

# CMDB 如何管理主机

## 情景

新业务上线（常见的三层架构：接入层、逻辑层、存储层），需要用 CMDB 管理业务上线依赖的主机资源，便于后续实现发布、变更、故障处理等场景的自动化流程。

## 前提条件

在配置平台中 **新建业务**，并导入主机及将主机分配到业务中。

## 操作步骤

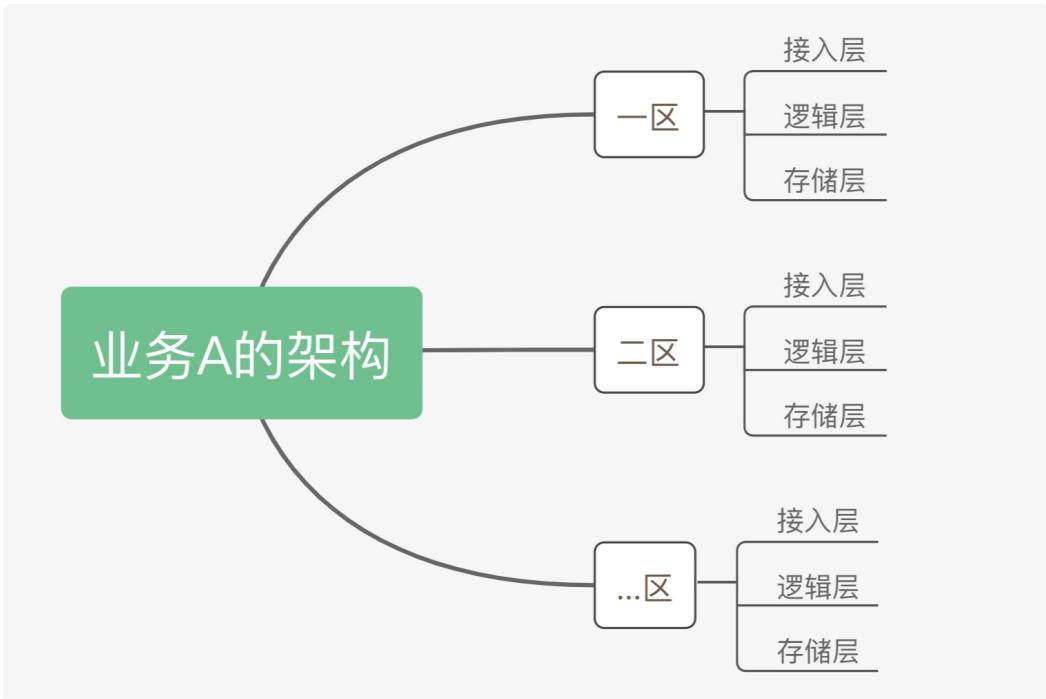
- 梳理业务架构
- 新建服务模板+集群模板
- 通过集群模板新建集群
- 主机分配
- 使用场景：在作业平台中查询接入层的磁盘使用率

### 梳理业务架构

业务的架构设计最终要体现在 CMDB 的业务拓扑树上，所以我们先梳理业务的架构。

以业务“欢乐游戏( demo )"为例，其架构是常见的三层架构： **接入层 -> 逻辑层 -> 存储层**

为了满足全国各地用户就近接入，按照地域划分了不同的区，但架构一致。



### 新建服务模板+集群模板

按照上述业务架构，可将做如下对应：

- 业务中的区 <-> CMDB 中的集群
- 业务中的分层架构 <-> CMDB 中模块

相关概念：【**服务模板**】、【**集群模板**】

#### 1. 新建服务模板

服务模板列表界面，显示了三个已有的服务模板：存储层、逻辑层、接入层。右侧有一个“新建”按钮，带有红色箭头指向。

ID	模块名称	服务分类	进程数量	已应用模块数	修改人	修改时间	操作
99	存储层	存储 / Ceph	0 (未配置)	0	admin	2021-04-07 14:48	<a href="#">删除</a>
98	逻辑层	消息队列 / RabbitMQ	0 (未配置)	0	admin	2021-04-07 14:48	<a href="#">删除</a>
97	接入层	HTTP 服务 / Tomcat	0 (未配置)	0	admin	2021-04-07 14:48	<a href="#">删除</a>

“

服务分类用于标识当前模块的实际功能组件，例如 mysql 组件、Nginx 组件，甚至业务的自定义组件等

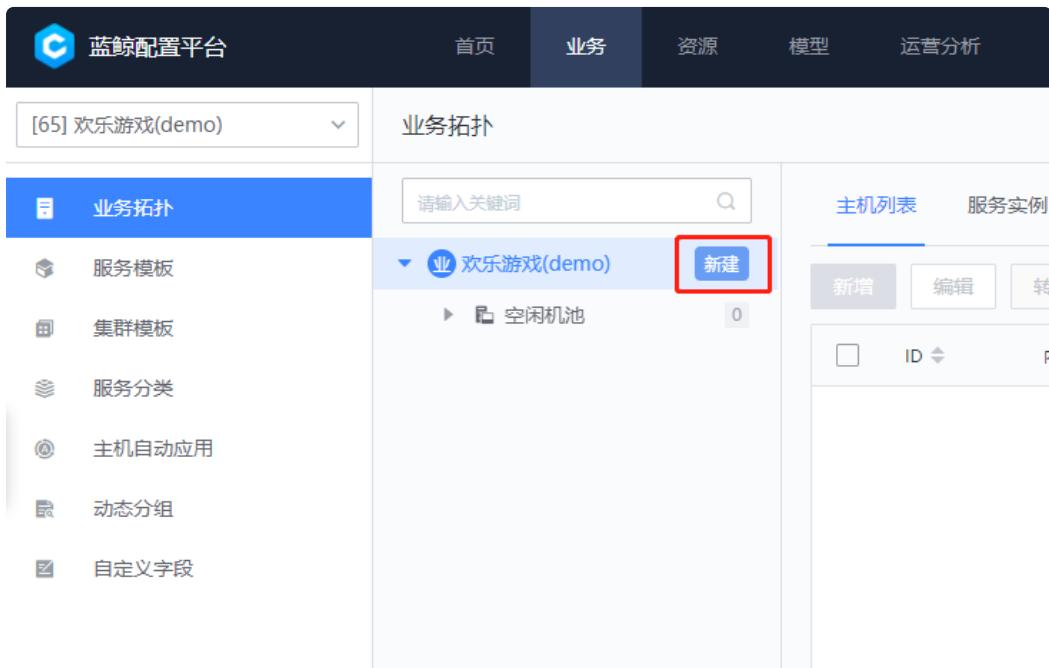
”

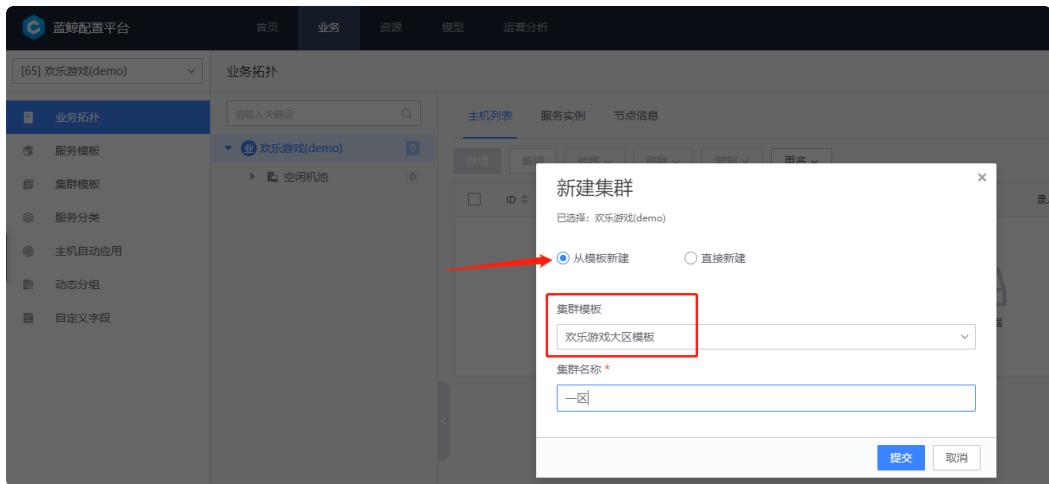
#### 1. 新建集群模板



### 通过集群模板新建集群

在 **业务拓扑** 中, 选中根节点 **业务**, 新建节点 **集群** 一区。





## Host Allocation

If no hosts have been assigned to the business idle pool, please refer to [this document](#).

Assign the hosts under the business idle pool to the corresponding modules.

[65] 欢乐游戏(demo)

业务拓扑

主机列表 服务实例 节点信息

新增 编辑 移除 复制 更多

至空闲模块  
至业务模块

ID: 14831

default area[0] -- --

共计 1 条 每页 20 条 已选择 1 条

## 节点信息配置

根据业务侧的需要进行配置 业务、集群、模块 的节点信息。

- 业务节点信息配置



业务节点信息，只能去【资源-业务-属性】进行修改



资源目录

业务 【欢乐游戏(demo)】

业务 属性 关联 变更记录

基础信息

业务名	:	欢乐游戏(demo)	生命周期	:	已上线
时区	:	Asia/Shanghai	语言	:	中文

角色

运维人员	:	admin(老大)	产品人员	:	admin(老大)
测试人员	:	admin(老大)	开发人员	:	admin(老大)
操作人员	:	admin(老大)	fdsafa	:	--
一级部门Id	:	--			

- 集群节点信息配置



“

环境类型：一般保持正式；测试环境设置为测试，让测试环境的发布流程模板只能选择测试集群；服务状态：一般保持开放，在某些场景如分区准备的时候，状态可置为关闭，让发布系统无法选中；

”

- 模块节点信息配置



“

模块类型：一般为 **普通**，当模块类型为数据库类时，选择 **数据库** 主要维护人、备份维护人：该模块维护人，一般在告警推送时会用到；

”

### 使用场景：在作业平台中查询接入层的磁盘使用率

通过一个简单的场景，体验 **作业平台** 如何消费主机实例。

```
{% video %}media/cmdb_job_consume.mp4{% endvideo %}
```

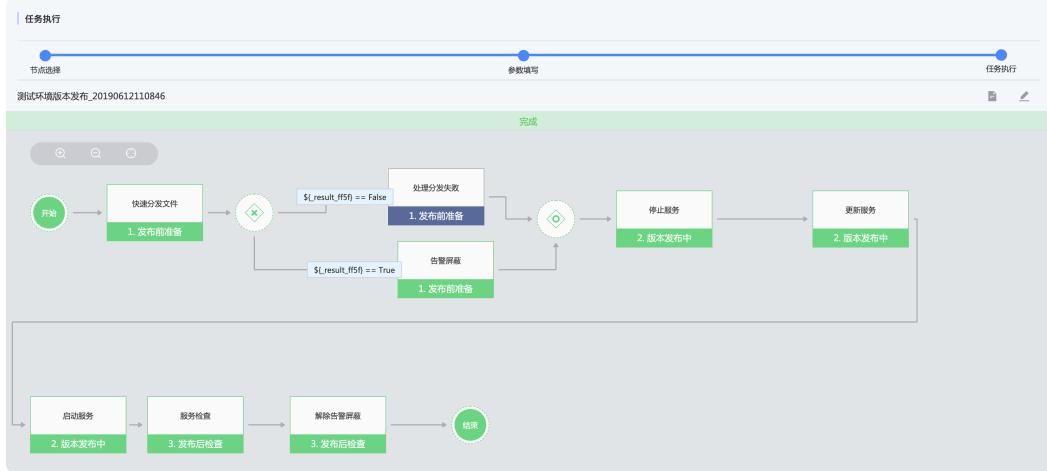
### 扩展阅读

#### 蓝鲸内置 SaaS 的 CMDB 消费场景

- 应用发布、变更：资源编排工具 标准运维

应用发布、变更流程包含版本在多台主机上的文件分发、命令执行等操作，如何优雅的选择这批主机，

需要使用 CMDB 的查询主机实例功能。



在参数中使用动态 IP 变量，运维无需关心主机扩缩容、故障替换等场景带来的主机变更，无需担心漏更新主机或更新错主机。

参数信息

版本发布目标IP \*

静态IP  动态IP

已选择 2 个节点

搜索节点 🔍

- ▼  欢乐游戏(demo)
- ▼  空闲机池
- 空闲机
- 故障机
- ▼  Android\_Weixin
- ▼  一区
  - 接入层
  - 逻辑层
  - 存储层
- ▼  二区
  - 接入层
  - 逻辑层
  - 存储层
- Android\_QQ

---

筛选条件 ( 同时满足 ) :

增加一条筛选条件

排除条件 ( 同时满足 ) :

集群 ▼

二区

+ -

上一步

- 故障处理：监控、故障自愈

针对业务架构中的某一层级模块（如接入层）设置一个告警检测策略，无需关心实例的新增、删除及修改。

## 性能指标

\* 指标分类

磁盘

指标名称

磁盘使用率

采集周期: 1分钟

## 触发条件

\* 监控对象  按主机IP  按业务拓扑



如何实现实时感知，背后的逻辑是通过 CMDB 的事件推送功能，实时感知实例的新增、修改、删除等动作。

The screenshot shows two main sections. On the left is a table titled '事件推送' (Event Push) with columns: 推送名称 (Push Name), 系统名称 (System Name), 操作人 (Operator), and 更新时间 (Update Time). It lists two entries: 'gse\_agentId' and 'process instance refresh [Do not remove...]' both by 'migrate' operator at '2019-06-11 02:3'. On the right is a '编辑推送' (Edit Push) panel with fields for '推送名称' (Push Name: gse\_agentId), '系统名称' (System Name: gse), 'URL' (http://10.0.4.51:52050/gsesyncdata/v1/c), and a '测试推送' (Test Push) button. Below these are sections for '成功确认方式' (Success Confirmation Method: HTTP状态 or 正则验证), a status code input (200), and a timeout input (10 s). A note says '至少选择1个事件, 已选择 1 个' (At least one event selected, 1 selected). There are also dropdown menus for '主机业务' (Host Business), '业务拓扑' (Business Topology), and '组织架构' (Organizational Structure).

在 **故障自愈** 中的消费场景也是如此，一个或多个模块的某一个告警，关联对应的处理动作。

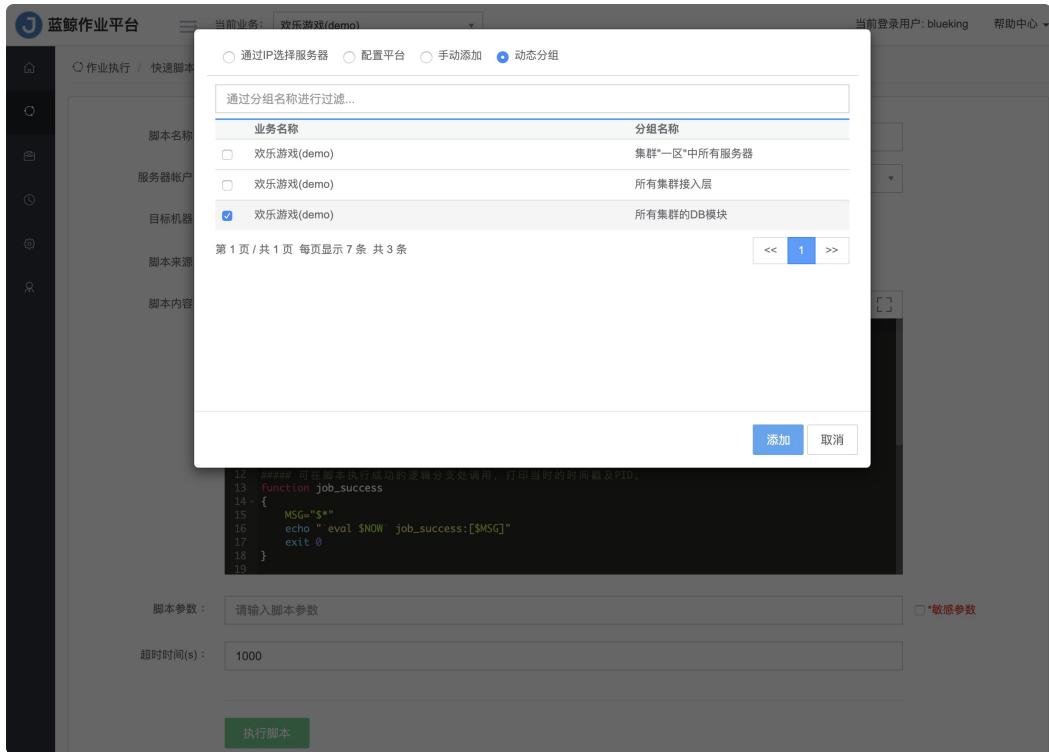
The screenshot shows the '故障自愈' (Fault Recovery) interface. At the top, it says '欢乐游戏(demo)'. The main area has two sections: '自愈场景' (Recovery Scenario) and '自愈处理' (Recovery Action). In '自愈场景', there are filters for '告警类型' (Alert Type: [主机监控] 磁盘使用率), '平台' (Platform: 默认全选), and '模块' (Module: 接入层). In '自愈处理', there is a dropdown menu for '自愈套餐' (Recovery Package: 接入层磁盘清理) with a '查看' (View) and '+' (Add) button.

- 平台团队对资源的管控，例如全业务 DBA、AIX SA

以 DBA 为例，需要管控所有 DB，可将运行数据库的主机所属模块的类型设置为 **数据库**，然后通过 **动态分组** 功能查询模块类型为 **数据库** 的主机。

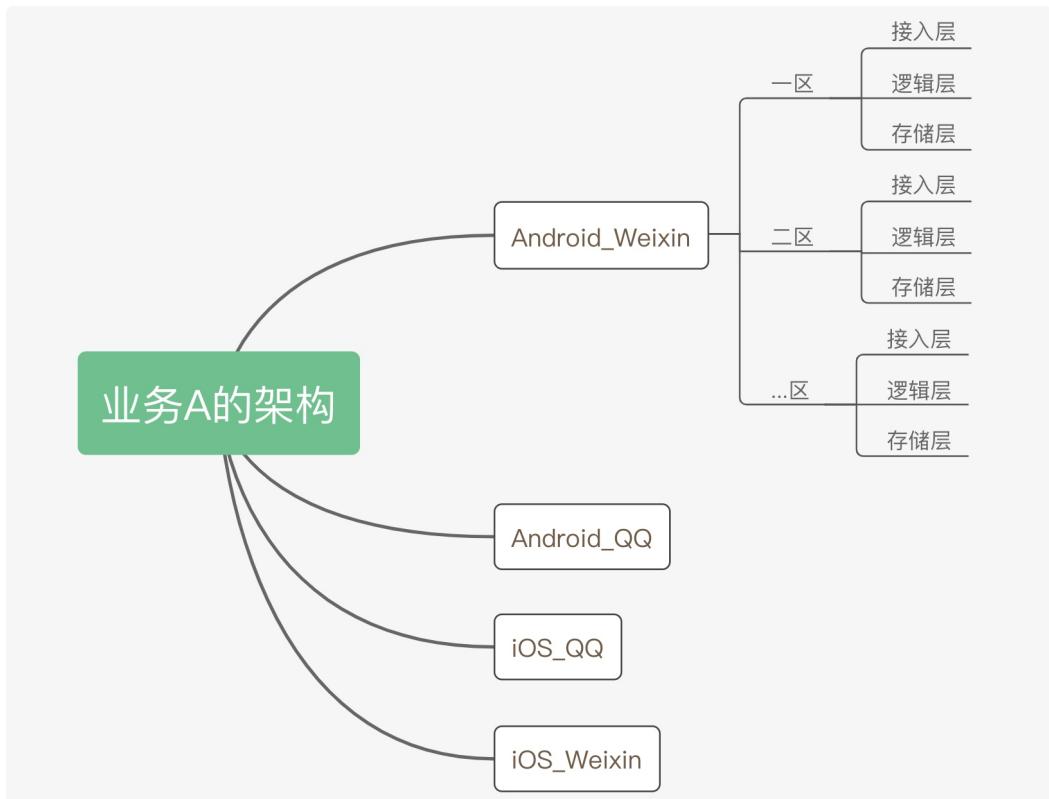
The screenshot shows the '蓝鲸配置平台' (Blue Whale Configuration Platform) with a sidebar for '业务拓扑' (Business Topology) and a main panel for '动态分组' (Dynamic Group). The sidebar has sections for 业务拓扑, 服务模块, 集群层级, 服务分组, 主机自动应用, 动态分组, and 自定义应用. The main panel shows a table for '所有集群的DB模块' (All clusters' DB modules) with columns: 分组名称 (Group Name), ID, 逻辑对象 (Logical Object), 创建用户 (Create User), 创建时间 (Create Time), and 修改人 (Modify User). One entry is shown: '所有集群的DB模块' (ID: f440e587-9776-11eb-a2a0-000c296ac9a0) by '逻辑为' at '2021-04-07 15:58:10'. On the right, there is a '编辑动态分组' (Edit Dynamic Group) panel with tabs for '业务' (Business) and '所有集群的DB模块' (All clusters' DB modules). It includes search fields for '分组名称' (Group Name) and '逻辑名称' (Logical Name), and a '主机组' (Primary Group) section with radio buttons for '逻辑为主机组' (Logical Primary Group) and '集群' (Cluster). There are also buttons for '高级搜索' (Advanced Search) and '继续添加' (Continue to Add).

