刚才填写的信息，如有信息填写错误可返回上一步修改

4.3.6. 应用配置文件

安装程序会自动根据前面步骤提供的安装信息，初始化应用配置模板，但还是需要逐个检查所有应用模板，修改个性化应用配置，具体配置说明见安装界面。

确认无误后，提交创建配置文件。

4.3.7. 应用镜像

- 选择正确的**共享存储**，即你前面步骤中创建的 **storage class** 名称
- 应用镜像会根据你选的 **Launcher** 版本，自动填写无需修改，确认无误后开始 **创建应用**

4.3.8. 应用状态

列出所有应用的服务启动状态，待全部服务都成功启动之后，会自动保存当前应用的版本信息，需要等待看到 **写入版本信息成功!** 字样，表示安装成功

4.4. 域名解析

将除 **df-kodo.dataflux.cn** 之外的其他所有子域名，都解析到 ACK 自动创建的 SLB 公网 IP 地址上：

- dataflux.dataflux.cn
- df-api.dataflux.cn
- df-management.dataflux.cn
- df-management-api.dataflux.cn
- df-websocket.dataflux.cn
- df-func.dataflux.cn
- df-integration.dataflux.cn

服务安装完成之后，集群会自动为 **kodo** 服务创建一个公网 SLB，可使用 `kubectl get svc -n forethought-kodo` 命令，查看到 **kodo-nginx** 服务的 EXTERNAL-IP，**df-kodo.dataflux.cn** 子域名区别单独解析到此 SLB 的公网 IP 上，如下图：

```
luhongming@~/k8s $ kubectl get svc -n forethought-kodo
NAME          TYPE          CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
kodo          ClusterIP     172.21.1.11     <none>           9527/TCP         32d
kodo-inner    ClusterIP     172.21.1.12     <none>           9527/TCP         32d
kodo-nginx    LoadBalancer 172.21.1.13     47.114.11.11     443/TCP,80/TCP  32d
```

df-kodo.dataflux.cn 域名
单独解析到此公网 IP 上

4.5. 很重要的步骤!!!

经过以上步骤，DataFlux 都安装完毕，可以进行验证，验证无误后一个很重要的步骤，将 launcher 服务下线，防止被误访问而破坏应用配置，可在**运维操作机**上执行以下命令，将 launcher 服务的 pod 副本数设为 0：

```
kubectl patch deployment launcher \
-p '{"spec": {"replicas": 0}}' \
-n launcher
```

5. 升级安装

5.1. Launcher 服务升级

升级 DataFlux，首先第一步是升级 **Launcher** 服务，获取到需要升级的目标版本的 **launcher** 服务镜像地址，在**运维操作机**上执行以下命令，升级 **launcher**：

```
kubectl patch launcher \
-p '{"spec": {"template": {"spec": {"containers": [{"image": "镜像地址", "name": "launcher"}]}}}}' \
-n launcher
```

执行以下命令，将 **launcher** 副本上设置为 1：

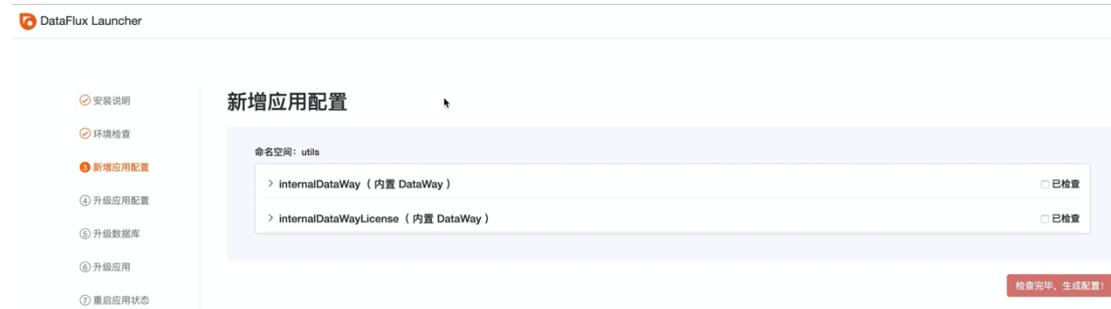
```
kubectl patch deployment launcher \
-p '{"spec": {"replicas": 1}}' \
-n launcher
```