在 **蓝鲸作业平台** 中可使用该动态分组来选择 DB 主机。

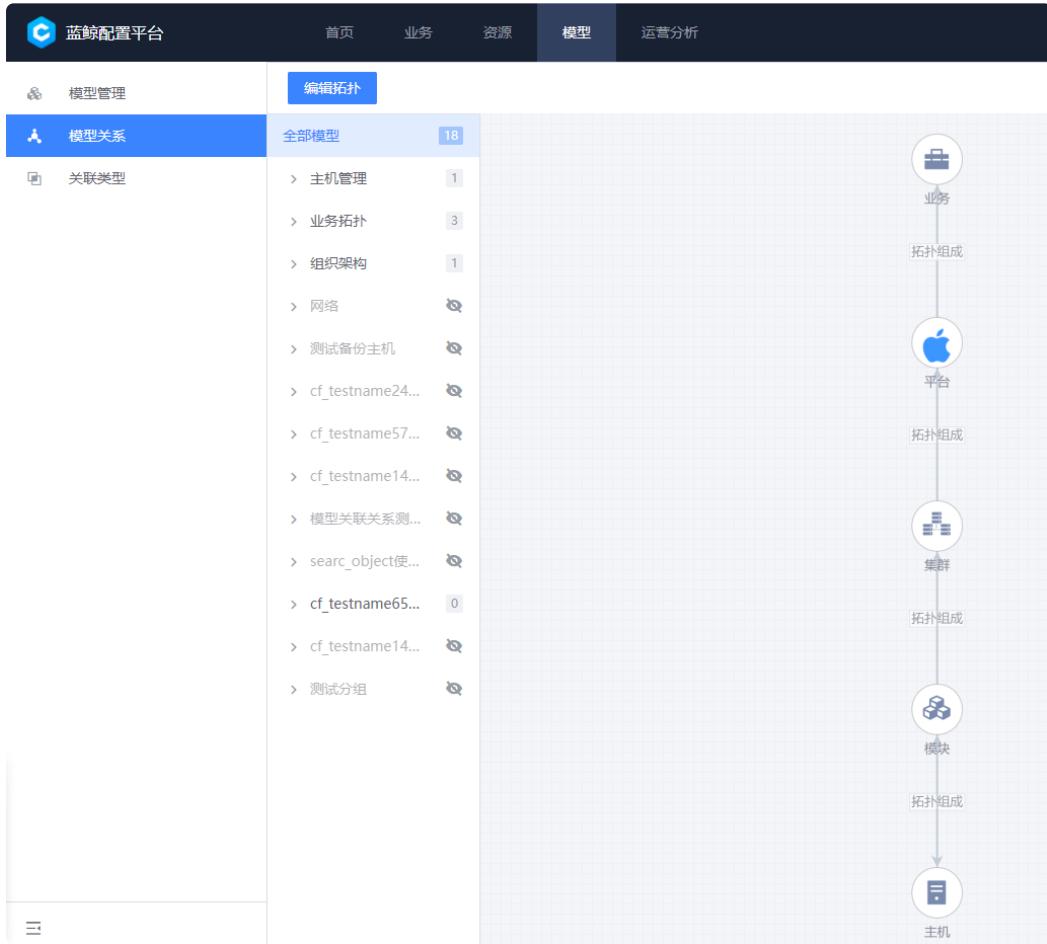


### 三级拓扑不够用，怎么办？新建多级拓扑节点

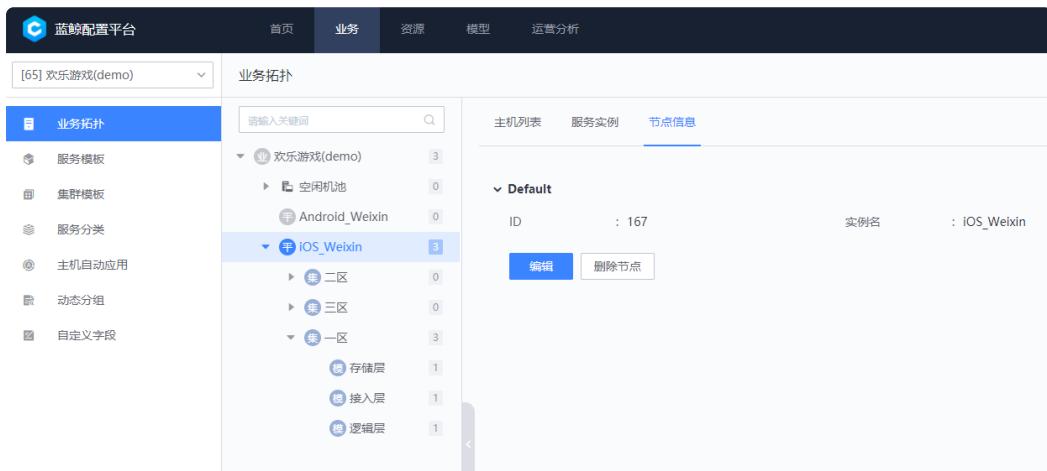
如果业务架构在大区（**集群**）之上还有一级（平台：如 **Android\_Weixin**、**Android\_QQ** 等）



可在【模型-模型关系-编辑拓扑】中，**业务** 与 **集群** 间新建一级或多级拓扑



新的业务拓扑如下：



注：当前版本新增层级拓扑的生效范围是所有业务，在未来的某一个版本中可调整生效范围；

蓝鲸体系在未来的某一个迭代中，将重点弱化业务和细化权限。

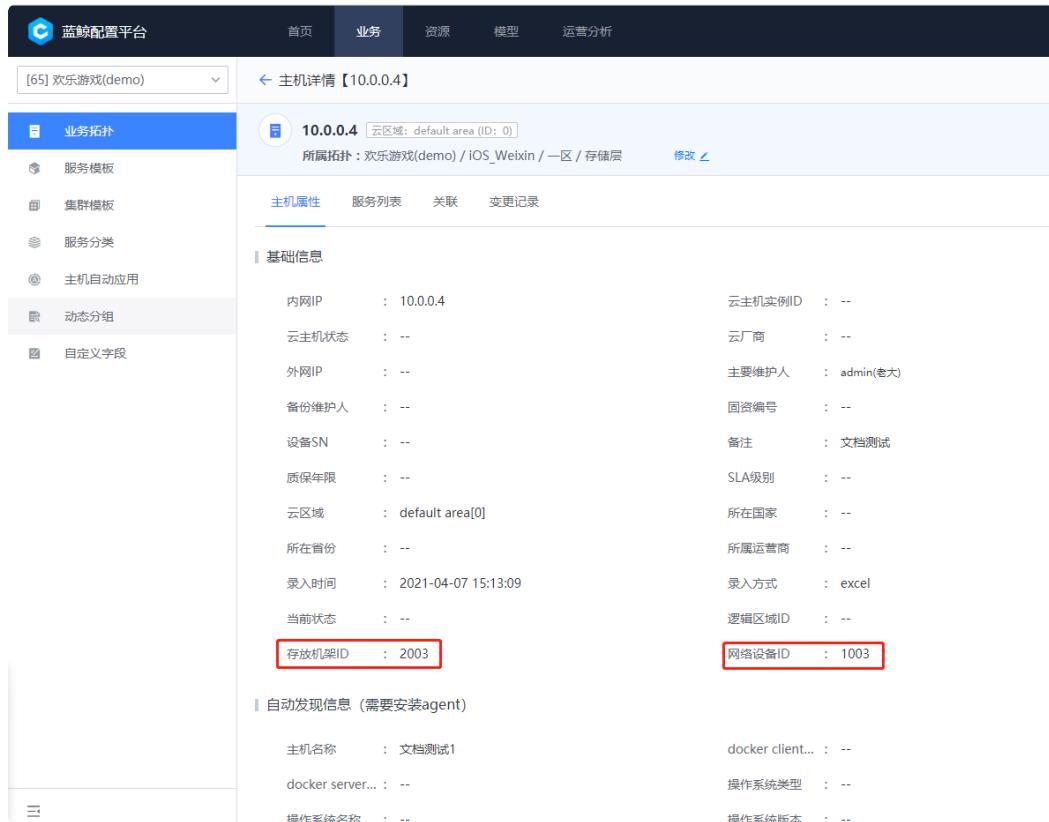
”

## Master DB 和 Slave DB 如何实现物理架构的高可用

两者一定不要放在同一个机架或交换机下，否则机架或交换机掉电，亦或是交换机故障，Master DB 和 Slave DB 都宕机，无法正常切换，导致业务无法对外提供服务。

操作方法：[模型] -> [模型管理] -> [主机] -> [模型字段]，新增 存放机架 ID、网络设备 ID。

此外，配置主机与机架或上联交换机的关联可使用 模型关联，确保数据的唯一性可设置 唯一校验，另外 字段分组 可调整 CI 属性的呈现方式。



基础信息

内网IP	:	10.0.0.4	云主机实例ID	:	--
云主机状态	:	--	云厂商	:	--
外网IP	:	--	主要维护人	:	admin(老大)
备份维护人	:	--	固资编号	:	--
设备SN	:	--	备注	:	文档测试
质保年限	:	--	SLA级别	:	--
云区域	:	default area[0]	所在国家	:	--
所在省份	:	--	所属运营商	:	--
录入时间	:	2021-04-07 15:13:09	录入方式	:	excel
当前状态	:	--	逻辑区域ID	:	--
存放机架ID	:	2003	网络设备ID	:	1003

自动发现信息 (需要安装agent)

主机名称	:	文档测试1	docker client...	:	--
docker server...	:	--	操作系统类型	:	--
操作系统名称	:	--	操作系统版本	:	--

## CMDB 如何管理云资源

随着云计算的发展，上云已经成为了常态，云主机覆盖了不同的厂商，如何才能高效统一地纳管是必不可少的思考；蓝鲸配置平台提供了一套基础云资源同步功能，可以自动同步云资源的管理，以解决更新不及时，信息不准确等问题。

具体步骤为：

1. 创建云账户
2. 创建云资源发现任务

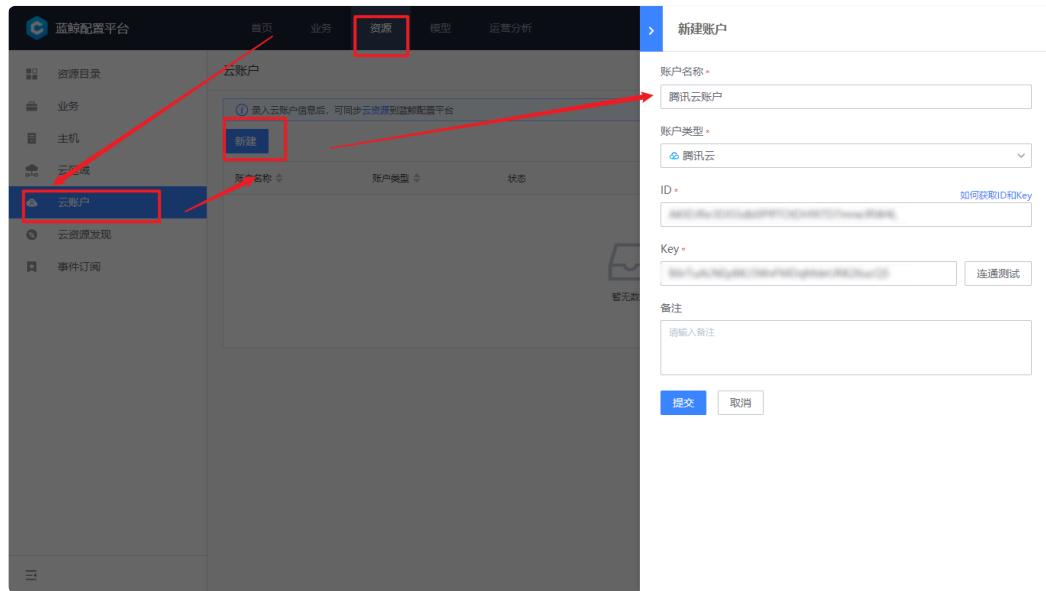
### 3. 分配云资源

## 1. 创建云账户

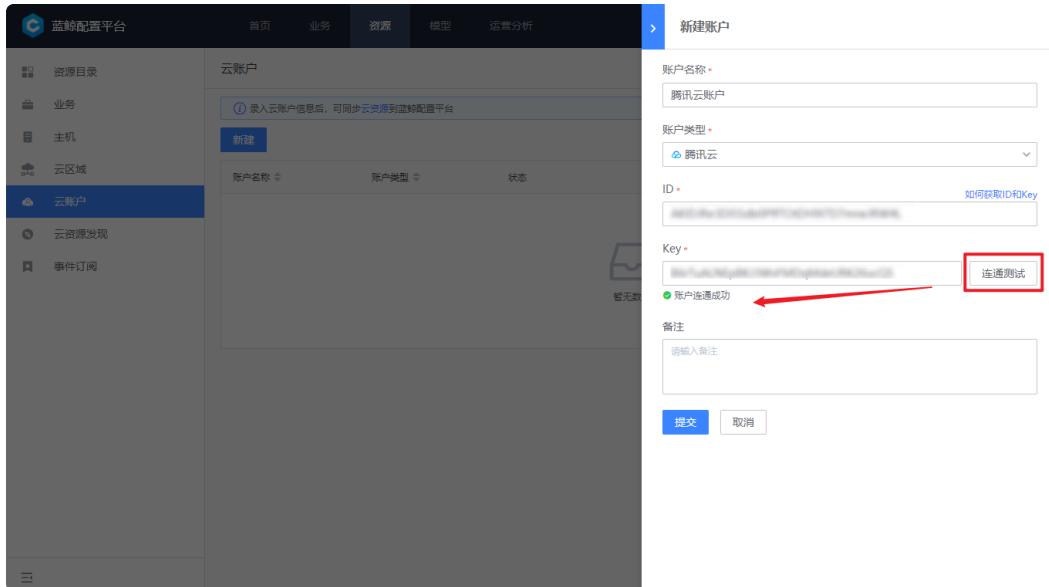
首先需要在公有云申请创建一组可访问所有资源的“子账户”。不同公有云申请方式略有差异。目前配置平台暂支持腾讯云和 AWS 的账号同步能力，相关的账户申请方式可以参考以下文档：

- 腾讯云：<https://cloud.tencent.com/document/product/598/37140>
- AWS：[https://docs.aws.amazon.com/IAM/latest/UserGuide/id\\_roles\\_providers\\_enable-console-custom-url.html](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_providers_enable-console-custom-url.html)

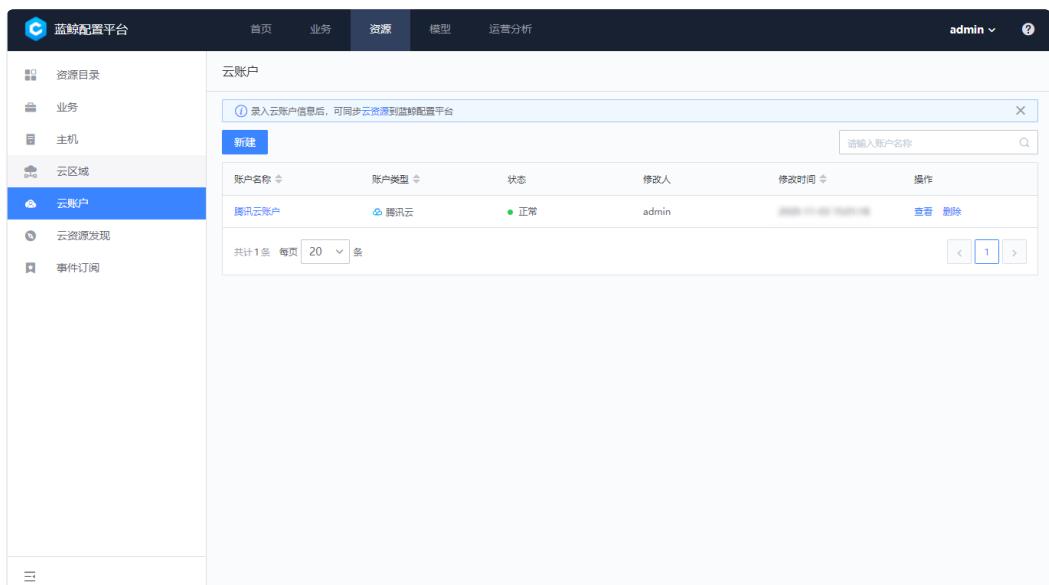
申请创建了账户以后，需要记录访问的 ID 和 KEY，在配置平台中创建账户。



填写了 ID 和 KEY 以后，点击“连通测试”按钮，以确保配置平台可以正常访问公有云的接口地址，同时校验 ID 和 KEY 的可用性。

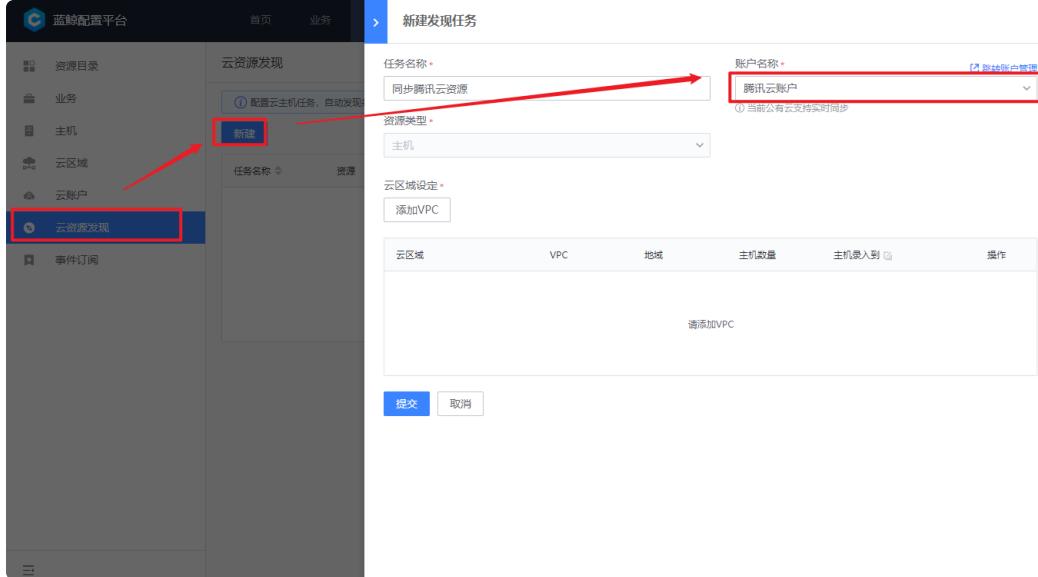


创建账户成功以后，我们接下来可以去创建云资源发现任务。

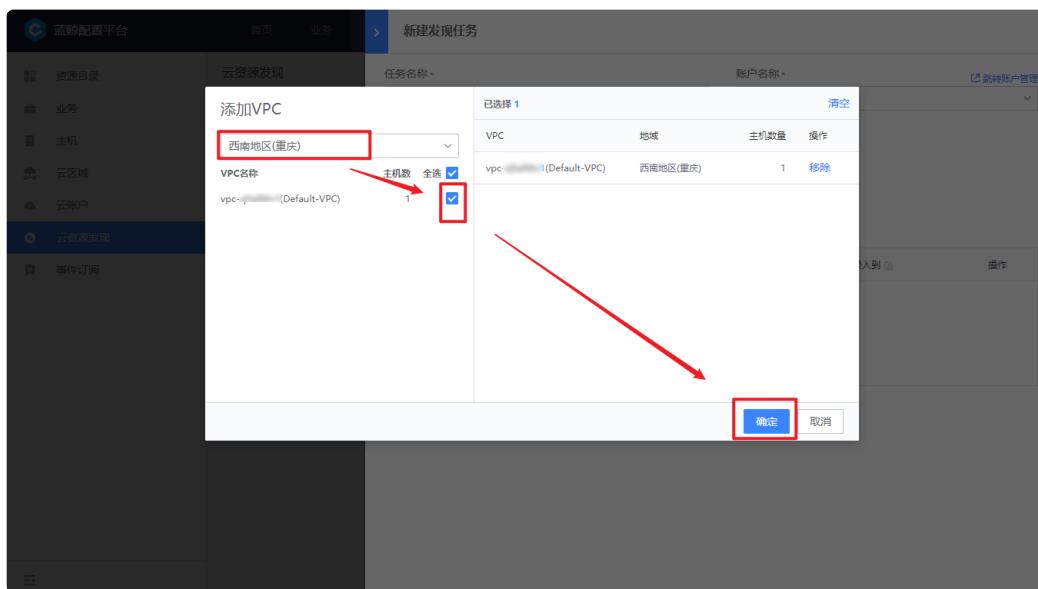


## 2. 创建云资源发现任务

进入“云资源发现”功能，点击“新建”按钮，选择刚刚创建的腾讯云账户。



目前支持按照 VPC 的粒度同步主机资源，通过选择公有云的区域，找到需要同步的 VPC。



在云区域设定中完善设置：

1. 补充 VPC 对应的云区域名称：公有云一个 VPC 内主机 IP 资源不重复，与配置平台中云区域是一一对应的关系。后续此 VPC 下公有云资源都会同步到此 VPC 中
2. 设定云主机录入到资源池的目录：配置平台发现一台新云主机，会优先放置到资源池的目录下，默认在“空闲机”目录。用户也可根据实际的部门或者用途划分，创建一个目录专用于存储新增的云主机资源

新建发现任务

任务名称： 同步腾讯云资源

账户名称： 腾讯云账户

资源类型： 主机

云区域设定：

云区域	VPC	地域	主机数量	主机映入到	操作
<input type="text" value="请输入云区域"/>	vpc-XXXXXX	西南地区(重庆)	1	<input type="text" value="空闲机"/>	<button>删除</button>

提交 取消

创建完成会立刻启动一次同步任务，可以在列表中看到同步状态。

资源目录

业务

主机

云区域

云账户

云资源发现

事件订阅

云资源发现

配置云主机任务，自动发现并同步新增或属性有更改的主机到资源池

新建

任务名称： 同步腾讯云资源

资源： 主机

账户名称： 腾讯云账户

最近同步状态： 成功

最近同步时间： 2020-11-03 15:...

操作人： admin

操作： 编辑 | 删除

共计1条 每页 20 条

### 3. 分配云资源

公有云上的主机资源会被同步到资源池的设定目录中，通过“分配到”功能分配到实际的业务中应用。

# CMDB 如何管理进程

## 情景

应用的存储是 MariaDB，在 CMDB 中注册 MariaDB，以便在监控系统做进程监控。

## 前提条件

在配置平台中**新建业务** 及业务拓扑。

## 步骤

- 服务模板中新建进程
- 服务模板实例同步
- 监控系统自动实现进程端口监控

### 服务模板中新建进程

在业务 -> 服务模板 -> 新建或选择模板中新建进程

- 进程名称：程序的二进制名称：**mysqld**；
- 进程别名：对外显示的服务名：**MariaDB**；

在配置功能名称时，使用命令查询该程序的二进制名称

```
bash $ ps -ef | grep -i mysqld mysql 7800 1 0 7月08 ? 00:00:00 /bin/sh
/usr/bin/mysqld_safe --basedir=/usr mysql 7980 7800 0 7月08 ? 00:01:55
/usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysql --plugin-
dir=/usr/lib64/mysql/plugin --log-error=/var/log/mariadb/mariadb.log --pid-
file=/var/run/mariadb/mariadb.pid --socket=/var/lib/mysql/mysql.sock
```

“

注：监控系统一般完全匹配二进制名称。

”

获取二进制名称的方法：`basename $(readlink -f /proc/7980/exe)`

查询 mysqld 监听的 IP 和端口

```
bash $ netstat -antp | grep mysqld tcp 0 0 10.0.4.29:3306 0.0.0.0:* LISTEN
7980/mysqld
```

- IP: MariaDB 为存储层，一般绑定内网 IP，故选择第一内网 IP。
- Port: 进程监听的端口，`3306`
- Protocol: `TCP`

## 服务模板实例同步

[CMDB 如何管理主机](#) 提到依据业务架构来划分业务拓扑，业务拓扑中模块代表服务，而服务将由一个或多个进程监听的端口来对用户或其他模块提供服务。

所以，需要将进程绑定至对应的模块上，这里将**MariaDB**进程绑定至存储层模块上，点击**服务模板实例**  $\rightarrow$  **批量同步**  $\rightarrow$  确认并同步。

The screenshot shows the 'Service Template Instance Sync' page. On the left, there's a sidebar with navigation items like 'Business Topology', 'Service Template', 'Cluster Management', 'Service Classification', 'Host Application', 'Host Distribution', and 'Customized'. The 'Service Template' item is selected and highlighted in blue. In the main area, there's a 'Service Template Configuration' section with a red box around the 'Service Template Instance' button. Below it is a 'Batch Sync' button with a red arrow pointing to it. A table lists three service instances under the 'Storage Layer' module, each with a sync status column.

模块名称	模块描述	上次同步时间	操作
存储层	欢乐游戏(demo) / iOS_Weixin / 一区 / 存储层	2021-04-07 15:40:03	去同步
存储层	欢乐游戏(demo) / iOS_Weixin / 二区 / 存储层	2021-04-07 14:56:30	去同步
存储层	欢乐游戏(demo) / iOS_Weixin / 三区 / 存储层	2021-04-07 14:56:36	去同步

The screenshot shows the BlueWhale Configuration Platform interface. The top navigation bar includes tabs for 首页 (Home), 业务 (Business), 资源 (Resources), 模型 (Models), and 运营分析 (Operational Analysis). The current page is '业务' (Business). A dropdown menu at the top left shows the project name [65] 欢乐游戏(demo). The main content area has a title '同步模板' (Sync Template) with a back arrow icon. On the left, a sidebar under '服务模板' (Service Template) lists options: 集群模板 (Cluster Template), 服务分类 (Service Category), 主机自动应用 (Host Auto Application), 动态分组 (Dynamic Group), and 自定义字段 (Custom Field). The right side displays a summary of what will be synchronized: '服务分类变更' (Service Category Change) and '变更内容' (Change Content). It lists three MySQL instances across three zones: '欢乐游戏(demo) / iOS\_Weixin / 一区 / 存储层 涉及实例', '欢乐游戏(demo) / iOS\_Weixin / 二区 / 存储层 涉及实例', and '欢乐游戏(demo) / iOS\_Weixin / 三区 / 存储层 涉及实例'. At the bottom are two buttons: '确认并同步' (Confirm and Sync) and '取消' (Cancel).

This screenshot shows the 'Service Template' list page. The top navigation bar is identical to the previous one. The sidebar on the left is also the same. The main content area shows a table of service templates. The columns include ID, 模板名称 (Template Name), 服务分类 (Service Category), 进程数量 (Process Count), 已启用模块数 (Enabled Module Count), 修改人 (Modifier), 修改时间 (Modification Time), and 操作 (Operations). One row is highlighted with a red border. At the bottom, there are pagination controls: 共计3条 每页 20 条.

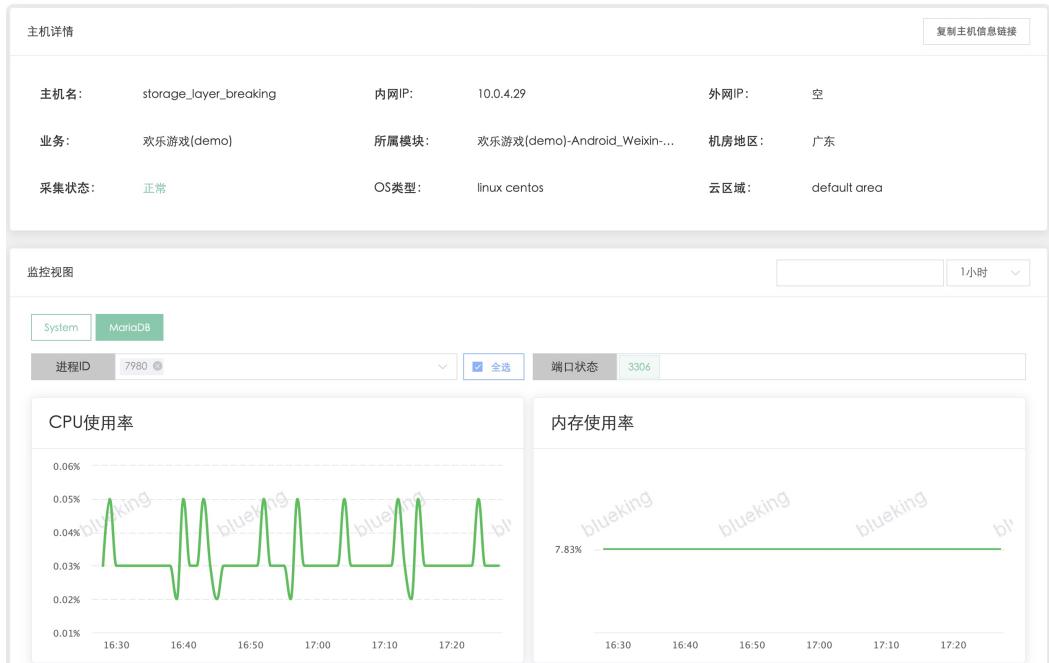
## 监控系统自动实现进程端口监控 给模块 分配主机

This screenshot shows the 'Business Topology' page with a focus on host assignment. The sidebar on the left includes '业务拓扑' (Business Topology), '服务模板' (Service Template), '集群模板' (Cluster Template), '服务分类' (Service Category), '主机自动应用' (Host Auto Application), '动态分组' (Dynamic Group), and '自定义字段' (Custom Field). The main content area has tabs for '主机列表' (Host List), '服务实例' (Service Instance), and '节点信息' (Node Information). Under '服务实例', there is a table for host assignment. The columns are 服务实例 (Service Instance), 进程数量 (Process Count), 端口 (Port), and 操作 (Operations). One entry is highlighted with a red border. At the bottom, there are pagination controls: 共计1条 每页 20 条.

等候一分钟，在蓝鲸自带的监控系统 监控平台 中可以看到 进程的运行情况。

条件过滤						
<input type="checkbox"/> 全选		<input type="checkbox"/> 批量查看		<input type="checkbox"/> 批量采集上报		<input type="checkbox"/> 复制已选IP
						<input type="checkbox"/> 新增告警策略
IP	采集状态	告警事件(今日)	CPU使用率	磁盘IO使用率	CPU5分钟负载	进程服务
10.0.4.102	● 数据未上报	<span style="color: yellow;">0</span> <span style="color: orange;">1</span> <span style="color: red;">0</span>	无查询数据	无查询数据	无查询数据	System
10.0.4.29	● 正常	<span style="color: yellow;">0</span> <span style="color: orange;">0</span> <span style="color: red;">0</span>	1.47%	0.12%	0.01	System MariaDB
10.0.4.128	● 正常	<span style="color: yellow;">0</span> <span style="color: orange;">0</span> <span style="color: red;">0</span>	1.40%	0.49%	0.01	System Nginx

点击 MariaDB 图标，可以查看其占用的 CPU、内存使用率以及文件句柄数等进程占用的资源指标。



## 扩展阅读

### 监控系统消费 CMDB 中进程配置背后的逻辑

当给模块分配完主机后，该主机的大部分 CI 属性将被自动推送至 `/var/lib/gse/host/hostid`

```
json { ..... "process" : [ { "bind_ip" : "3", "bind_modules" : [ 68 ], "bk_func_id" : "", "bk_func_name" : "mysqld", // 对应进程名称 "bk_process_id" : 110, "bk_process_name" : "MariaDB", // 对应进程别名 "bk_start_param_regex" : "", "port" : "3306", "protocol" : "1" } ] }
```

蓝鲸内置的进程监控采集器 `processbeat` 会读取该文件，并写入自身的配置文

件 `/usr/local/gse/plugins/etc/processbeat.conf`，以实现进程和端口的监控。

```
yaml ..... processbeat.processes: - name: mysqld // 二进制名称 displayname: MariaDB protocol: tcp ports: - 3306 paramregex: "" bindip: 10.0.4.29
```

### 二进制名称均为 java，该如何配置

如 ZooKeeper、Hadoop 的二进制均为 **java**，可用进程启动参数区分

进程的功能名称均为 **java**，**启动参数匹配规则** 中分别输入 **zookeeper**、**kafka** 即可。

```
```bash $ ps -ef | grep -i zookeeper

root 5897 1 0 Nov14 ? 01:49:00 /data/bkce/service/java/bin/java -Dzookeeper.log.dir=/data/bkce/logs/zk/ -
Dzookeeper.root.logger=INFO,ROLLINGFILE -
Dzookeeper.DigestAuthenticationProvider.superDigest=bkadmin:1bF5dHUwvnyrhMDaPLkHwFS1JOg= -
cp
/data/bkce/service/zk/bin/..../build/classes:/data/bkce/service/zk/bin/..../build/lib/.jar:/data/bkce/service/zk/bin/..../lib/slf4j-
log4j12-1.6.1.jar:/data/bkce/service/zk/bin/..../lib/slf4j-api-1.6.1.jar:/data/bkce/service/zk/bin/..../lib/netty-
3.10.5.Final.jar:/data/bkce/service/zk/bin/..../lib/log4j-1.2.16.jar:/data/bkce/service/zk/bin/..../lib/jline-
0.9.94.jar:/data/bkce/service/zk/bin/..../zookeeper-
3.4.10.jar:/data/bkce/service/zk/bin/..../src/java/lib/.jar:/data/bkce/etc:/data/bkce/service/zk/conf:/data/bkce/service/java/lib
-Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.local.only=false
org.apache.zookeeper.server.quorum.QuorumPeerMain /data/bkce/etc/zoo.cfg root 10220 1 2 Nov14 ?
13:17:41 /data/bkce/service/java/bin/java -Xmx1G -Xms1G -server -XX:+UseG1GC -
XX:MaxGCPauseMillis=20 -XX:InitiatingHeapOccupancyPercent=35 -XX:+DisableExplicitGC -
Djava.awt.headless=true -Xloggc:/data/bkce/logs/kafka/kafkaServer-gc.log -verbose:gc -
XX:+PrintGCDetails -XX:+PrintGCDateStamps -XX:+PrintGCTimeStamps -
Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.authenticate=false -
Dcom.sun.management.jmxremote.ssl=false -Dkafka.logs.dir=/data/bkce/logs/kafka -
Dlog4j.configuration=file:../../config/log4j.properties -cp
/data/bkce/service/java/lib:/data/bkce/service/kafka/bin/..../libs/aopalliance-repackaged-2.5.0-
b05.jar:/data/bkce/service/kafka/bin/..../libs/argparse4j-0.7.0.jar:/data/bkce/service/kafka/bin/..../libs/connect-
api-0.10.2.0.jar:/data/bkce/service/kafka/bin/..../libs/connect-file-
0.10.2.0.jar:/data/bkce/service/kafka/bin/..../libs/connect-json-
0.10.2.0.jar:/data/bkce/service/kafka/bin/..../libs/connect-runtime-
0.10.2.0.jar:/data/bkce/service/kafka/bin/..../libs/connect-transforms-
0.10.2.0.jar:/data/bkce/service/kafka/bin/..../libs/guava-18.0.jar:/data/bkce/service/kafka/bin/..../libs/hk2-api-
2.5.0-b05.jar:/data/bkce/service/kafka/bin/..../libs/hk2-locator-2.5.0-
b05.jar:/data/bkce/service/kafka/bin/..../libs/hk2-utils-2.5.0-b05.jar:/data/bkce/service/kafka/bin/..../libs/jackson-
annotations-2.8.0.jar:/data/bkce/service/kafka/bin/..../libs/jackson-annotations-
2.8.5.jar:/data/bkce/service/kafka/bin/..../libs/jackson-core-
2.8.5.jar:/data/bkce/service/kafka/bin/..../libs/jackson-databind-
2.8.5.jar:/data/bkce/service/kafka/bin/..../libs/jackson-jaxrs-base-
2.8.5.jar:/data/bkce/service/kafka/bin/..../libs/jackson-jaxrs-json-provider-
2.8.5.jar:/data/bkce/service/kafka/bin/..../libs/jackson-module-jaxb-annotations-
2.8.5.jar:/data/bkce/service/kafka/bin/..../libs/javassist-3.20.0-
GA.jar:/data/bkce/service/kafka/bin/..../libs/javax.annotation-api-
1.2.jar:/data/bkce/service/kafka/bin/..../libs/javax.inject-1.jar:/data/bkce/service/kafka/bin/..../libs/javax.inject-
2.5.0-b05.jar:/data/bkce/service/kafka/bin/..../libs/javax.servlet-api-
3.1.0.jar:/data/bkce/service/kafka/bin/..../libs/javax.ws.rs-api-
```

2.0.1.jar:/data/bkce/service/kafka/bin/..libs/jersey-client-2.24.jar:/data/bkce/service/kafka/bin/..libs/jersey-common-2.24.jar:/data/bkce/service/kafka/bin/..libs/jersey-container-servlet-2.24.jar:/data/bkce/service/kafka/bin/..libs/jersey-container-servlet-core-2.24.jar:/data/bkce/service/kafka/bin/..libs/jersey-guava-2.24.jar:/data/bkce/service/kafka/bin/..libs/jersey-media-jaxb-2.24.jar:/data/bkce/service/kafka/bin/..libs/jersey-server-2.24.jar:/data/bkce/service/kafka/bin/..libs/jetty-continuation-9.2.15.v20160210.jar:/data/bkce/service/kafka/bin/..libs/jetty-http-9.2.15.v20160210.jar:/data/bkce/service/kafka/bin/..libs/jetty-io-9.2.15.v20160210.jar:/data/bkce/service/kafka/bin/..libs/jetty-security-9.2.15.v20160210.jar:/data/bkce/service/kafka/bin/..libs/jetty-server-9.2.15.v20160210.jar:/data/bkce/service/kafka/bin/..libs/jetty-servlet-9.2.15.v20160210.jar:/data/bkce/service/kafka/bin/..libs/jetty-servlets-9.2.15.v20160210.jar:/data/bkce/service/kafka/bin/..libs/jetty-util-9.2.15.v20160210.jar:/data/bkce/service/kafka/bin/..libs/jopt-simple-5.0.3.jar:/data/bkce/service/kafka/bin/..libs/kafka\_2.12-0.10.2.0.jar:/data/bkce/service/kafka/bin/..libs/kafka\_2.12-0.10.2.0-test-sources.jar:/data/bkce/service/kafka/bin/..libs/kafka-clients-0.10.2.0.jar:/data/bkce/service/kafka/bin/..libs/kafka-log4j-appendер-0.10.2.0.jar:/data/bkce/service/kafka/bin/..libs/kafka-streams-0.10.2.0.jar:/data/bkce/service/kafka/bin/..libs/kafka-streams-examples-0.10.2.0.jar:/data/bkce/service/kafka/bin/..libs/kafka-tools-0.10.2.0.jar:/data/bkce/service/kafka/bin/..libs/log4j-1.2.17.jar:/data/bkce/service/kafka/bin/..libs/lz4-1.3.0.jar:/data/bkce/service/kafka/bin/..libs/metrics-core-2.2.0.jar:/data/bkce/service/kafka/bin/..libs/osgi-resource-locator-1.0.1.jar:/data/bkce/service/kafka/bin/..libs/reflections-0.9.10.jar:/data/bkce/service/kafka/bin/..libs/rocksdbjni-5.0.1.jar:/data/bkce/service/kafka/bin/..libs/scala-library-2.12.1.jar:/data/bkce/service/kafka/bin/..libs/scala-parser-combinators\_2.12-1.0.4.jar:/data/bkce/service/kafka/bin/..libs/slf4j-api-1.7.21.jar:/data/bkce/service/kafka/bin/..libs/slf4j-log4j12-1.7.21.jar:/data/bkce/service/kafka/bin/..libs/snappy-java-1.1.2.6.jar:/data/bkce/service/kafka/bin/..libs/validation-api-1.1.0.Final.jar:/data/bkce/service/kafka/bin/..libs/zkclient-0.10.jar:/data/bkce/service/kafka/bin/..libs/zookeeper-3.4.9.jar kafka.Kafka ..config/server.properties

```
ps -ef | grep -i kafka root 10220 1 2 Nov14 ? 13:17:44 /data/bkce/service/java/bin/java -Xmx1G -Xms1G -server -XX:+UseG1GC -XX:MaxGCPauseMillis=20 -XX:InitiatingHeapOccupancyPercent=35 -XX:+DisableExplicitGC -Djava.awt.headless=true -Xloggc:/data/bkce/logs/kafka/kafkaServer-gc.log -verbose:gc -XX:+PrintGCDetails -XX:+PrintGCDateStamps -XX:+PrintGCTimeStamps -Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.authenticate=false -Dcom.sun.management.jmxremote.ssl=false -Dkafka.logs.dir=/data/bkce/logs/kafka -Dlog4j.configuration=file:./config/log4j.properties -cp /data/bkce/service/java/lib:/data/bkce/service/kafka/bin/..libs/aopalliance-repackaged-2.5.0-b05.jar:/data/bkce/service/kafka/bin/..libs/argparse4j-0.7.0.jar:/data/bkce/service/kafka/bin/..libs/connect-api-0.10.2.0.jar:/data/bkce/service/kafka/bin/..libs/connect-file-0.10.2.0.jar:/data/bkce/service/kafka/bin/..libs/connect-json-
```

0.10.2.0.jar:/data/bkce/service/kafka/bin/..libs/connect-runtime-  
0.10.2.0.jar:/data/bkce/service/kafka/bin/..libs/connect-transforms-  
0.10.2.0.jar:/data/bkce/service/kafka/bin/..libs/guava-18.0.jar:/data/bkce/service/kafka/bin/..libs/hk2-api-  
2.5.0-b05.jar:/data/bkce/service/kafka/bin/..libs/hk2-locator-2.5.0-  
b05.jar:/data/bkce/service/kafka/bin/..libs/hk2-utils-2.5.0-b05.jar:/data/bkce/service/kafka/bin/..libs/jackson-  
annotations-2.8.0.jar:/data/bkce/service/kafka/bin/..libs/jackson-annotations-  
2.8.5.jar:/data/bkce/service/kafka/bin/..libs/jackson-core-  
2.8.5.jar:/data/bkce/service/kafka/bin/..libs/jackson-databind-  
2.8.5.jar:/data/bkce/service/kafka/bin/..libs/jackson-jaxrs-base-  
2.8.5.jar:/data/bkce/service/kafka/bin/..libs/jackson-jaxrs-json-provider-  
2.8.5.jar:/data/bkce/service/kafka/bin/..libs/jackson-module-jaxb-annotations-  
2.8.5.jar:/data/bkce/service/kafka/bin/..libs/javassist-3.20.0-  
GA.jar:/data/bkce/service/kafka/bin/..libs/javax.annotation-api-  
1.2.jar:/data/bkce/service/kafka/bin/..libs/javax.inject-1.jar:/data/bkce/service/kafka/bin/..libs/javax.inject-  
2.5.0-b05.jar:/data/bkce/service/kafka/bin/..libs/javax.servlet-api-  
3.1.0.jar:/data/bkce/service/kafka/bin/..libs/javax.ws.rs-api-  
2.0.1.jar:/data/bkce/service/kafka/bin/..libs/jersey-client-2.24.jar:/data/bkce/service/kafka/bin/..libs/jersey-  
common-2.24.jar:/data/bkce/service/kafka/bin/..libs/jersey-container-servlet-  
2.24.jar:/data/bkce/service/kafka/bin/..libs/jersey-container-servlet-core-  
2.24.jar:/data/bkce/service/kafka/bin/..libs/jersey-guava-2.24.jar:/data/bkce/service/kafka/bin/..libs/jersey-  
media-jaxb-2.24.jar:/data/bkce/service/kafka/bin/..libs/jersey-server-  
2.24.jar:/data/bkce/service/kafka/bin/..libs/jetty-continuation-  
9.2.15.v20160210.jar:/data/bkce/service/kafka/bin/..libs/jetty-http-  
9.2.15.v20160210.jar:/data/bkce/service/kafka/bin/..libs/jetty-io-  
9.2.15.v20160210.jar:/data/bkce/service/kafka/bin/..libs/jetty-security-  
9.2.15.v20160210.jar:/data/bkce/service/kafka/bin/..libs/jetty-server-  
9.2.15.v20160210.jar:/data/bkce/service/kafka/bin/..libs/jetty-servlet-  
9.2.15.v20160210.jar:/data/bkce/service/kafka/bin/..libs/jetty-servlets-  
9.2.15.v20160210.jar:/data/bkce/service/kafka/bin/..libs/jetty-util-  
9.2.15.v20160210.jar:/data/bkce/service/kafka/bin/..libs/jopt-simple-  
5.0.3.jar:/data/bkce/service/kafka/bin/..libs/kafka\_2.12-  
0.10.2.0.jar:/data/bkce/service/kafka/bin/..libs/kafka\_2.12-0.10.2.0-  
sources.jar:/data/bkce/service/kafka/bin/..libs/kafka\_2.12-0.10.2.0-test-  
sources.jar:/data/bkce/service/kafka/bin/..libs/kafka-clients-  
0.10.2.0.jar:/data/bkce/service/kafka/bin/..libs/kafka-log4j-appender-  
0.10.2.0.jar:/data/bkce/service/kafka/bin/..libs/kafka-streams-  
0.10.2.0.jar:/data/bkce/service/kafka/bin/..libs/kafka-streams-examples-  
0.10.2.0.jar:/data/bkce/service/kafka/bin/..libs/kafka-tools-  
0.10.2.0.jar:/data/bkce/service/kafka/bin/..libs/log4j-1.2.17.jar:/data/bkce/service/kafka/bin/..libs/lz4-  
1.3.0.jar:/data/bkce/service/kafka/bin/..libs/metrics-core-2.2.0.jar:/data/bkce/service/kafka/bin/..libs/osgi-  
resource-locator-1.0.1.jar:/data/bkce/service/kafka/bin/..libs/reflections-  
0.9.10.jar:/data/bkce/service/kafka/bin/..libs/rocksdbjni-5.0.1.jar:/data/bkce/service/kafka/bin/..libs/scala-  
library-2.12.1.jar:/data/bkce/service/kafka/bin/..libs/scala-parser-combinators\_2.12-