5.2. 升级应用

在**安装操作机**的浏览器上访问 **launcher.dataflux.cn**，根据安装引导步骤完成 DataFlux 的升级。

5.2.1. 新增应用配置

launcher 自动检测当前的 DataFlux 版本，到目标升级版本之间，新增加的应用配置，并列出，根据配置模板，修改相应的值后，点击“检查完毕，生成配置”。



5.2.2. 升级应用配置

- **launcher** 自动检测当前的 DataFlux 版本，到目标升级版本之间，有更新的应用配置，根据列出的更新内容，修改相应的值。

- ① 安装说明
- ② 环境检查
- ③ 新增应用配置
- ④ 升级应用配置
- ⑤ 升级数据库
- ⑥ 升级应用
- ⑦ 重启应用状态

升级应用配置

- 展开标记有 **配置有更新** 的配置项，左侧列出了需要升级的历史版本，右侧是应用当前的配置，根据左侧的配置升级描述，在右侧修改配置内容。

- ① 安装说明
- ② 环境检查
- ③ 新增应用配置
- ④ 升级应用配置
- ⑤ 升级数据库
- ⑥ 升级应用
- ⑦ 重启应用状态

升级应用配置

- 同时也可以修改未标记**配置有更新**的应用配置，勾选**升级配置**选项后，可以修改配置。
- 确认所有配置修改完毕后，点击 **确认升级配置** 按钮升级配置。

5.2.3. 升级数据库

launcher 自动检测当前的 DataFlux 版本，到目标升级版本之间，有数据库升级的应用，并列出每个版本的数据库升级内容，点击**确认升级**：

- ① 安装说明
- ② 环境检查
- ③ 新增应用配置
- ④ 升级应用配置
- ⑤ 升级数据库
- ⑥ 升级应用
- ⑦ 重启应用状态

升级数据库

数据库的历史升级版本

项目: core (命名空间: forethought-core)

序列: 4

```
-- 增加内置 DataWay
INSERT INTO `main_agent` (`uid`, `name`, `version`, `host`, `port`, `domainName`, `workspaceUID`, `status`, `creator`, `updater`, `createAt`, `uploadAt`
VALUES (`agnt_internal_dataway_1`, '内置 DataWay', '0.1-349-gfff24b0', '', 0, '', `wksp_system`, 0, '', '', UNIX_TIMESTAMP(), UNIX_TIMESTAMP(), -1, UNIX_
```