```
1.0.4.jar:/data/bkce/service/kafka/bin/../libs/slf4j-api-1.7.21.jar:/data/bkce/service/kafka/bin/../libs/slf4j-
log4j12-1.7.21.jar:/data/bkce/service/kafka/bin/../libs/snappy-java-
1.1.2.6.jar:/data/bkce/service/kafka/bin/../libs/validation-api-
1.1.0.Final.jar:/data/bkce/service/kafka/bin/../libs/zkclient-
0.10.jar:/data/bkce/service/kafka/bin/../libs/zookeeper-3.4.9.jar kafka.Kafka ..//config/server.properties ````
```

## CMDB 案例-DB 实例的管理

### 情景

应用使用的存储是 MySQL，为了便于 MySQL 的日常维护（如 SQL 变更），需要在 CMDB 中创建 MySQL CI 对象，录入 MySQL 实例。

“

蓝鲸 CMDB 拥有灵活的 CI 能力，掌握该教程后，可以管理数据库、中间件、硬件等 CI 对象

”

### 前提条件

在配置平台中 **新建业务**，并 定义拓扑及分配主机。

**术语解释 - CI** : (Configuration Items)，资源对象，如 **MySQL**、主机、交易系统、交换机、路由器等 - **CI 属性** : (Configuration Items Attribute)，资源对象的配置属性，如 MySQL CI 属性为实例名、IP、端口、存储引擎、数据库版本等 - **CI 实例** : CI 的实例化，唯一识别一个资源对象，如 MySQL CI 实例为 **gd\_area\_master\_01**

### 操作步骤

- 梳理：梳理 MySQL CI 属性
- 建模：创建 MySQL CI 对象
- 实例化：添加 MySQL 实例

#### 梳理 MySQL CI 属性

根据消费 MySQL 的场景，梳理 MySQL 常见的 CI 属性。

字段分组	唯一标识	名称	字段类型	录入方式	是否唯一	是否必填
	bk_inst_name	实例名	短字符	自动	是	是
	ip_addr	IP地址	短字符	自动	是	是
	port	端口	数字	自动	是	是
	version	数据库版本	短字符	自动		

Default	install_dir	安装路径	长字符	自动		
	dbfile_dir	数据库文件路径	长字符	自动		
	enable_binlog	是否开启binlog	枚举	自动		
	enable_slowlog	是否开启慢查询日志	枚举	自动		
	chart_set	字符集	短字符	自动		
	storage_engine	存储引擎	短字符	自动		
	db_size	数据库大小	数字	自动		
核心参数	innodb_buffer_pool_size	innodb缓存池大小	数字	自动		
	innodb_log_buffer_size	innodb日志缓存大小	数字	自动		
	innodb_flush_log_at_trx_commit	innodb日志磁盘写入策略	短字符	自动		
	thread_cache_size	线程缓存大小	数字	自动		
	query_cache_size	查询缓存大小	数字	自动		
	max_connections	最大连接数	数字	自动		

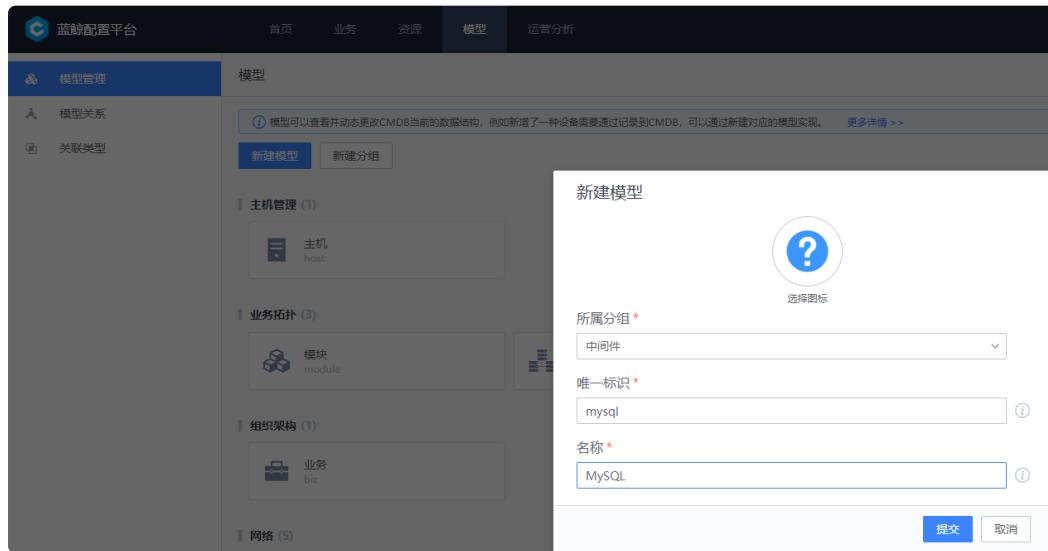
通过 实例名 可以唯一标识一个 MySQL 实例，具体是 IP 和 端口 的组合。

## 创建 MySQL CI 对象

第一步梳理完 MySQL CI 属性后，接下来开始建模：创建 MySQL CI。

### 创建 CI

选择 模型管理 中选择 模型， 创建 MySQL 模型。



### 新增 CI 属性

按照梳理 MySQL CI 属性 中梳理的结果来添加 CI 属性。

如果手工操作繁琐，也可以导入一个 MySQL 模型示例。

## 新增唯一校验

通过 **实例名** 可以唯一标识一个 MySQL 实例，具体是 **IP** 和 **端口** 的组合，所以将 **IP** 和 **端口**作为一个组合校验。

## 设立 CI 关联

针对 MySQL 的管理，除了 CI 属性外，我们同时还关心 MySQL 运行在哪台主机上，所以需要在模型中新建一个“主机上运行 MySQL”的关联。

1 个主机可以运行多个 MySQL 实例，所以 **源到目标的约束条件** 为“1-N”

## 添加 MySQL 实例

完成 MySQL CI 的建模之后，接下来添加 MySQL 实例

### 新增或导入 CI 实例

从 资源 -> 资源目录进入 MySQL 实例列表页

点击新建按钮，按提示添加 MySQL 实例，也可以批量导入 MySQL 实例。

### 创建 CI 实例的关联关系

打开一个 MySQL 实例的详情页，点击 **关联** TAB 中的 **关联管理**，**关联** 当前实例运行在哪台主机上。

再次点击 **关联管理**，可预览 MySQL 实例与主机的关联关系。

如果参照本篇教程将 **机架**、**交换机**、**机房管理单元**、**数据中心** 等 IT 基础设施均录入 CMDB 中，将可以查询一个 MySQL 实例完整的关联关系。



## 混合云设施管控

### 情景

随着云计算浪潮的推进，多云管控逐渐成为趋势，多云间网络无法互通。

接下来看下蓝鲸是如何管控多云主机。

### 前提条件

- 部署完蓝鲸

### 术语解释

- GSE Server**: 蓝鲸管控平台的后台服务，由文件分发、命令执行、数据上报三个模块组成。
- GSE Proxy**: 跨云网络中，GSE Server 和跨云网络中被管控主机的代理，实现对跨云网络主机的管控。
- Agent**: 运行在被管控主机上的代理程序，实现文件分发、命令执行以及数据上报。
- VPC**: (Virtual Private Cloud)，私有网络，逻辑隔离的网络空间，每个私有网络内的服务资源内网互通，不同私有网络之间内网不通，但可通过新建对等连接来连通。
- 云区域**: 标识 VPC 网络，蓝鲸多云管控的关键字段，通过租户 ID( `bk_supplier_id` )、云区域、IP 三者唯一标识主机。
- 直连网络**: 蓝鲸后台服务所在的网络，该网络下管控的主机与蓝鲸后台互通。

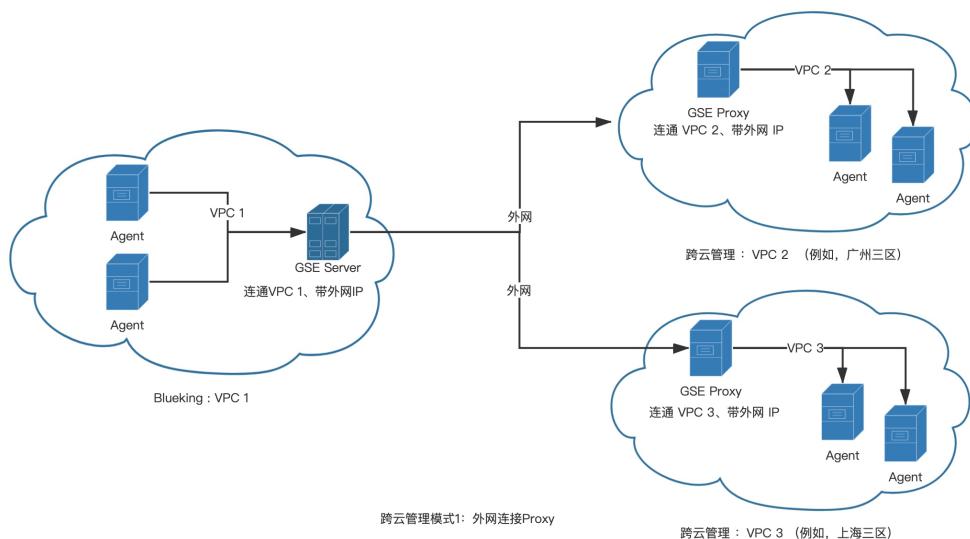
## 操作步骤

- 梳理网络拓扑
- 管理直连网络区域的主机
- 管理跨云网络区域的主机

### 梳理网络拓扑

以下是一个经典的多云管控网络拓扑图。

蓝鲸所在的网络为 **VPC 1**，通过 **直连方式** 管理该网络区域的主机，通过带有外网的 **GSE Proxy** 来管理公有云的主机（网络为 **VPC 2** 和 **VPC 3**）。



“上图是从管控角度绘制的网络拓扑图。

在跨云管控的场景下，在蓝鲸所在 VPC 1 网络下发安装 Agent 的行为，还需要蓝鲸后台的 **Nginx 模块** 具备外网 IP，供 VPC 2 和 VPC 3 网络的 GSE Proxy 下载 Proxy 和 Agent 安装包。

### 管理直连网络区域的主机

先介绍如何管控蓝鲸后台服务所在网络（也称直连网络）的主机。

#### 安装蓝鲸 AGENT（直连区域）

请参考 [节点管理文档：直连区域 安装 Agent](#)

#### 命令执行和分发文件测试

使用作业平台 **执行脚本** 和 **分发文件** 做测试。

The screenshot shows the BlueKing Job Platform interface. On the left, there's a sidebar with options like '作业执行', '快进脚本执行', '快进分发文件', '使用作业执行', '新建作业', '定时作业', '作业管理', '执行历史', and '个人设置'. The main area has tabs for '作业执行 / 快进脚本执行' and '快进脚本执行'. A form on the left contains fields for '脚本名称': '快速执行脚本-20', '服务器端口': 'root', '目标机器': '选择服务器列表', '脚本来源': '手工录入 / 脚本', and '脚本内容' which contains a shell script. On the right, a table lists hosts: 'rnode1' (IP 10.0.5.92, Agent正常), 'VM\_5\_103\_centos' (IP 10.0.5.103, Agent正常), and 'VM\_5\_118\_centos' (IP 10.0.5.118, Agent正常). Below the table are buttons for '全选', '复制已选IP', '复制全部IP', '添加', and '取消'.

## 管理跨云网络区域的主机

接下来介绍，如何管控跨云网络（例如 VPC 2 和 VPC 3）的主机。

### 安装蓝鲸 AGENT（自定义云区域）

请参考 [节点管理文档：自定义云区域 安装 Agent](#)

### 命令执行和分发文件测试

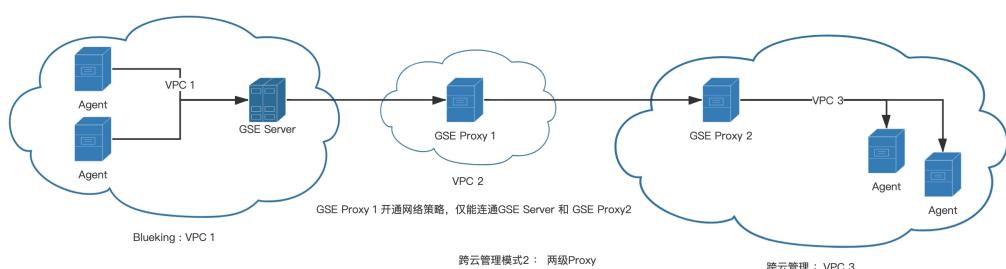
使用作业平台 **执行脚本** 和 **分发文件** 做测试。

```
{% video %}media/bk_nodeman.mp4{% endvideo %}
```

## 扩展阅读

### 多级级联：管理隔离网络的主机

在部分企业网络环境中，存在 **VPC 1（蓝鲸所处网络）** 和 **VPC 3** 不互通，但需要管控 VPC 3 网络主机的场景，如下图：



这时候，需要企业网络环境中，存在一个 VPC 2 的网络，通过申请网络策略，连通 VPC1 和 VPC 3 中的指定（最小策略，安全）服务器。

这时需要使用蓝鲸管控平台的多级级联功能，VPC 3 网络中的主机与 GSE Proxy 2 互通，GSE Proxy 2 通过上联 GSE Proxy 1 与 蓝鲸 GSE Server 通信。

“

多级级联 功能尽请期待。

”

## 作业平台：批量完成多台服务器的文件分发和脚本检查

### 情景

对运行在多台服务器上的业务服务做新版本的文件分发，自动化的批量快速分发、执行脚本检查新版本文件的 md5。

### 前提条件

- 服务器已在 [CMDB 注册](#)
- 拥有服务器所在 CMDB 中业务的运维权限

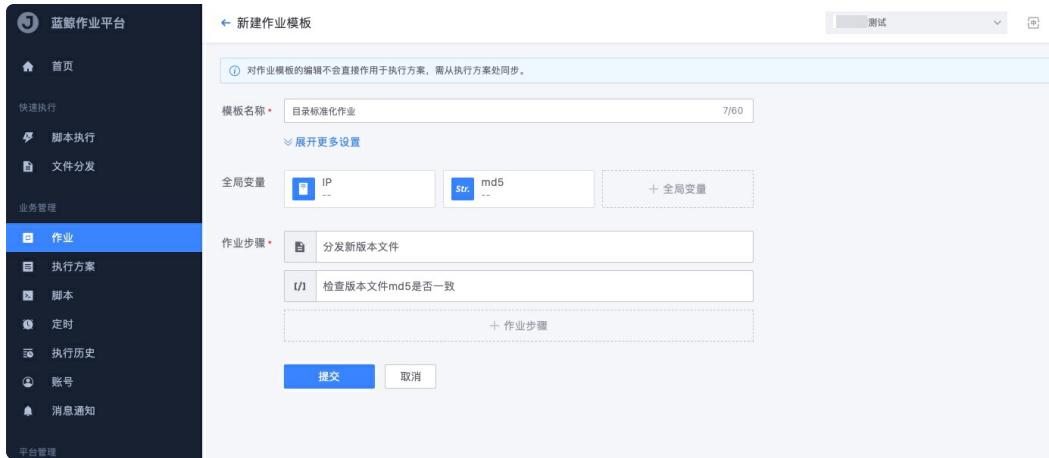
### 操作步骤

- 新建作业
- 执行和查看执行结果

#### 新建作业

按照新版本发布的需求，我们需要将 `newfile.txt` 文件推送至 `/data/` 目录，为了确保万无一失，做 MD5 校验。

作业模板如下：



提示：我们配置两个全局变量，“主机列表”类型变量 IP（用于批量给作业步骤传输主机 ip 信息）和“字符串”类型变量 md5（用于进行新版本 md5 的填参）

IP 变量，可以通过“静态 ip 选择”、“动态拓扑选择”、“动态分组选择”和“手动输入”四种方式进行 ip 选择。

编辑全局变量

变量类型 \*

主机列表

变量名称 \*

IP

初始值

+ 选择主机

变量描述

这里可以备注变量的用途、使用说明等信息 [可选]

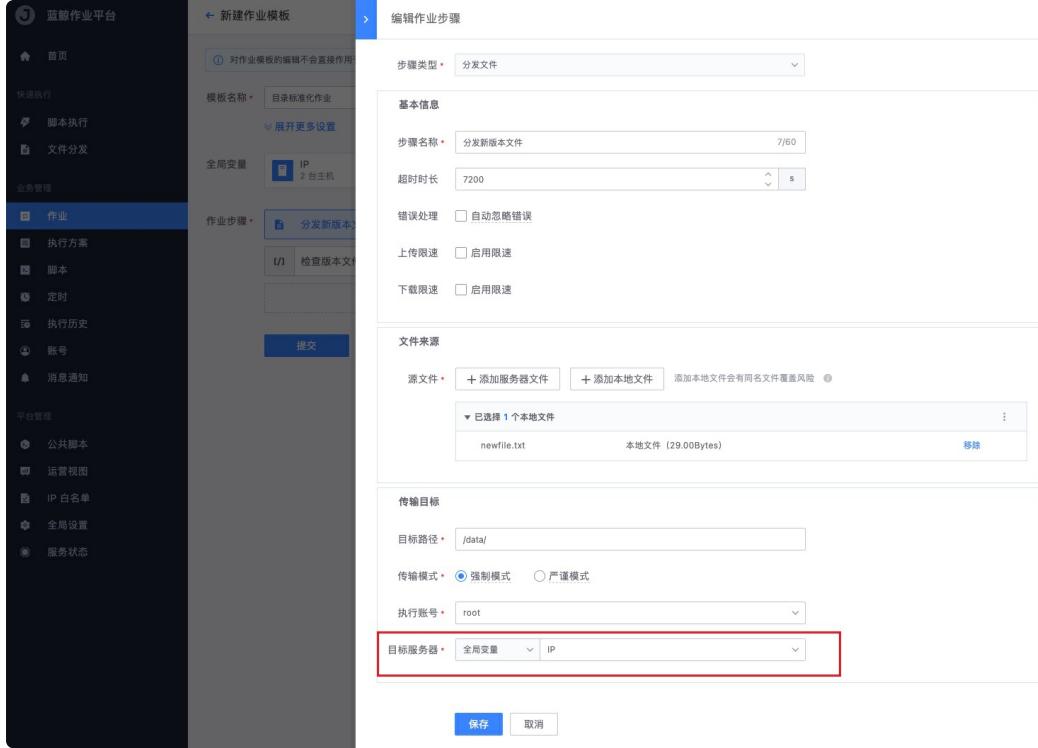
执行时必填

静态 - IP 选择      动态 - 拓扑选择      动态 - 分组选择      手动输入

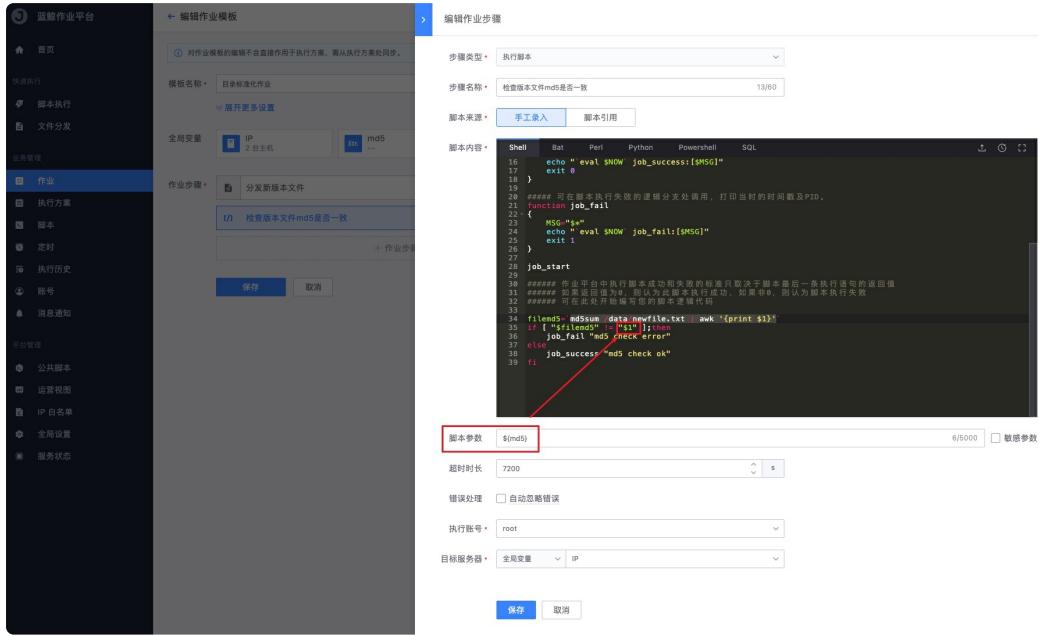
搜索拓扑节点

- 测试 (3)
  - 广东一区 (1)
  - 湖南一区 (1)
    - SVR (1)
    - svrd (1)
  - 广东二区 (1)
  - 空闲机池 (0)

在步骤中通过选择目标服务器——全局变量——IP 的方式进行引用。



md5 变量，可以在脚本参数中直接使用。



## 设置执行方案

保存作业后，需要将作业模板设置为一个执行方案，进行全局变量和执行步骤的选择，使其满足该单一场景的使用。该执行方案即可重复使用。



## 执行作业及查看执行结果

作业执行后，可以查看到执行步骤的进度。



每个步骤点击后，可以查看执行详情。切换步骤，可以切换执行详情。



## 如何实现不同场景的作业编排

当一个操作场景需要多个步骤串联执行时，如果手工一个个去点击执行，那么效率实在太低了！并且，也没办法很好的沉淀下来，方便后续持续使用和维护。

作业平台的作业管理功能很好的解决了这个问题，用户可以在「作业模板」中配置好相应的执行步骤，然后再根据需求场景衍生对应的「执行方案」；

如此，即清晰的区分开作业模板和实例的关系，避免强耦合关系，也便于后续对使用场景的管理和维护。

具体步骤：

## 一、创建作业模板

1. 在 **作业** 页面中，点击 **新建** 进入

image-20201104205455285

1. 按照表单中的要求输入作业的基础信息，和步骤内容

image-20200407170537403

1. 设置一个 **name** 字符串，和 **target** 的主机列表变量，提供作业步骤中使用

创建 **主机列表** 变量：

image-20200407170400057

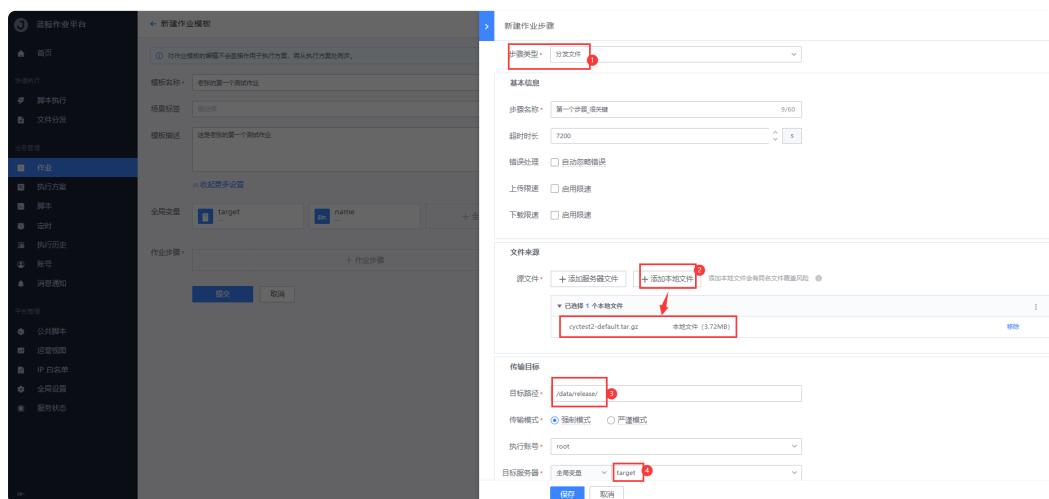
创建 **字符串** 变量：

image-20200407170918316

1. 添加执行步骤，有[执行脚本]、[分发文件]、人工确认]三个类型

这里我们添加三个步骤：

步骤一、分发本地文件



步骤二、脚本执行，直接写临时脚本，也可以引用已有脚本，用上全局变量 **name**

image-20200407171444377

步骤三、加个人工确认步骤，演示一下

image-20200407171536319

1. 最后点击 **提交** 按钮，保存作业模板；至此，作业模板就创建完毕

image-20200407171643379

## 二、创建执行方案

作业「执行方案」是由模板衍生出来的实体对象，不同的执行方案通常用于面向不同的使用场景

1. 进入刚才创建的作业模板，并点击 **选择方案** 前往创建或查看执行方案

image-20200407172705035

1. 点击「新建执行方案」来创建一个全新的作业执行方案

image-20200407172907623

1. 提交保存后，即可在列表中看到刚刚创建的执行方案

image-20200407172941308

## 三、执行作业

1. 从刚才创建的执行方案页面中，点击「去执行」来触发执行动作

image-20200407195149316

1. 执行前，还可以根据自己的需求场景来修改全局变量的值

image-20200407195351347

1. 全局变量的值确认后，点击「执行」即可进入执行总览页面：

image-20200407200026973

1. 点击单个步骤可查看该步骤的执行详情

image-20200407200229158

# 标准运维：一次标准的应用交付自动化案例

## 情景

应用发布是运维这个岗位的职能之一，发布关联多个 ITIL 系统的功能模块，比如发布单、监控的告警屏蔽、DB 变更、业务内公告、统一登录入口等，频繁在多个系统间切换，不但影响效率而且容易出错，同

时无法可视化查看发布进度以及事后的回溯。

接下来，一起看下标准运维是如何解决这些痛点。

## 前提条件

- CMDB 如何管理主机

## 术语解释

- 流程模板**：标准化的资源和应用交付模式，通过资源编排引擎，实现对资源的创建、配置，实现自动化交付资源或应用，行业中一般称之为 **pipeline**、**资源编排模板**，比如一次发布任务可以编排为一个流程模板。
- 标准插件**：多个执行节点通过编排规则实现流程模板，其中的执行节点称之为 **标准插件**，比如 **执行脚本** 为一个标准插件

更多详见 [术语定义](#)

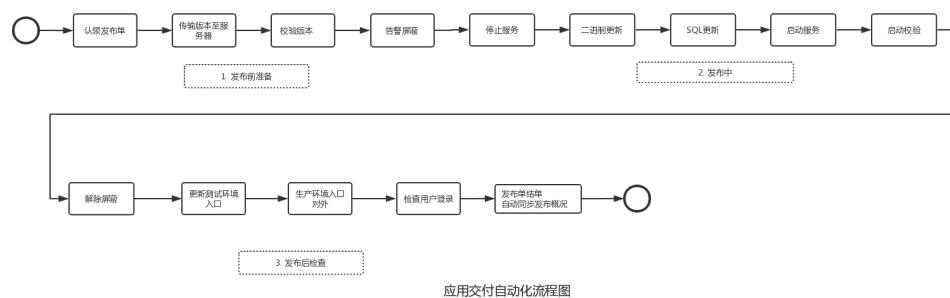
## 操作步骤

- 梳理：梳理标准化模板
- 建模：新建流程模板
- 执行：执行流程

### 梳理标准化模板

配置管理标准化中提到，运维服务“四化建设”的标准化包含配置管理、发布、变更、故障处理、监控告警等场景的流程制定。以发布为例，通过流程图梳理应用交付的流程。

分为发布前准备、发布中、发布后检查三部分。

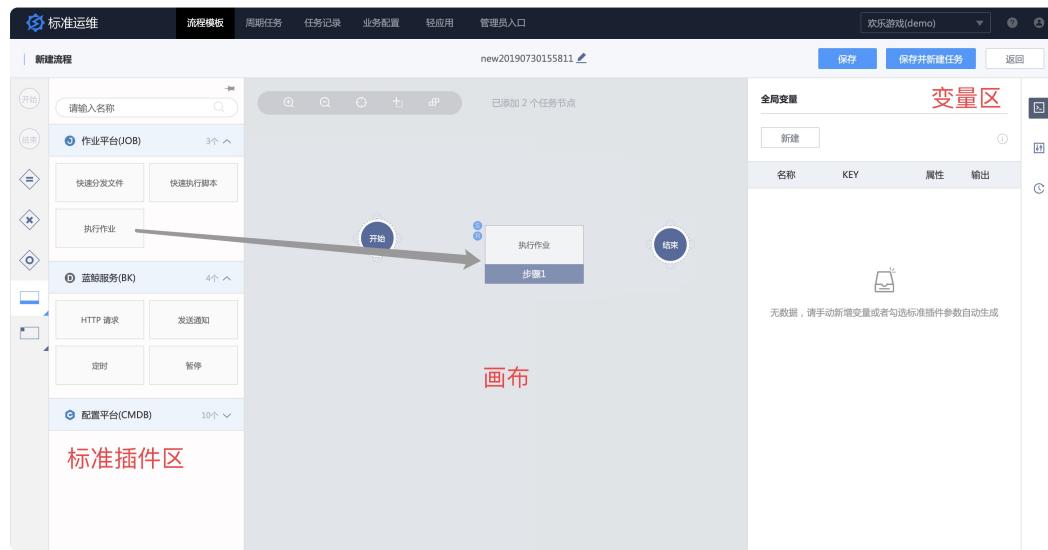


## 创建流程模板

为了简化演示，将流程图中的关键节点在标准运维的业务流程模板中体验出来。

选择 [流程模板] -> [业务流程]，点击 新建 来创建业务流程模板。

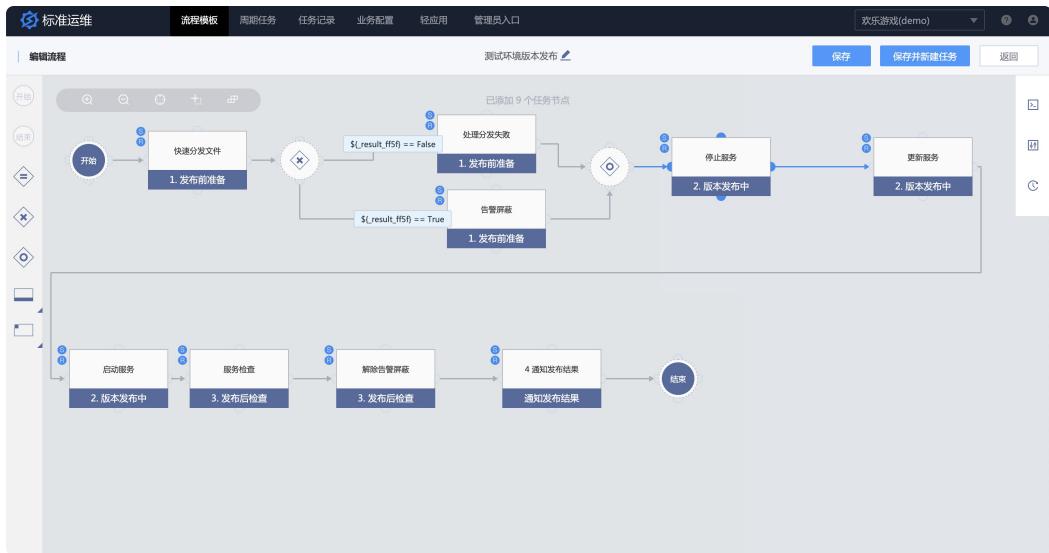
从左侧标准插件区，选择发布流程中需要的标准插件作为流程的节点，比如 执行作业，向右拖动到画布。



选择作业平台中准备好的 作业模板，然后新建 全局变量，并将全局变量填充到节点的参数中。



按照上述步骤，完成一个应用发布的流程模板。



“

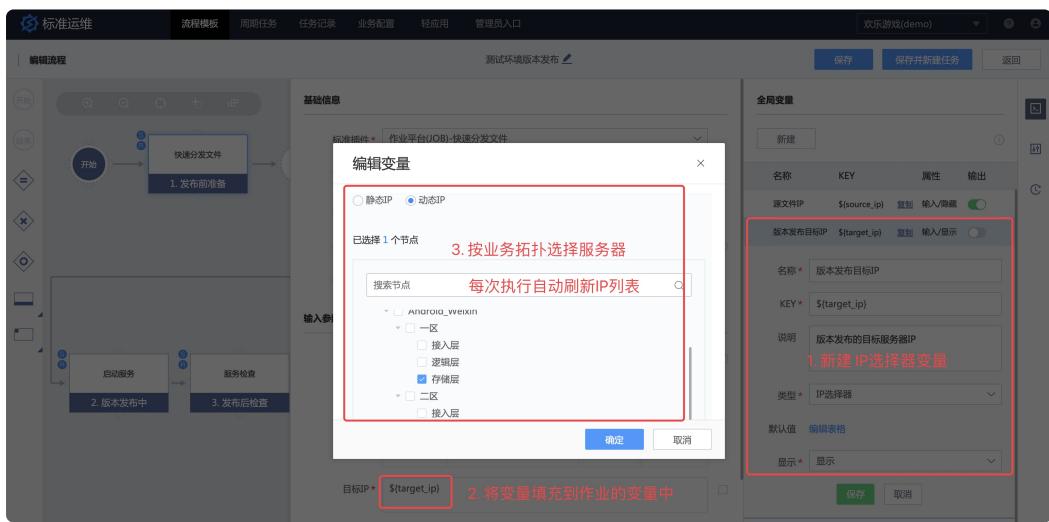
标准运维内置了 CMDB、作业平台、通知等标准组件，如果不在此列的，需要开发标准运维插件来集成企业内部 ITIL 系统。

”

这里重点说明 **全局参数** 和 **流程分支**。

## 全局参数

服务器发生故障后，保障下一次应用发布获取最新的 IP 列表，可以通过 IP 选择器实现。

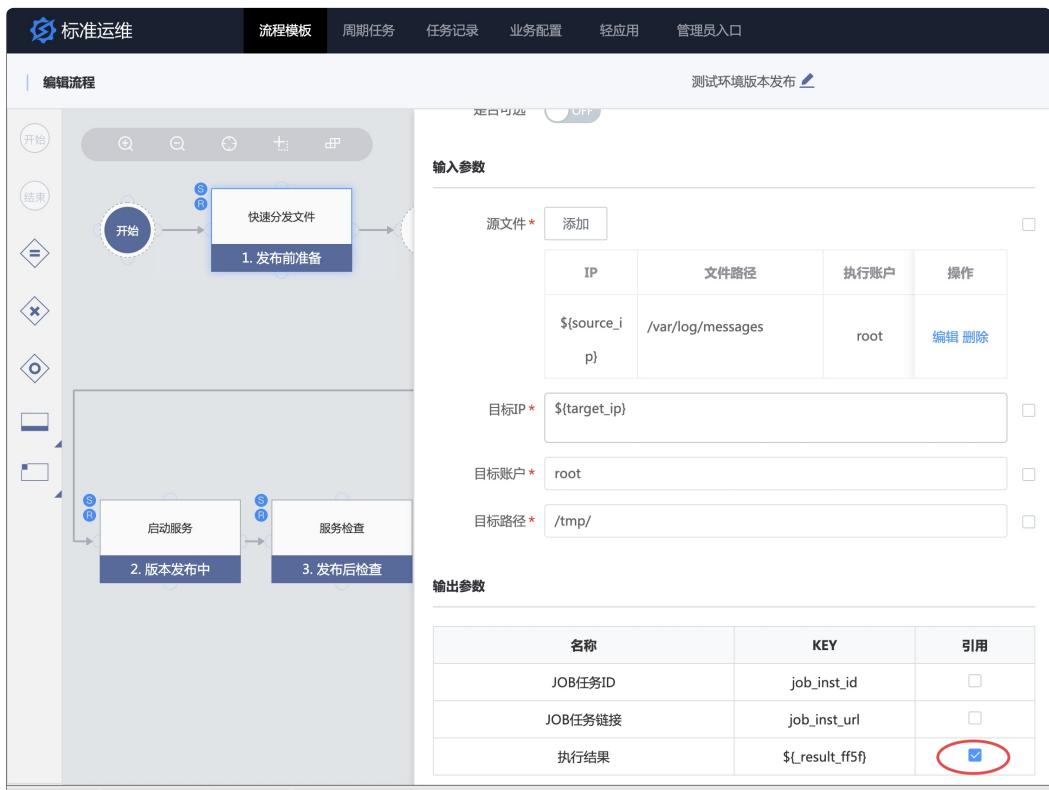


## 流程分支

应用发布过程中，执行成功 和 执行失败的处理分支不同，可以通过流程分支功能对上一步执行结果为真或为假来判断。



提前引用上一步流程节点的输出参数 **执行结果**，将其用于上图中的流程分支表达式。



## 执行流程

在业务流程列表中，点击 **新建任务**

The screenshot shows a list of business processes in a standard operations management system. The columns include ID, Process Name, Category, Update Time, Sub-processes, Creator, and Operations. One row for '测试环境版本发布' has its 'Create Task' button highlighted with a red box.

ID	流程名称	分类	更新时间	子流程更新	创建人	操作
10	[测试自助]修改角色	开发工具	2019-07-23 15:04:29 +0800	--	blueking	<a href="#">新建任务</a> <a href="#">编辑</a> <a href="#">...</a>
9	测试环境版本发布_clone	运维工具	2019-07-01 16:00:53 +0800	--	blueking	<a href="#">新建任务</a> <a href="#">编辑</a> <a href="#">...</a>
7	节点间传参	其它	2019-06-27 17:59:31 +0800	--	blueking	<a href="#">新建任务</a> <a href="#">编辑</a> <a href="#">...</a>
3	测试环境版本发布	运维工具	2019-07-09 20:04:28 +0800	--	breaking	<a href="#">新建任务</a> <a href="#">编辑</a> <a href="#">...</a>

点击执行任务流程

{% video %}media/sops\_execution.mp4{% endvideo %}

## 扩展阅读

### 上下文传参

将一个流程节点的输出作为另一个流程节点的输入。

比如第 1 步输出 MD5 值，第 2 步分发版本，第 3 步使用第 1 步中生成的 MD5 值来校验版本的一致性，效果如下：

The screenshot shows the execution details of a task. It includes a flow diagram with three steps: '开始' (Start), '输出MD5 生成版本' (Generate MD5 and Version), and '分发版本 发布准备' (Distribute Version Preparation). The '执行详情' (Execution Details) section shows the execution information table:

开始时间	结束时间	耗时	失败后跳过	失败自动忽略	手动重试
2019-06-27 18:04:18 +0800	2019-06-27 18:04:21 +0800	2 秒	否	否	否

The '输入参数' (Input Parameters) section shows the parameter mapping:

全局变量	参数名称	参数值	描述
	ip	10.0.4.29	服务器IP
	release_md5	38558e4f46302390ff644f eb29f501a4	原始版本包md5

主要用到标准运维流程节点中的 引用输出参数，引用第 1 步中的 `release_md5` 变量。



`release_md5` 变量需要提前在作业模板中设置，如下图：

作业属性

作业名称: [发布准备] 生成版本包和md5

作业描述: 请输入作业描述

全局IP设置

全局IP: ip

云参

参数名: release\_md5

默认值: 版本md5

脚本来源:

脚本名称: ~

版本号: blueking.2019062717504...

版本备注: v1.0

脚本内容:

```

shell
20 ##### 可在异常执行失败的逻辑分支处调用，打印当时的日期及PID。
21 function job_fail
22 {
23     MSG=$*
24     echo " eval \$NOW` job_fail:[\$MSG]"
25     exit 1
26 }
27
28 job_start
29
30 ##### 可在此处开始编写您的脚本逻辑代码
31 ##### 作业平台中执行脚本成功和失败的标准只取决于脚本最后一条执行语句的返回值
32 ##### 如果返回值为0，则认为此脚本执行成功，如果非0，则认为脚本执行失败
33
34 ## 计算版本md5
35 release_md5=$(md5sum /var/log/messages | awk '{print $1}')
36
37 ## 只是为了在JOB中看结果，实际不需要
38 echo ${release_md5}

```

超时时间: 1000

目标机器: 选择服务器

已被全局变量替代，点击查看！

## 标准运维：操作员做应用交付，让运维更专注

情景

应用交付虽然做到了自动化，但运维仍需和周边团队跟进流程，耗费大量时间，此时通过 **标准运维 职能化** 的能力将应用交付的流程，转移给操作员执行。

运维 **blueking** 提交一个应用发布的职能化任务给到 **caozuo(操作员A)**，**caozuo(操作员A)** 执行任务，跟进任务执行的进展。

## 前提条件

- 在标准运维中 **建立一个流程**
- 准备两个角色：**运维** 和 **职能化(操作员)**

## 操作步骤

- 准备职能化角色
- 运维发起职能化任务流程
- 职能化认领任务

### 准备职能化角色

在 **用户管理** 中新增 **职能化账户**，如 **操作员A**

将其添加至业务的 **操作人员** 角色中

The screenshot shows the BlueKing Configuration Platform's business management interface. The left sidebar has a dark theme with categories like '首页', '基础资源', '业务', '主机', '业务资源', '审计与分析', '权限控制', '模型管理', and '我的收藏'. The '业务' category is currently selected. The main content area shows a list of businesses with columns for 'ID', '业务名', and status. One entry, '欢乐游戏(demo)', is selected. On the right, detailed information for this business is shown, including its name, life cycle status (已上线), time zone (Asia/Shanghai), language (中文), and assigned roles. The '操作人员' role is specifically highlighted with a red box around the value 'caozuo(操作员A)'. At the bottom of the right panel, there are '编辑' (Edit) and '归档' (Archive) buttons.

在标准运维的 **业务流程** 中给 **职能化** 授权

## 运维发起职能化任务流程

运维选择一个应用发布流程，点击 **新建任务**，选择 **职能化任务流程**

## 职能化认领任务

用 **操作员A** 的角色访问标准运维，在 **职能化中心** 中认领刚才新建的 **职能化任务**

如果不修改参数，点击 **认领** 即可

任务名称 \* 测试环境版本发布\_20190709173853

参数信息

版本发布目标IP ①  
版本发布目标IP \* 静态IP ② 动态IP

已选择 1 个节点

搜索节点

- 欢乐游戏(demo)
  - 空闲池
    - 空闲机
    - 故障机
  - Android\_Weixin
    - 一区
      - 逻辑层
      - 接入层
    - 二区
      - 逻辑层
      - 接入层
    - Android\_QQ

筛选条件 ( 同时满足 ) :

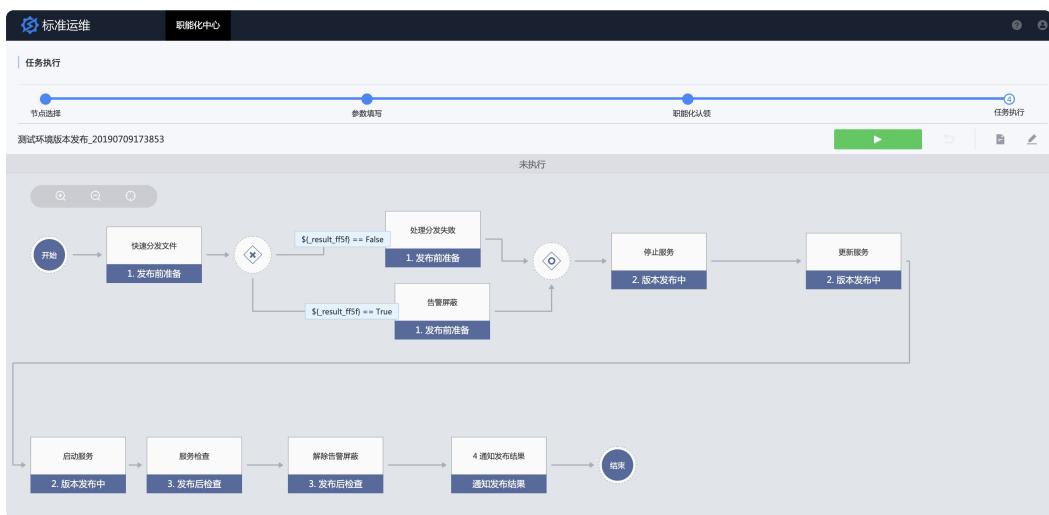
增加一条筛选条件

排除条件 ( 同时满足 ) :

增加一条筛选条件

预览 认领

点击 **执行** 按钮



可以看到任务执行的状态

{% video %}media/stag\_delivery.mp4{% endvideo %}

在 **职能化中心** 可以看到任务已经执行完成。

The screenshot shows a table of tasks with columns: 所属业务 (Business), 任务ID (Task ID), 任务名称 (Task Name), 提单时间 (Create Time), 认领时间 (Accepted Time), 提单人 (Creator), 认领人 (Accepted By), 状态 (Status), and 操作 (Operations). The tasks listed are related to '欢乐游戏(demo)'.

所属业务	任务ID	任务名称	提单时间	认领时间	提单人	认领人	状态	操作
欢乐游戏(demo)	37	测试环境版本发布_20190709200241	2019-07-09 20:04:54 +0800	2019-07-09 20:06:46 +0800	blueking	caozuo	● 完成	查看
欢乐游戏(demo)	36	测试环境版本发布_20190709195912	2019-07-09 20:01:18 +0800	2019-07-09 20:01:35 +0800	blueking	caozuo	● 已执行	查看
欢乐游戏(demo)	35	测试环境版本发布_20190709173853	2019-07-09 17:41:06 +0800	2019-07-09 17:48:47 +0800	blueking	caozuo	● 已执行	查看
欢乐游戏(demo)	34	测试环境版本发布_20190701162612	2019-07-01 16:45:42 +0800	--	zhinenghua	--	● 未认领	认领
欢乐游戏(demo)	33	测试环境版本发布_20190701160313	2019-07-01 16:05:06 +0800	2019-07-01 16:05:12 +0800	zhinenghua	zhinenghua	● 已执行	查看
欢乐游戏(demo)	32	测试环境版本发布_20190701160154	2019-07-01 16:04:02 +0800	2019-07-01 16:04:27 +0800	blueking	zhinenghua	● 已执行	查看

对应运维视角，也可以看到任务已执行完成。

The screenshot shows a table of tasks with columns: ID, 任务名称 (Task Name), 执行开始 (Start Time), 执行结束 (End Time), 任务分类 (Task Category), 创建人 (Creator), 执行人 (Executor), 创建方式 (Creation Method), 状态 (Status), and 操作 (Operations). The tasks listed are related to '欢乐游戏(demo)'.

ID	任务名称	执行开始	执行结束	任务分类	创建人	执行人	创建方式	状态	操作
37	测试环境版本发布_20190709200241	2019-07-09 20:08:09 +0800	2019-07-09 20:08:31 +0800	运维工具	blueking	caozuo	手动	● 完成	克隆 删除
36	测试环境版本发布_20190709195912	2019-07-09 20:01:47 +0800	--	运维工具	blueking	caozuo	手动	● 失败	克隆 删除
35	测试环境版本发布_20190709173853	2019-07-09 19:46:15 +0800	--	运维工具	blueking	caozuo	手动	● 失败	克隆 删除
34	测试环境版本发布_20190701162612	--	--	运维工具	zhinenghua	--	手动	● 未执行	克隆 删除
33	测试环境版本发布_20190701160313	2019-07-01 16:05:17 +0800	--	运维工具	zhinenghua	zhinenghua	手动	● 失败	克隆 删除
32	测试环境版本发布_20190701160154	2019-07-01 16:04:34 +0800	--	运维工具	blueking	zhinenghua	手动	● 失败	克隆 删除

如此，运维可以把日常的应用发布需求提交给操作员执行，释放自己，把精力多放在 **运维服务目录输出** 以及 **业务优化** 上。

## 扩展阅读：设置任务执行者

操作员执行任务流程时，标准运维从 **任务执行者** 字段获取运维人员列表，如果不填，则为该业务的随机运维人员。

The screenshot shows the configuration for task executors. It includes sections for '执行代理人设置' (Agent Executor Settings) and '人员分组设置' (User Group Settings). The '执行代理人' (Agent Executor) field is set to '--'. The '人员分组设置' section shows a table with a single row for '部门' (Department) '研发部' (R&D Department).

序号	分组名称	成员	操作
	研发部	蓝天的数据	

## 流程服务：服务器资源申请流程线上化

### 情景

传统的资源申请是 **通过邮件**，沟通成本太高，而且无法回溯。

新业务上线，需要申请几台服务器用于部署应用。接下来，看蓝鲸流程服务是如何实现服务器申请的流程线上化。

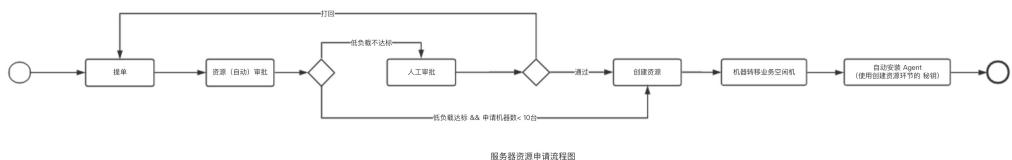
## 前提条件

- 准备 1 个 管理员账号，用于流程设计；1 个运维账号用于资源申请，1 个 普通账号 SA 用于审批和交付资源。

## 操作步骤

- 梳理资源申请流程
- 创建资源申请服务目录及流程
- 一次资源申请示例

### 梳理资源申请流程



流程图中是一个实践案例，部分数据需要从周边系统获取，此处功能需要做二次开发，本教程专注流程本身。

### 创建资源申请服务目录及流程

先设计资源申请的流程，流程依附在服务目录上对外提供服务。

### 角色设置

使用蓝鲸管理员账号给该业务分配运维人员的角色。

业务管理界面展示了“欢乐游戏(demo)”的详细信息。左侧菜单栏包括：首页、基础资源（业务、主机）、业务资源、审计与分析、权限控制、模型管理、我的收藏。右侧显示了“欢乐游戏(demo)”的属性、关联和变更记录。在“属性”标签页下，可以看到基础信息（业务名：欢乐游戏(demo)，时区：Asia/Shanghai）和角色（运维人员：blueking(blueking), ops\_a(运维小A)；测试人员：ceshi(测试)；操作人员：caozuo(操作员A)；产品人员：chanpin(产品)；开发人员：kaifa(开发)）。底部有“编辑”和“归档”按钮。

使用管理员账号新增通用角色 SA，并添加用户 sa\_zhang（该账户是前提条件中准备的）。

No.	角色名	人员	操作
1	服务台小组	admin(管理员)	<a href="#">编辑</a> <a href="#">删除</a>
2	故障派单员	admin(管理员)	<a href="#">编辑</a> <a href="#">删除</a>
3	变更经理	admin(管理员)	<a href="#">编辑</a> <a href="#">删除</a>
4	测试		<a href="#">编辑</a> <a href="#">删除</a>
5	运维		<a href="#">编辑</a> <a href="#">删除</a>
6	运营		<a href="#">编辑</a> <a href="#">删除</a>
7	开发经理		<a href="#">编辑</a> <a href="#">删除</a>
8	开发		<a href="#">编辑</a> <a href="#">删除</a>

## 设计资源申请流程

### 填写流程信息

选择菜单【流程设计】，点击【新增】按钮，按提示填写流程信息。

流程名称：\* 申请服务器

流程说明：\* 业务上线、扩容、变更等场景需要的计算资源。

流程类型：\* 请求

是否关联业务：\*  是  否

资源申请类，一般选择“请求”

流程类型选择【请求】，需要【关联业务】，因为资源申请和业务相关，资源管理员需要知道将资源分配给哪个业务。

点击【下一步】，进入【定义与配置流程】环节。

### 定义与配置流程



点击上图【流程画布】中的【齿轮】，配置【提单】流程节点的字段。

一般是业务的运维提服务器资源申请的单据，所以操作角色选择【CMDB 业务公用角色】->【运维人员】。

The screenshot shows the 'Configure Node' page for the 'Ticket' node. It includes a 'Basic Information' section with fields for 'Node Name' (Ticket) and 'Operation Role' (CMDB business public role, Operations and maintenance personnel). Below is a table for adding fields:

字段名	字段类型	字段值	字段描述	字段校验方式	布局要求	字段属性	操作
标题	单行文本	--	申请N台服务器	必填	半行	自定义	<a href="#">上移</a> <a href="#">下移</a> <a href="#">编辑</a>
用途	单选下拉框	业务上线,业务扩容	申请服务器做什么	必填	整行	自定义	<a href="#">上移</a> <a href="#">下移</a> <a href="#">编辑</a> <a href="#">删除</a>
期望交付时间	日期	--	期望交付时间	必填	整行	自定义	<a href="#">上移</a> <a href="#">下移</a> <a href="#">编辑</a> <a href="#">删除</a>
服务器配置	自定义表格	--	各种机型需要多少	可选	整行	自定义	<a href="#">上移</a> <a href="#">下移</a> <a href="#">编辑</a> <a href="#">删除</a>

点击【新增字段】，参照梳理资源申请流程，新增每个环节中需要的字段。

**基本信息**

节点名称：提单  
操作角色：CMDB业务公用角色

+ 新增字段

字段名	字段类型	字段值	字段描述
标题	单行文本	--	申请N台服务器
用途	单选下拉框	业务上线,业务扩容	申请服务器做什么
期望交付时间	日期	--	期望交付时间
服务器配置	自定义表格	--	各种机型需要多少

字段显示名：  
用途  
字段描述：  
申请服务器做什么  
至少输入5个字 8/100

字段类型：  
 单选下拉框 选择合适的字段类型

数据源：  
 自定义数据  接口数据  数据字典

自定义数据：  
业务上线  
业务扩容

布局要求：  
整行

字段校验方式：  
必填

正则规则表达式：  
无

提交 取消

参照 **梳理资源申请流程**，完成整个服务器申请流程的配置。



### 启用流程

【启用流程】，选择适合的通知策略，点击【提交】完成流程设计。

**设置流程启用管理**

是否启用该流程： ON

**设置任务通知策略**

是否通知： 是  否

是否督办： 是  否

通知频率： 首次通知，以后不再通知  
 首次通知后，次日起每天定时通知

通知类型： SMS短信  邮件  微信

[上一步](#) [提交](#)

### 流程模板实例化

选择菜单【流程设计】，找到刚编辑的申请服务器流程，点击【部署】，生成流程实例。

No.	流程名	说明	类型	更新人	更新时间	状态	操作
1	应用发布	将应用版本发布至生产环境	变更	blueking(blueking)	2019-08-14 16:15:25	启用	<a href="#">预览</a> <a href="#">编辑</a> <a href="#">导出</a> <a href="#">部署</a> <a href="#">删除</a>
2	申请服务器	业务上线、扩容、变更等场景需要的计算资源。	请求	blueking(blueking)	2019-08-14 20:06:01	启用	<a href="#">预览</a> <a href="#">编辑</a> <a href="#">导出</a> <a href="#">部署</a> <a href="#">删除</a>

共计1页

确认部署此流程？

流程部署成功以后，将会产生对应的流程版本

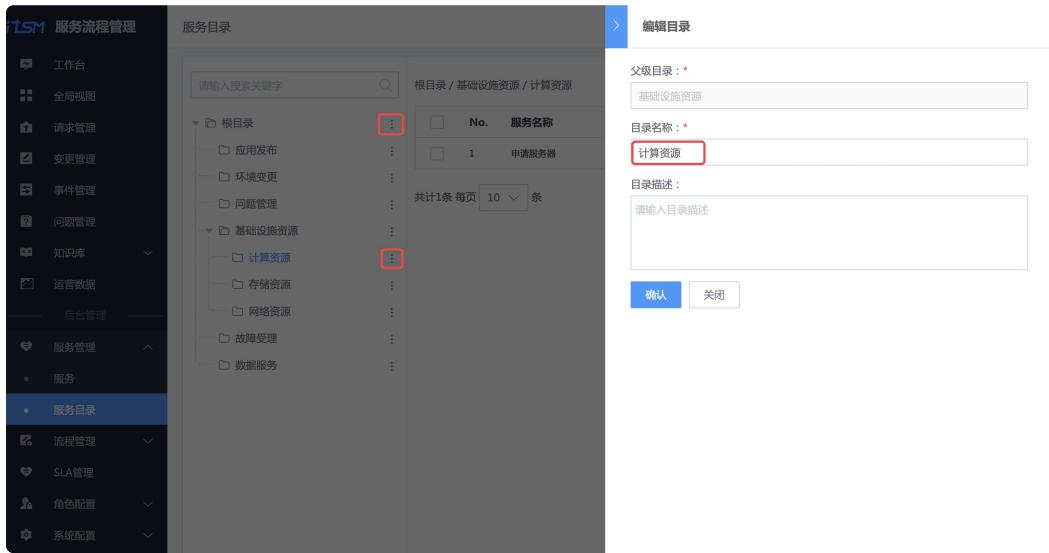
[确定](#) [取消](#)

### 在服务目录中新增 "申请服务器" 服务，并绑定流程

选择菜单【服务】，点击【新增】按钮，新增“申请服务器”服务，并关联刚生成的流程实例。



选择菜单【服务目录】，选中【根目录】，点击右侧【：】，点击【新增】，按提示新增一个名为**基础设施资源**的服务目录，然后在**基础设施资源**下新建一个**计算资源**的服务目录。