序列: 5

```
-- 节点新增 是否打开继承开关字段, 默认打开
ALTER TABLE `biz_node` ADD COLUMN `isInheritance` tinyint(1) NOT NULL DEFAULT 1 COMMENT '是否继承' AFTER `filter`;

-- 节点上新增2个字段, 用来控制子节点的图标和继承信息
ALTER TABLE `biz_node`
ADD COLUMN `subIconSet` json NULL COMMENT '子节点的图标信息' AFTER `exclude`,
ADD COLUMN `subIsInheritance` tinyint(1) NOT NULL DEFAULT 1 COMMENT '子节点是否继承父节点过滤条件' AFTER `subIconSet`;

-- 新增ck 黑名单表 03/16/2020
CREATE TABLE `main_ck_blacklist` (
  `id` int(11) NOT NULL AUTO_INCREMENT COMMENT '自增 ID',
  `uid` varchar(48) NOT NULL DEFAULT '' COMMENT '全局唯一 ID 前缀 ckb1-',
  `dbUID` varchar(48) NOT NULL DEFAULT '' COMMENT '原始 Measurement 所在 DB uid',
  `measurement` varchar(48) NOT NULL DEFAULT '' COMMENT '指标名',
  `status` int(11) NOT NULL DEFAULT '0',
  `creator` varchar(64) NOT NULL DEFAULT '' COMMENT '创建者 account-id',
  `updater` varchar(64) NOT NULL DEFAULT '' COMMENT '更新者 account-id',
  `createAt` int(11) NOT NULL DEFAULT '-1',
  `deleteAt` int(11) NOT NULL DEFAULT '-1',
  `updateAt` int(11) NOT NULL DEFAULT '-1',
  PRIMARY KEY (`id`),
  UNIQUE KEY `uk_uid` (`uid`) COMMENT 'UID 做成全局唯一'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

©2013-2020 上海轻云信息科技有限公司

5.2.4. 升级应用

launcher 自动检测当前的 DataFlux 版本, 到目标升级版本之间, 有升级或者是新增加的应用, 并显示当前版本与目标升级版本的镜像版本对比, 点击**确认升级应用**:

- ① 安装说明
- ② 环境检查
- ③ 新增应用配置
- ④ 升级应用配置
- ⑤ 升级数据库
- ⑥ 升级应用
- ⑦ 重启应用状态

升级应用

命名空间: forethought-core

列出了所有更新或者新增的应用

更新的应用:

后台管理平台 API:	pubrepo.jiagouyun.com/dataflux/1.2.29:core-9af8178-1582989625
	pubrepo.jiagouyun.com/dataflux/1.3.32:core-5a25218-1584511651
Inner API:	pubrepo.jiagouyun.com/dataflux/1.2.29:core-9af8178-1582989625
	pubrepo.jiagouyun.com/dataflux/1.3.32:core-5a25218-1584511651
用户前台 API:	pubrepo.jiagouyun.com/dataflux/1.2.29:core-9af8178-1582989625
	pubrepo.jiagouyun.com/dataflux/1.3.32:core-5a25218-1584511651
集成扫描 Worker:	pubrepo.jiagouyun.com/dataflux/1.2.29:core-9af8178-1582989625
	pubrepo.jiagouyun.com/dataflux/1.3.32:core-5a25218-1584511651
Websocket:	pubrepo.jiagouyun.com/dataflux/1.2.29:core-9af8178-1582989625
	pubrepo.jiagouyun.com/dataflux/1.3.32:core-5a25218-1584511651

©2013-2020 上海轻云信息科技有限公司

5.2.5. 应用启动状态

显示所有更新或者新增加的应用启动状态, 等全部启动完毕, 即完成了此版本 DataFlux 的升级安装。

- 安装说明
- 环境检查
- 新增应用配置
- 升级应用配置
- 升级数据库
- 升级应用
- 重启应用状态

重启应用状态 8 / 27

命名空间: forethought-core

后台管理平台 API:	pubrepo.jiagouyun.com/dataflux/1.3.32:core-5a25218-1584511651 ↻
Inner API:	pubrepo.jiagouyun.com/dataflux/1.3.32:core-5a25218-1584511651 ↻
用户前台 API:	pubrepo.jiagouyun.com/dataflux/1.3.32:core-5a25218-1584511651 ↻
集成扫描 Worker:	pubrepo.jiagouyun.com/dataflux/1.3.32:core-5a25218-1584511651 ↻
Websocket:	pubrepo.jiagouyun.com/dataflux/1.3.32:core-5a25218-1584511651 ↻

命名空间: forethought-inner-app

通知触发器:	pubrepo.jiagouyun.com/dataflux/1.3.32:trigger-d9c7b6c-1584511651 ↻
--------	--

命名空间: forethought-webclient

管理后台前端:	pubrepo.jiagouyun.com/dataflux/1.3.32:management-webclient-6d07025-1584511651 ↻
用户前台前端:	pubrepo.jiagouyun.com/dataflux/1.3.32:front-webclient-3d55fa3e-1584511651 ↻

命名空间: forethought-kodo

©2013-2020 上海驻云信息科技有限公司

5.3. 很重要的步骤!!!

升级完成后，进行升级后的验证，验证无误后一个很重要的步骤，将 launcher 服务下线，防止被误访问而破坏应用配置，可在**运维操作机**上执行以下命令，将 launcher 服务的 pod 副本数设为 0:

```
kubectl patch deployment launcher \
-p '{"spec": {"replicas": 0}}' \
-n launcher
```