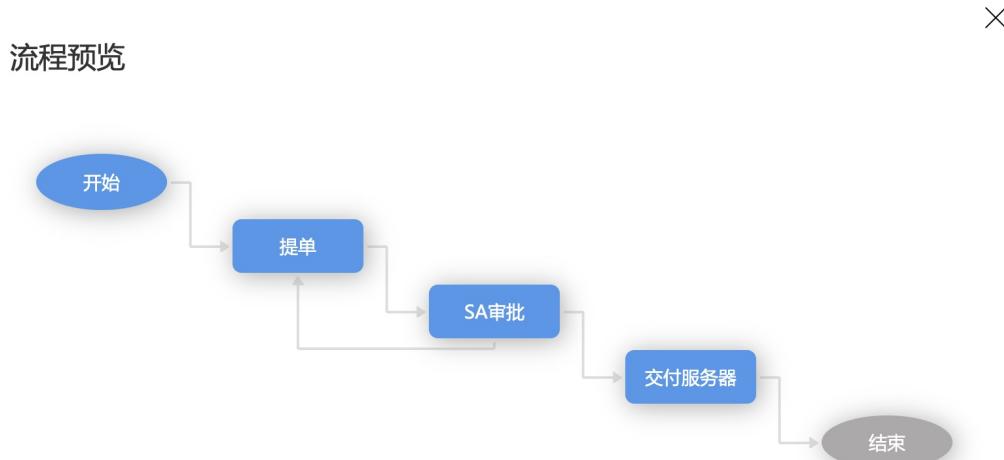


选中刚刚创建的服务目录【计算资源】，右侧会显示【添加】按钮，点击该按钮添加**申请服务器**服务。

No.	服务名称	服务类型
1	申请服务器	请求管理

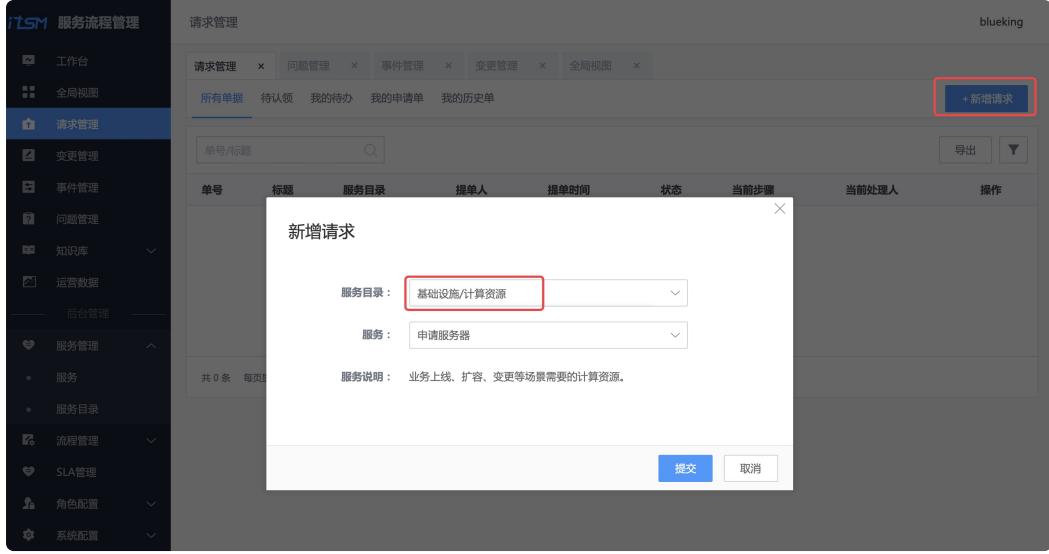
## 一次资源申请示例

创建完“申请服务器”流程及服务目录后，接下来做一次服务器申请演示。



## 运维提单申请服务器

用[运维](#)账号登录流程服务，选择【请求管理】菜单，点击【新增请求】，选择【申请服务器】服务，点击【提交】。



在申请服务器的提单界面，填写服务器申请的关键信息：关联业务、期望交付时间以及服务器的配置和地域。为了安全起见，逻辑层和存储层一般不需要外网 IP。

点击【提交】，创建服务器申请需求。

Machine	Operating System	External IP	Region	Quantity	Action
A1(Calculation): 8G/16G/1T/SAS/NORAILD	CentOS 7.6	Yes	Guangzhou	2	Add Remove
B1(Logical): 8G/64G/1T/SAS/RAID1	CentOS 7.6	No	Guangzhou	2	Add Remove
C2(Storage): 8G/16G/1T/SSD/RAID1	CentOS 7.6	No	Guangzhou	2	Add Remove

注：点击【保存模板】可以记录提单环节的关键信息，下次提单只需修改少部分字段即可。

## SA 审批单据

SA 收到一封待处理的服务器申请邮件。

**【ITSM】请求管理单 【待处理】** ☆

发件人: breaking <breaking@163.com> [回]

时间: 2019年8月14日(星期二) 晚上8:31

收件人: Pythoning <Pythoning@qq.com>

### 【ITSM通知】请求管理

Hi,您好!

您有1条请求管理单 【待处理】

标题: 欢乐游戏业务扩容, 申请6台服务器

单号: NO2019081420252287

服务目录: 基础设施资源->计算资源

当前环节: SA审批

当前处理人: sa\_zhang(SA)

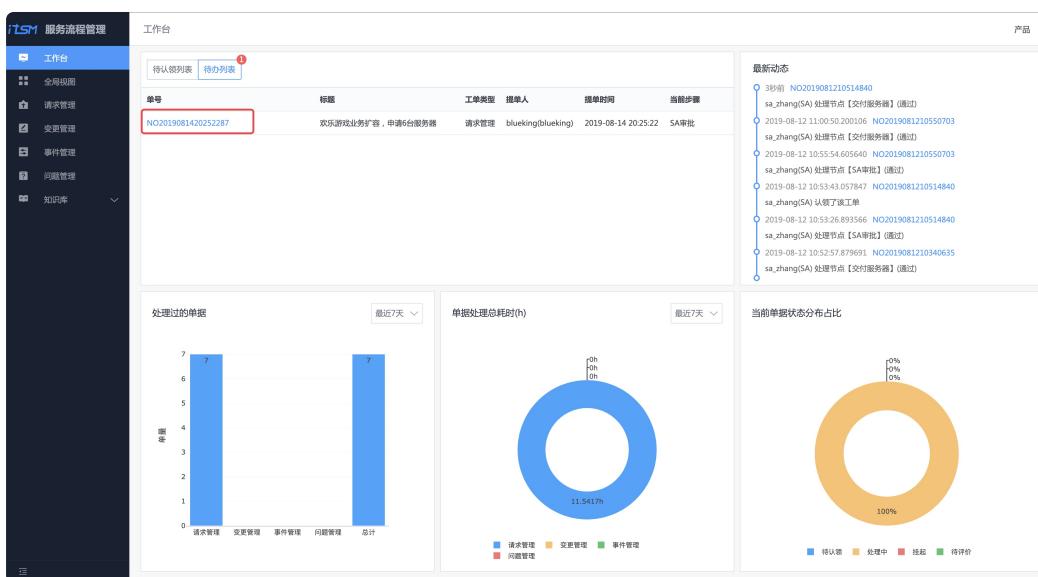
[点击查看详情](#)

如有任何问题, 可随时联系ITSM助手。

ITSM流程管理服务

2019年8月14日

使用 SA 账号登录流程服务, 在待办列表中, 找到一个刚刚运维提交的服务器申请单据。



点击链接, 同时在周边系统查看该业务的低负载指标, 确认达标, 点击【通过】, 完成本环节流程。

“

低负载指标：CPU、内存、网络等指标是否长期处于一个很低的水平，如果是的话，证明计算资源利用率低，SA一般不予通过资源申请。

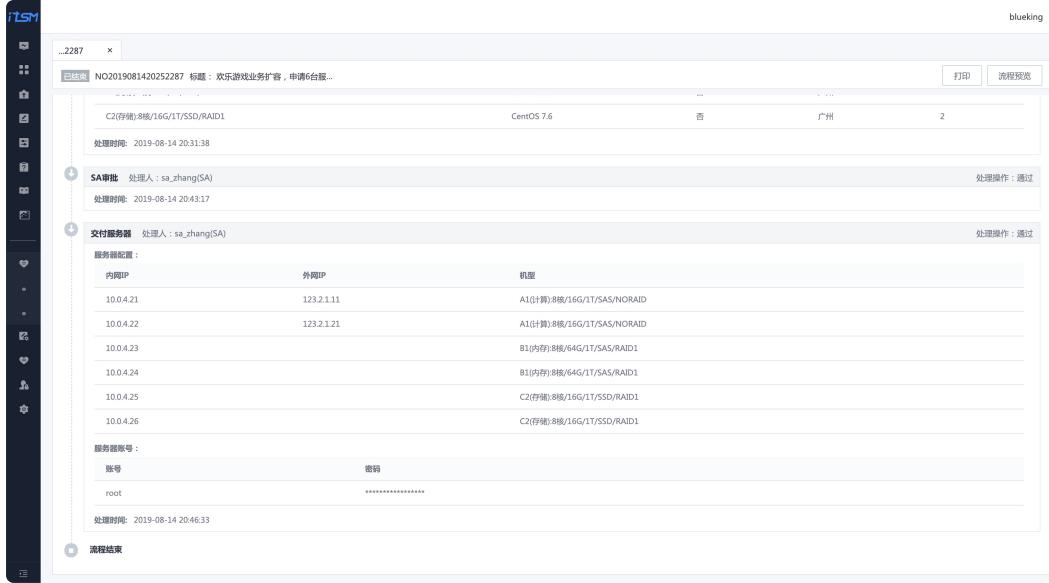
”

## 交付服务器

SA 创建资源，并录入到【交付服务器】环节的服务器配置中。

至此，服务器申请流程结束。

运维登录流程服务，即可查看申请的服务器资源。



## 流程服务：发布流程线上化

### 情景

应用发布是运维的职能之一，传统的应用发布通过邮件交付，沟通成本太高，而且不符合合规性检查。

接下来，以一个业务的应用发布为例，看蓝鲸流程服务是如何解决这个痛点的。

### 前提条件

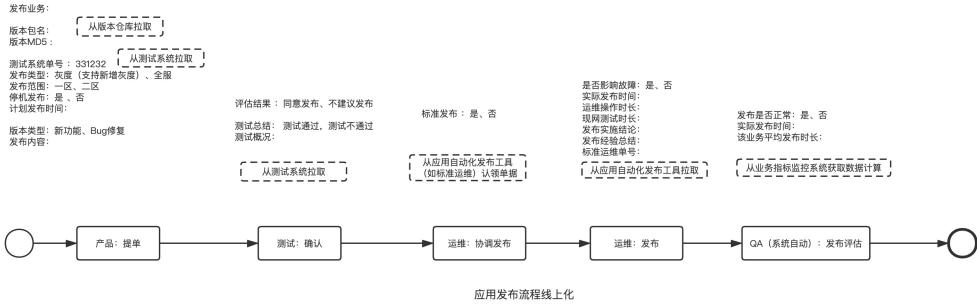
- 准备发布流程中的 **各个角色** 的账号，包含流程设计的 **管理员**，以及应用发布流程中的 **产品**、**运维**、**测试**、**QC**。

### 操作步骤

- 梳理应用发布流程
- 创建应用发布服务目录及流程
- 一次应用发布示例

#### 梳理应用发布流程

从流程服务理论出发，梳理一个应用发布流程图，将测试环境验收通过的版本，部署到生产环境。包含研发运营环节的测试、产品提交发布需求、运维协调发布资源、发布，以及最后质量保证(QA)对应用发布质量的管理。



“

流程图中是一个实践案例，部分数据需要从周边系统获取，此处功能需要做二次开发，本教程专注流程本身。

”

## 创建“应用发布”流程及服务目录

先设计应用发布的流程，流程依附在服务目录上对外提供服务。

### 角色设置

使用蓝鲸管理员账号给该业务分配对应角色的权限。

ID	业务名	运维人员
2	蓝鲸	blueking(blueking)
4	欢乐游戏(demo)	blueking(bluekin...)
5	For_haha	admin(管理员)
6	IT(TEST)	admin(管理员)
7	跨业务	blueking(blueking)

使用管理员账号新增通用角色 QC，并添加用户 qc\_c（该账户是前提条件中准备的）。

No.	角色名	人员	操作
1	QC	qc_c(QC)\C	<a href="#">编辑</a> <a href="#">删除</a>
2	SA	sa_zhang(SA)	<a href="#">编辑</a> <a href="#">删除</a>
3	服务台小组	admin(管理员)	<a href="#">编辑</a> <a href="#">删除</a>
4	故障派单员	admin(管理员)	<a href="#">编辑</a> <a href="#">删除</a>
5	变更经理	admin(管理员)	<a href="#">编辑</a> <a href="#">删除</a>
6	运营	admin(管理员),sa_zhang(SA)	<a href="#">编辑</a> <a href="#">删除</a>

## 设计应用发布流程

### 填写流程信息

选择菜单【流程设计】，点击【新增】按钮，按提示填写流程信息。

流程名称：应用发布

流程说明：将应用版本发布至生产环境

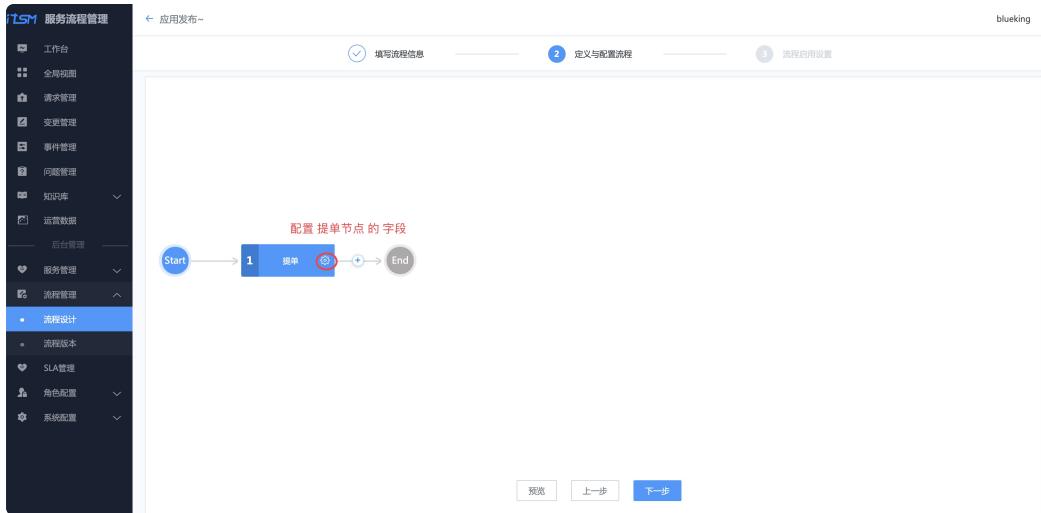
流程类型：变更

是否关联业务：是

流程类型选择【变更】，需要【关联业务】，因为应用发布和业务相关，同时关联业务对应的角色：产品、运维、测试。

点击【下一步】，进入【定义与配置流程】环节。

### 定义与配置流程



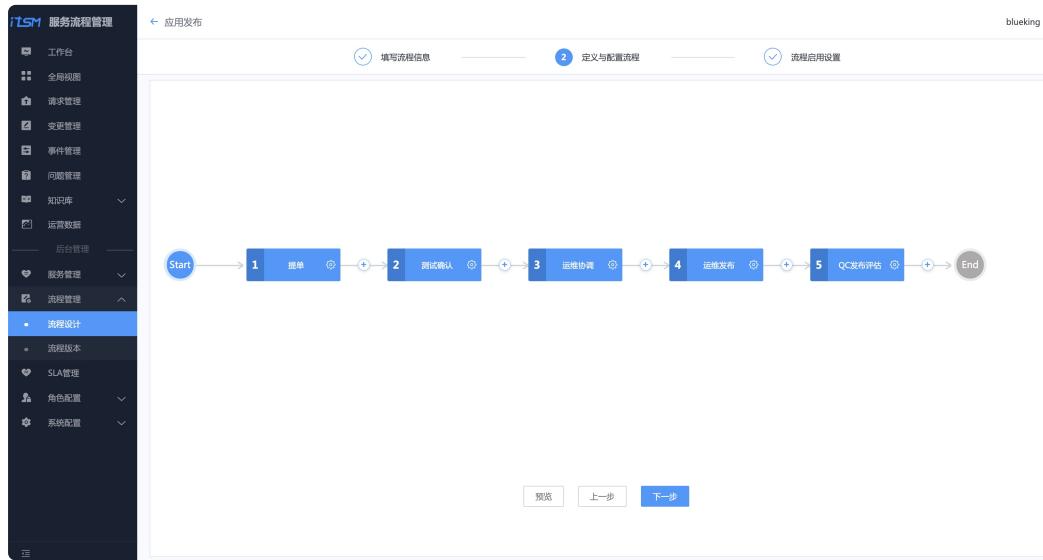
点击上图【流程画布】中的【齿轮】，配置【提单】流程节点的字段。

一般是业务的产品提应用的发布单据，所以操作角色选择【CMDB 业务公用角色】->【产品人员】。

字段名	字段类型	字段值	字段描述	字段校验方式	布局要求	字段属性	操作
标题	单行文本	--	请输入标题	必填	半行	自定义	<a href="#">上移</a> <a href="#">下移</a> <a href="#">编辑</a> <a href="#">删除</a>
版本包路径	单行文本	--	在版本仓库中，版本包的相对路径	必填	半行	自定义	<a href="#">上移</a> <a href="#">下移</a> <a href="#">编辑</a> <a href="#">删除</a>
版本MD5	单行文本	--	本次发布的版本md5值，用于版本对比	必填	半行	自定义	<a href="#">上移</a> <a href="#">下移</a> <a href="#">编辑</a> <a href="#">删除</a>
发布类型	单选框	灰度（先发布一部分节点，用...）是否先灰度		必填	半行	自定义	<a href="#">上移</a> <a href="#">下移</a> <a href="#">编辑</a> <a href="#">删除</a>
测试系统单号	单行文本	--	本次发布在测试系统报测的单号	必填	半行	自定义	<a href="#">上移</a> <a href="#">下移</a> <a href="#">编辑</a> <a href="#">删除</a>

点击【新增字段】，参照梳理应用发布流程，新增每个环节中需要的字段。

参照梳理应用发布流程，完成整个应用发布流程的配置。



### 启用流程

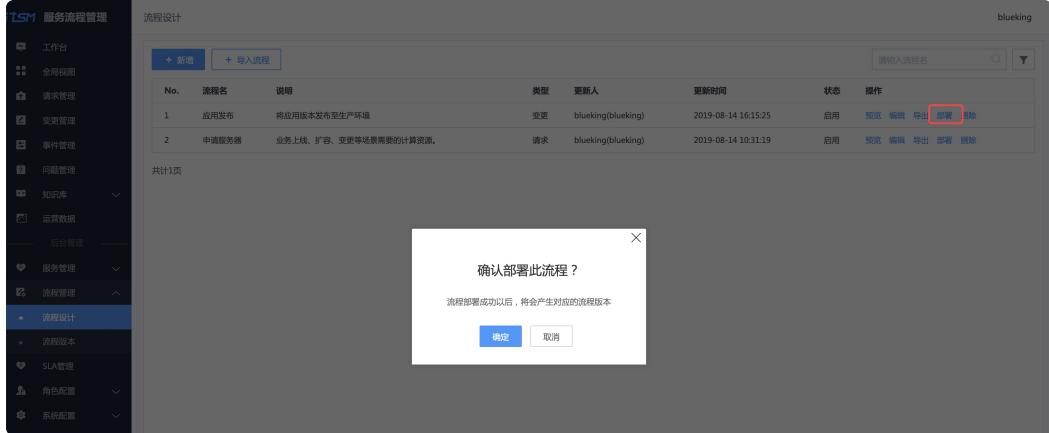
【启用流程】，选择适合的通知策略，点击【提交】完成流程设计。

设置流程启用管理  
是否启用该流程：  ON

设置任务通知策略  
是否通知： 是  否  
是否督办： 是  否  
通知频率： 首次通知，以后不再通知  
 首次通知后，次日起每天定时通知  
通知类型： SMS短信  邮件  微信

### 流程模板实例化

选择菜单【流程设计】，找到刚编辑的应用发布流程，点击【部署】，生成流程实例。

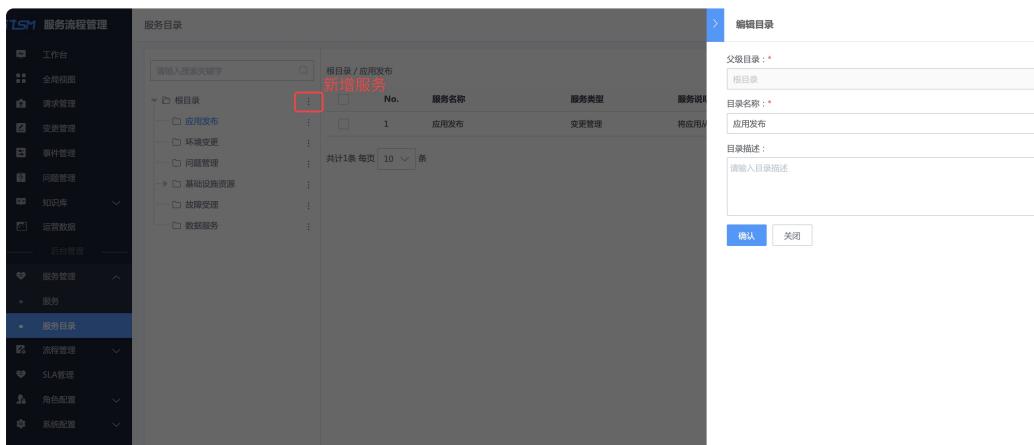


## 在服务目录中新增"应用发布"服务，并绑定流程

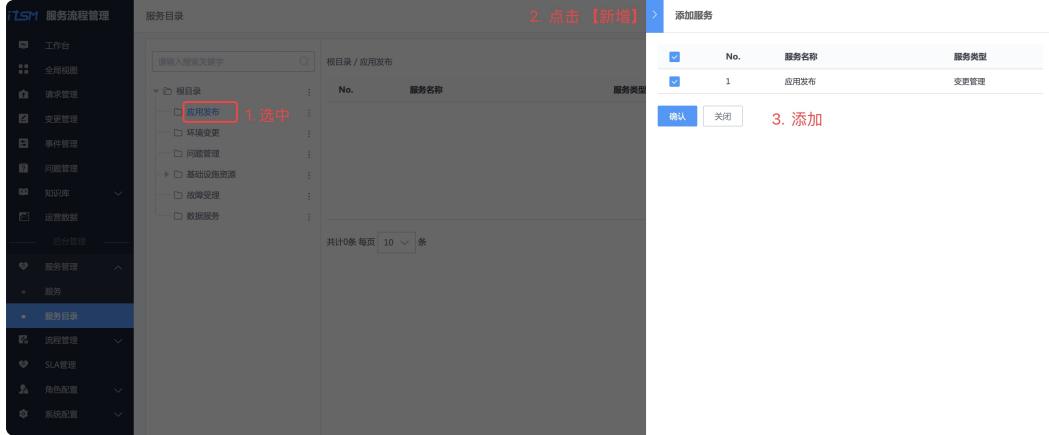
选择菜单【服务】，点击【新增】按钮，新增“应用发布”服务，并关联刚生成的流程实例。



选择菜单【服务目录】，选中【根目录】，点击右侧【：】，点击【新增】，按提示新增一个名为应用发布的服务目录。

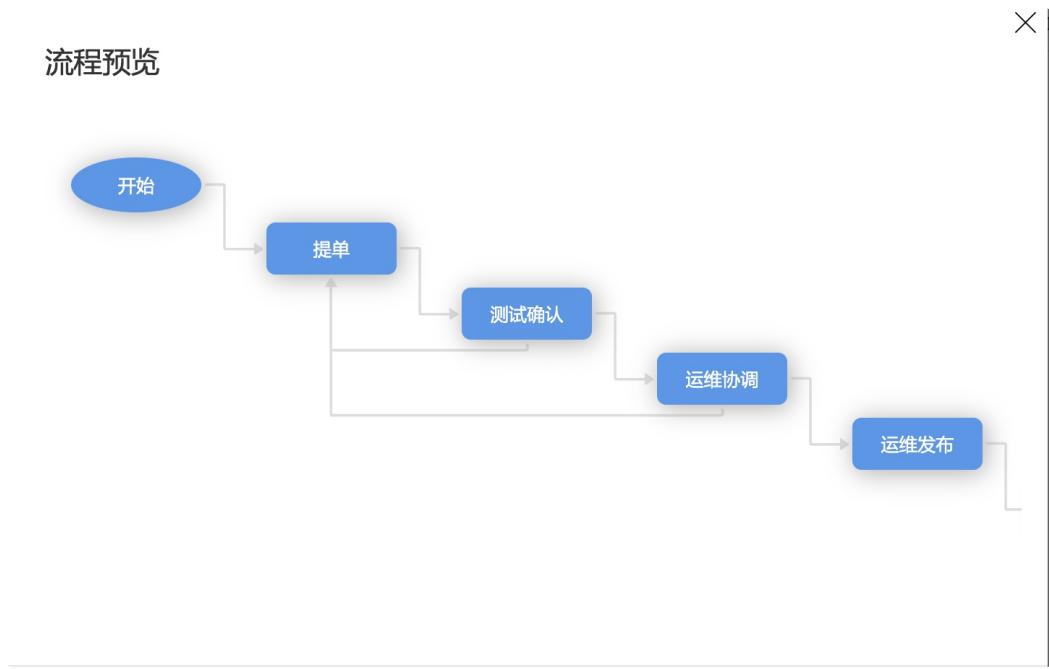


选中刚刚创建的服务目录【应用发布】，右侧会显示【添加】按钮，点击该按钮添加应用发布服务。



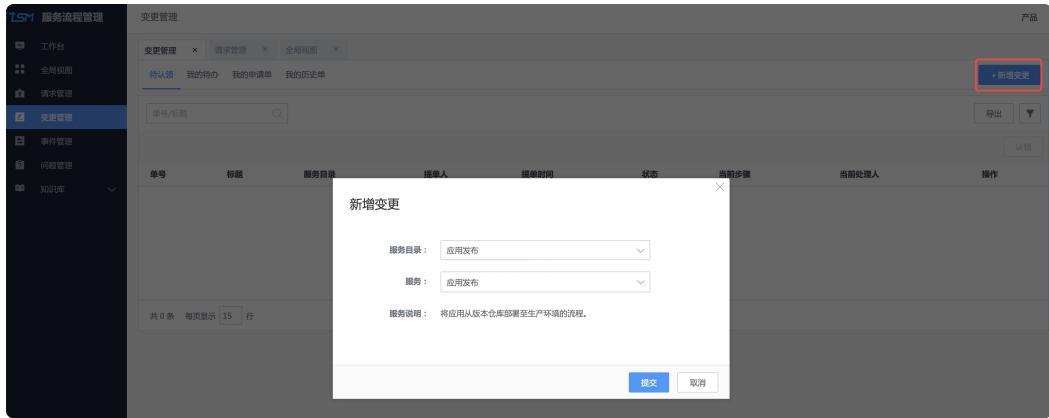
## 一次应用发布示例

创建完“应用发布”流程及服务目录后，接下来做一次应用发布演示。



## 产品提交发布需求

用 **产品** 账号登录流程服务，选择【变更管理】菜单，点击【新增变更】，选择【应用发布】服务，点击【提交】。



在变更申请界面，填写本次版本发布的关键信息，包括版本包路径、MD5，用于自动化应用发布系统直接从仓库中获取版本，以及版本发布范围和时间。

点击【提交】，创建发布需求。



## 测试确认版本质量

测试人员收到一封待处理的测试确认邮件。



**【ITSM通知】变更管理**

Hi,您好!

您有1条变更管理单【待处理】

标题: V5.1

单号: NO2019081413015540

服务目录: 应用发布

当前环节: 测试确认

当前处理人: ceshi(测试)

[点击查看详情](#)

如有任何问题, 可随时联系ITSM助手。

ITSM流程管理服务

2019年8月14日

使用测试账号登录流程服务, 在待办列表中, 找到刚刚产品提的发布单。

点击链接, 选择评估结果和测试结果, 并填写测试概况, 点击【通过】, 完成本环节流程。

## 运维协调资源

使用运维的账号登录流程服务，在待办列表找到单据。

点击【通过】，开始协调发布资源，做好发布准备。

ITSM 服务流程管理

NO2019081413015540 标题：V5.1

处理中

标题：v5.1  
发布类型：全部（发布到所有节点）  
版本包路径：/git/tms/ok\_itcm\_2.1.14.tgz  
发布范围：一区.二区.三区.四区  
停用状态：是  
计划发布时间：2019-08-15 08:00:00  
结束时间：2019-08-15 10:00:00  
版本日志：新特性  
新增欢乐模式  
礼物可以赠送给朋友  
**BUG修复**  
玩家阅读不对应先FPS帧数下降  
处理时间：2019-08-14 13:06:11

**测试确认** 处理人：ceshi(测试) 处理操作：通过  
评估结果：同意发布  
测试结果：测试通过  
测试概况：严重以上缺陷数: 0个  
测试用例: 8 小时  
测试人员: ceshi(测试)  
处理时间：2019-08-14 13:12:10

**运维协调**  
标准发布：•  是  否  
通过 通过 打回 保存字段

流转日志 通知关注人  
2019-08-14 13:06:11 blueking(blueking) 处理节点【提单】(提交)  
2019-08-14 13:12:10 ceshi(测试) 处理节点【测试确认】(通过)  
2019-08-14 13:14:34 blueking(blueking) 处理节点【运维协调】(通过)

## 运维实施发布

完成了发布准备，到了计划发布时间，在自动化应用发布系统（如标准运维）上完成应用发布后，在该发布单的运维发布环节填写本次运维发布的概况。

ITSM 服务流程管理

NO2019081413015540 标题：V5.1

处理中

**测试确认** 处理人：ceshi(测试) 处理操作：通过  
评估结果：同意发布  
测试结果：测试通过  
测试概况：严重以上缺陷数: 0个  
测试用例: 8 小时  
测试人员: ceshi(测试)  
处理时间：2019-08-14 13:12:10

**运维协调** 处理人：blueking(blueking) 处理操作：通过  
标准发布：是  
处理时间：2019-08-14 13:14:34

**运维发布**  
引发故障：•  是  否  
实际发布时间：•  2019-08-15 08:00:00 结束时间：•  2019-08-15 09:10:00  
运维操作时长：• 20分钟  
回滚操作时长：• 50分钟  
发布结果：• 完全成功  
经验总结：• 传统的应用发布模式仍然较慢，如果架构改造为微服务，使用K8S部署，可以更快，且实现不停机更新。  
发布单据：• 82837  
通过 保存字段

流转日志 通知关注人  
2019-08-14 13:06:11 blueking(blueking) 处理节点【提单】(提交)  
2019-08-14 13:12:10 ceshi(测试) 处理节点【测试确认】(通过)  
2019-08-14 13:14:34 blueking(blueking) 处理节点【运维协调】(通过)

## QC 发布评估

QC（质量保障）人员在工作台的待办列表中，找到本次发布的单据。

The screenshot shows the ITSM Service Management Platform's workstation interface. At the top, there's a header with the platform logo and a search bar. Below the header is a sidebar with navigation links: 全局视图, 请求数理, 变更管理, 事件管理, 问题管理, and 知识库. The main content area is divided into several sections:

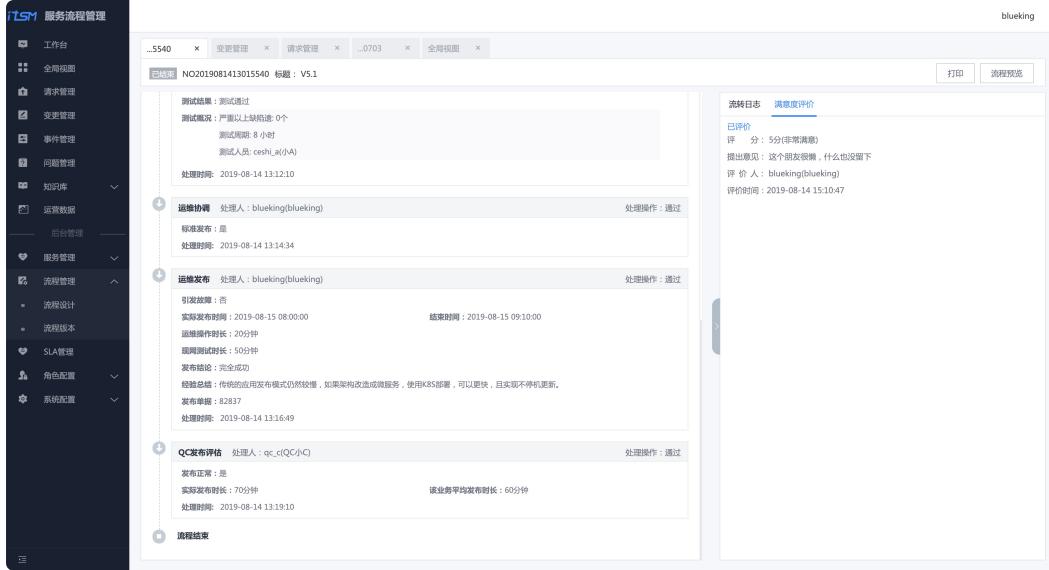
- 待办事项**: A table showing a single ticket: NO2019081413015540, Title: V5.1, 工单类型: 变更管理, 提单人: blueking(blueking), 提单时间: 2019-08-14 13:01:55, 当前步骤: QC发布评估.
- 最新动态**: A section stating "暂无动态" and "您目前还没有新动态更新！".
- 处理过的单据**: A chart showing 1 handled tickets over the last 7 days.
- 单据处理总耗时(h)**: A donut chart showing the total processing time for tickets.
- 当前单据状态分布占比**: A donut chart showing the distribution of ticket states: 待认领 (0%), 处理中 (100%), 挂起 (0%), and 待评价 (0%).

QC 根据业务指标监控系统的数据以及事件管理中是否存在关联的事件，对此次发布做出发布评估。

This screenshot shows a detailed view of a specific ticket (NO2019081413015540) in the ITSM platform. The ticket details include:

- 处理单**: NO2019081413015540 标题: V5.1
- 测试概况**: 里面有以上缺陷数: 0个, 测试用时: 8 小时, 测试人员: ceshi\_h(jyA).
- 处理时间**: 2019-08-14 13:12:10
- 运维协调**: 处理人: blueking(blueking), 处理操作: 通过, 标准发布: 是, 处理时间: 2019-08-14 13:14:34
- 运维发布**: 处理人: blueking(blueking), 处理操作: 通过, 引擎故障: 否, 实际发布开始时间: 2019-08-15 08:00:00, 结束时间: 2019-08-15 09:10:00, 详细操作时长: 209分钟, 观察测试时长: 505分钟, 发布结论: 完全成功, 经验总结: 传统的应用发布模式仍然较慢, 如果架构改造或微服务, 使用K8S部署, 可以更快, 目前实不停机更新, 发布单量: 82837, 处理时间: 2019-08-14 13:16:49
- QC发布评估**: A form where the user has selected "是" for "发布正常" and entered "70分钟" for "实际发布时长" and "60分钟" for "该业务平均发布时长". Buttons for "通过" (Pass) and "保存字段" (Save fields) are at the bottom.
- 流转日志**: A log of recent activity steps and their times.
- 通知关注人**: A list of users who have been notified or are following the ticket.

至此，一次应用发布的流程结束。



## 流程服务：环境变更流程线上化

### 情景

变更是运维的职能之一，包含应用的配置、基础设施的环境等变更。

应用所依赖的服务器已经服役 5 年，不在维保期限，故障率抖升，为了保障应用的稳定性，运维团队计划在业务低高峰期，完成服务器的以旧换新。

流程服务中的变更管理（Change Management）规范了变更环节的操作，确保高效有序的完成环境变更，对应用的可用性负责，并对生产环境的变更满足合规性检查。

变更管理是蓝鲸流程服务的一个模块，接下来介绍在蓝鲸流程服务如何完成这次服务器的替换升级。

### 前提条件

- 准备环境变更流程中 **多个角色** 的账号，包含 **运维**、**QC**、**产品**，以及流程设计的 **管理员**。

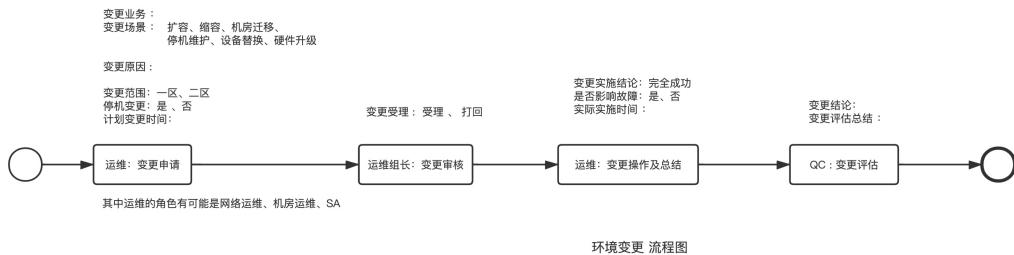
### 操作步骤

- 梳理环境变更流程
- 创建环境变更服务目录及流程
- 一次环境变更示例

### 梳理环境变更流程

从流程服务理论出发，梳理环境变更流程图，包含运维变更申请、受理变更、变更操作和总结，以及

最后质量保证 (QC) 对变更的评估管理。



“

流程图中是一个实践案例，部分数据需要从周边系统获取，此处功能需要做二次开发，本教程专注流程本身。

”

## 创建环境变更服务目录及流程

先设计环境变更的流程，流程依附在服务目录上对外提供服务。

### 角色设置

参照 [角色设置](#) 完成对 [运维](#)、[产品](#) 和 [QC](#) 的授权。

### 设计环境变更流程

#### 填写流程信息

选择菜单【流程设计】，点击【新增】按钮，按提示填写流程信息。



The screenshot shows the ITSM Service Flow Management interface. On the left, there's a sidebar with various management modules like Workbench, Global View, Request Management, Change Management, Event Management, Problem Management, Knowledge Base, Operation Data, and more. The 'Process Design' module is currently selected. In the main area, a wizard titled 'Add New Process' is displayed with three steps: 1. Fill in Process Information (active), 2. Define and Configure Process, 3. Set Process Activation Configuration. Step 1 contains fields for 'Process Name' (set to '环境变更'), 'Process Description' (set to '应用的配置、基础设施的环境等变更'), 'Process Type' (set to '变更'), and 'Is Associated Business' (set to 'Yes'). At the bottom are 'Back' and 'Next' buttons.

流程类型选择【变更】，需要【关联业务】，因为环境变更和业务相关，同时关联业务对应的角色：产品、运维。

点击【下一步】，进入【定义与配置流程】环节。

## 定义与配置流程



点击上图【流程画布】中的【齿轮】，配置【提单】流程节点的字段。

一般是运维提环境变更单据，所以操作角色选择【CMDB 业务公用角色】->【运维人员】(角色授权详见给角色分配权限)。

The screenshot shows the 'Configure Node' step (Step 3) of the process. On the left, the sidebar shows the 'Flow Design' section is selected. The main area displays the 'Basic Information' tab for a node named '提单'. It shows the 'Operation Role' dropdown set to 'CMDB Business Public Role' and the 'Environment Change, generally maintenance' role set to 'Maintenance Personnel'. A red box highlights the 'Maintenance Personnel' selection. Below this, a table titled '+ Add Field' lists a single field: 'Title' (Single-line Text) with value '''. The table includes columns for 'Field Name', 'Field Type', 'Field Value', 'Field Description', 'Field Validation Method', 'Layout Requirements', 'Field Properties', and 'Operations'. At the bottom, there are 'Save' and 'Field Preview' buttons.

点击【新增字段】，参照梳理环境变更流程，新增每个环节中需要的字段。

**基本信息**

节点名称 :	提单		
操作角色 :	CMDB业务公用角色		
<b>+ 新增字段</b>			
字段名	字段类型	字段值	字段描述
标题	单行文本	--	一句话
变更场景	单选下拉框	扩容,缩容,机房迁移,停机维护,设...	这是哪一类变更

**字段类型 :** 单选下拉框

**数据源 :** 自定义数据 (radio selected), 接口数据 (radio unselected), 数据字典 (radio unselected)

**自定义数据 :** 扩容, 缩容

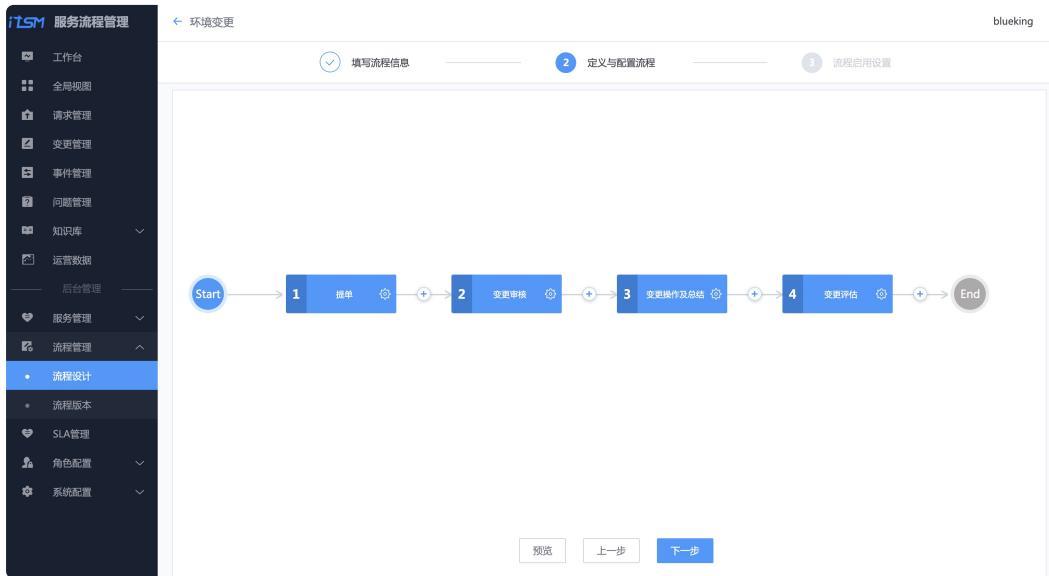
**布局要求 :** 半行

**字段校验方式 :** 必填

**正则规则表达式 :** 无

**按钮:** 保存, 字段预览, 提交, 取消

参照 [梳理环境变更流程](#)，完成整个环境变更流程的配置。



### 启用流程

【启用流程】，选择适合的通知策略，点击【提交】完成流程设计。

### 流程模板实例化

选择菜单【流程设计】，找到刚编辑的环境变更流程，点击【部署】，生成流程实例。

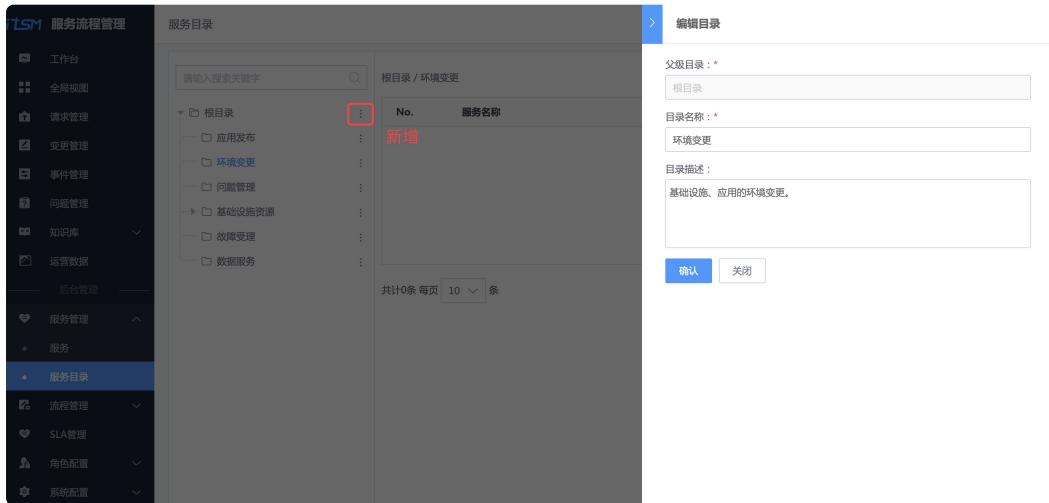
No.	流程名	说明	类型	更新人	更新时间	状态	操作
1	环境变更	应用的配置、基础设置的环境等变更	变更	blueking(blueking)	2019-08-15 19:27:44	启用	<a href="#">预览</a> <a href="#">编辑</a> <a href="#">导出</a> <a href="#">部署</a> <a href="#">删除</a>
2	故障受理	应对生产环境故障的流程。	事件	blueking(blueking)	2019-08-15 16:59:06	启用	<a href="#">预览</a> <a href="#">编辑</a> <a href="#">导出</a> <a href="#">部署</a> <a href="#">删除</a>
3	应用发布	将应用版本发布至生产环境	变更	blueking(blueking)	2019-08-14 16:15:25	启用	<a href="#">预览</a> <a href="#">编辑</a> <a href="#">导出</a> <a href="#">部署</a> <a href="#">删除</a>
4	申请服务器	业务上线、扩容、变更等场景需要的计算资源。	请求	blueking(blueking)	2019-08-14 20:24:43	启用	<a href="#">预览</a> <a href="#">编辑</a> <a href="#">导出</a> <a href="#">部署</a> <a href="#">删除</a>

### 在服务目录中新增"环境变更"服务，并绑定流程

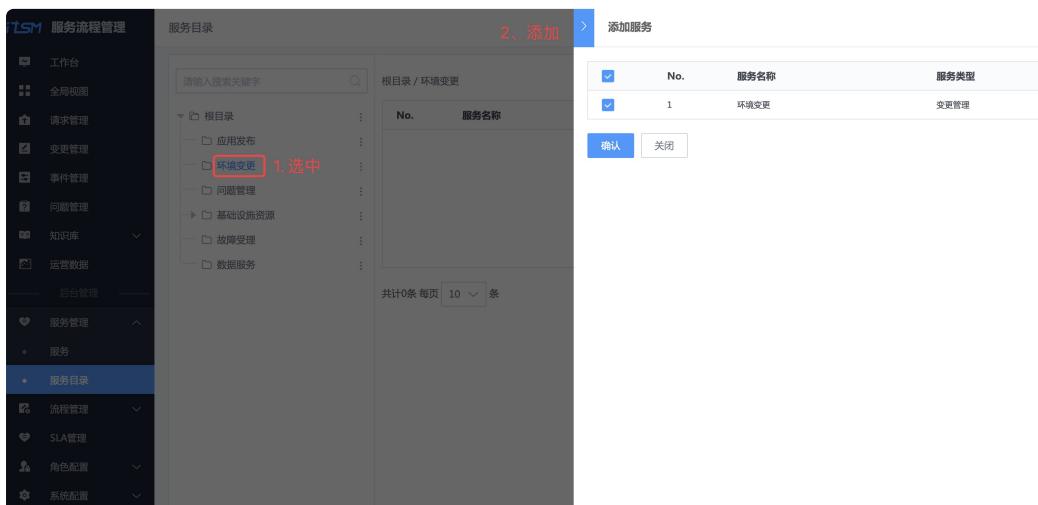
选择菜单【服务】，点击【新增】按钮，新增"环境变更"服务，并关联刚生成的流程实例。



选择菜单【服务目录】，选中【根目录】，点击右侧【：】，点击【新增】，按提示新增一个名为环境变更的服务目录。



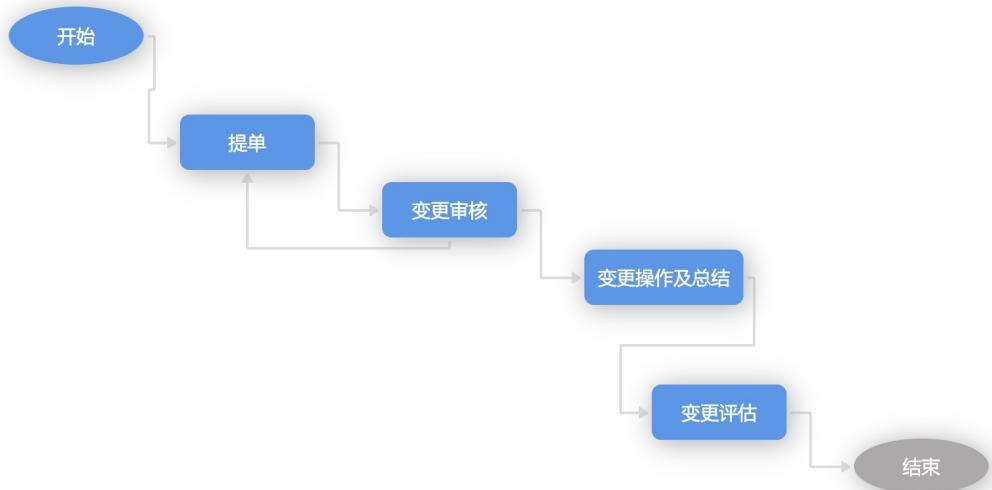
选中刚刚创建的服务目录【环境变更】，右侧会显示【添加】按钮，点击该按钮添加环境变更服务。



至此，流程设计和服务目录已新建好，接下来做一次环境变更演示。

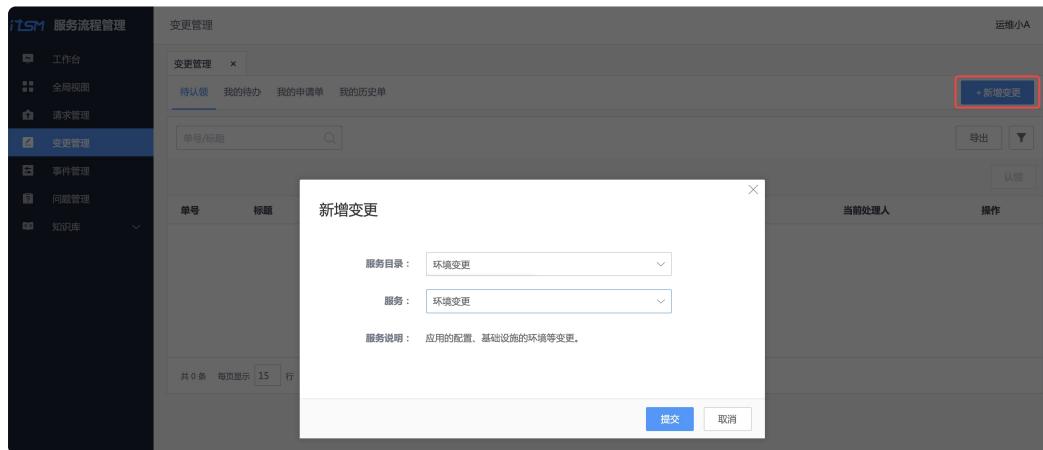
## 一次环境变更示例

### 流程预览



### 运维：变更申请

用[运维](#)账号登录流程服务，选择【变更管理】菜单，点击【新增变更】，选择【环境变更】服务，点击【提交】。



在【新增变更】界面，填写本次环境变更的关键信息，包括变更原因、范围以及时间等。

点击【提交】，完成变更申请。

ITSM 服务流程管理

运维小A

变更申请

是否使用模板：  使用模板可以快速填写字段信息

标题：**一区、二区服务器以旧换新**

关联业务：**欢乐游戏(demo)**

变更场景：**硬件升级**

变更原因：**应用所依赖的服务器已经服役5年，不在维保期限，故障率飙升，为了保障应用的稳定性，运维团队计划在业务低峰期，完成服务器的以旧换新。**

变更范围：**一区,二区**

停机变更：**是**

计划变更时间：**2019-08-15 20:00:00**

结束时间：**2019-08-15 21:00:00**

**提交** **保存模板** **取消**

## 运维组长：变更审核

运维组长收到一封待处理的变更审核邮件。



### 【ITSM通知】变更管理

Hi,您好!

您有1条变更管理单【待处理】

标题: **一区、二区服务器以旧换新**

单号: NO2019081519494023

服务目录: 环境变更

当前环节: 变更审核

当前处理人: blueking(blueking),ops\_a(运维小A)

[点击查看详情](#)

如有任何问题, 可随时联系ITSM助手。

ITSM流程管理服务

2019年8月15日

使用运维账号登录流程服务, 在待办列表中, 找到待处理的环境变更单。

LSM 服务流程管理

工作台

待办列表 1

单号	标题	工单类型	提交人	提交时间	当前步骤
NO2019081519494023	一区、二区服务器以旧换新	变更管理	ops_a(运维小A)	2019-08-15 19:49:40	变更审核

最新动态

- 14秒前 NO2019081519494023 ops\_a(运维小A) 处理节点【提单】(提交)
- 5分钟前 NO2019081519403742 ops\_a(运维小A) 处理节点【变更操作及总结】(通过)
- 6分钟前 NO2019081519403742 ops\_a(运维小A) 处理节点【变更审核】(通过)
- 7分钟前 NO2019081519403742 ops\_a(运维小A) 处理节点【提单】(提交)
- 2小时前 NO2019081517062115 ops\_a(运维小A) 处理节点【故障总结】(通过)
- 2小时前 NO2019081517062115 ops\_a(运维小A) 处理节点【故障跟进】(通过)

处理过的单据 最近7天

单据处理总耗时(h) 最近7天

当前单据状态分布占比

点击链接，审核本次变更，点击【通过】，完成本环节流程。

LSM 服务流程管理

工作台

全局视图 变更管理 求索赔 环境变更 全局视图 ...3742

处理中 NO2019081519494023 标题：一区、二区服务器以旧换新 挂起 撤单 打印 流程预览

**基本信息**

工单类型： 变更管理 服务目录： 环境变更  
提单人： ops\_a(运维小A) / 18111112222 提单部门： --  
当前处理人： blueking(blueking),ops\_a(运维小A)  
提单时间： 2019-08-15 19:49:40

**工单处理**

**提单** 处理人：ops\_a(运维小A) 处理操作：提单  
标题：一区、二区服务器以旧换新 关联业务：欢乐游戏(demo)  
变更场景：硬件升级  
变更原因：应用所依赖的服务器已经服役5年，不在维保期限，故障率上升，为了保障应用的稳定性，运维团队计划在业务低高峰期，完成服务器的以旧换新。  
变更范围：一区、二区 停机变更：是  
计划变更时间：2019-08-15 20:00:00 结束时间：2019-08-15 21:00:00  
处理时间：2019-08-15 19:51:41

**变更审核**

变更受理： 受理 通过 打回 保存字段

## 运维：变更操作及总结

环境变更完毕后，填写变更总结。

ITSM 服务流程管理

..4023

处理中 NO2019081519494023 标题：一区、二区服务器以旧换新

当前处理人：blueking(jueying).ops\_a(运维小A)

提单时间：2019-08-15 19:49:40

**工单处理**

**提单** 处理人：ops\_a(运维小A) 处理操作：提单

标题：一区、二区服务器以旧换新  
关联业务：欢乐游戏(demo)  
变更场景：硬件升级  
变更原因：应用所依赖的服务器已经服役5年，不在维保期限，故障率飙升，为了保障应用的稳定性，运维团队计划在业务低高峰期，完成服务器的以旧换新。  
变更范围：一区、二区  
计划变更时间：2019-08-15 20:00:00  
停机变更：是  
结束时间：2019-08-15 21:00:00  
处理时间：2019-08-15 19:51:41

**变更审核** 处理人：ops\_a(运维小A) 处理操作：通过

变更类型：受理  
处理时间：2019-08-15 19:53:06

**变更操作及总结**

变更实施结论： 完全成功 是否引发故障： 是  否  
计划变更时间： 2019-08-15 20:00:00 结束时间： 2019-08-15 20:30:00

通过 保存字段

## QC：变更评估

QC（质量保障）人员在工作台的待办列表中，找到本次环境变更的单据。

ITSM 服务流程管理

工作台

待认领列表 待办列表 1

单号	标题	工单类型	提单人	提单时间	当前步骤
NO2019081519494023	一区、二区服务器以旧换新	变更管理	ops_a(运维小A)	2019-08-15 19:49:40	变更评估

最新动态

- 4小时前 NO2019081519403742 qc\_c(QCjC) 处理节点【变更评估】(通过)
- 2小时前 NO2019081517062115 qc\_c(QCjC) 处理节点【故障评估】(通过)
- 1天前 NO2019081413015540 qc\_c(QCjC) 处理节点【QC发布评估】(通过)

QC 依据该业务的业务关键指标数据、投诉情况，做出变更评估。

The screenshot shows a detailed view of a change request (NO2019081519494023) titled '一区、二区服务器以旧换新'. The interface includes sections for '变更审核' (Change Review), '变更操作及总结' (Change Operation and Summary), and '变更评估' (Change Evaluation). The evaluation section shows a successful outcome ('完全成功') and matches with expectations ('符合变更预期'). On the right, a '流转日志' (Flow Log) and '通知关注人' (Notify Stakeholders) are listed.

至此，一次环境变更的流程结束。

The screenshot shows the final state of the change request (NO2019081519494023) with all stages and details filled in. The '工单处理' (Work Order Processing) section shows the final status as '已结案' (Case Closed). The entire process is summarized in the '流程结束' (Process Completed) section.

## 流程服务：故障提报流程线上化

### 情景

流程服务中的故障管理(Incident Management)帮企业构建了一套应对故障的处理机制，确保故障来临之时可以高效有序的响应、处理以及回溯故障的表面和本质，并将根源问题通过转单到问题管理模块，实现故障的闭环。

故障管理是蓝鲸流程服务内置的模块之一，接下来介绍在蓝鲸流程服务是如何应对部分地区用户集中无法访问网站的投诉。

## 前提条件

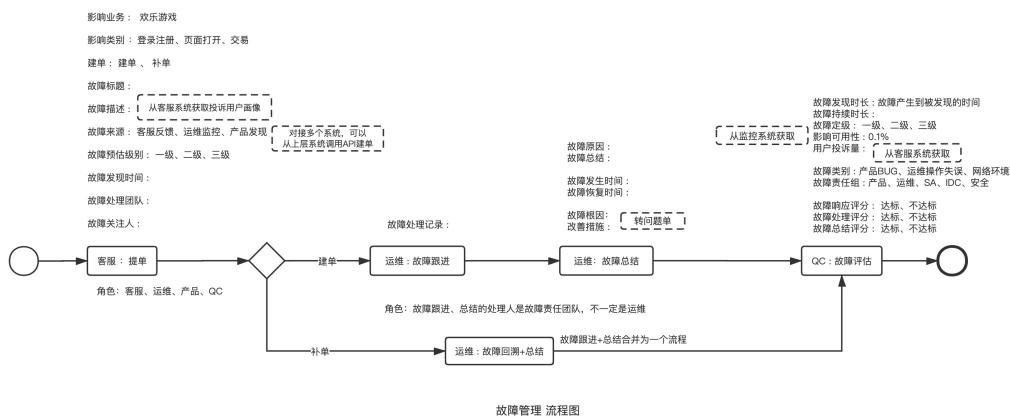
- 准备故障汇报流程中 **多个角色** 的账号，包含 **运维**、**QC**、**产品**，以及流程设计的 **管理员**。

## 操作步骤

- 梳理故障汇报流程
- 创建故障汇报服务目录及流程
- 一次故障汇报示例

### 梳理故障汇报流程

从流程服务理论出发，梳理故障汇报的流程图，包含客服 提单、运维 故障跟进、运维 回溯总结故障，以及最后质量保证(QC)对故障的 评估管理。



流程图中是一个实践案例，部分数据需要从周边系统获取，此处功能需要做二次开发，本教程专注流程本身。

## 创建故障汇报服务目录及流程

先设计故障汇报的流程，流程依附在服务目录上对外提供服务。

### 角色设置

参照 **角色设置** 完成对 **运维**、**产品** 和 **QC** 的授权。

### 设计故障受理流程

## 填写流程信息

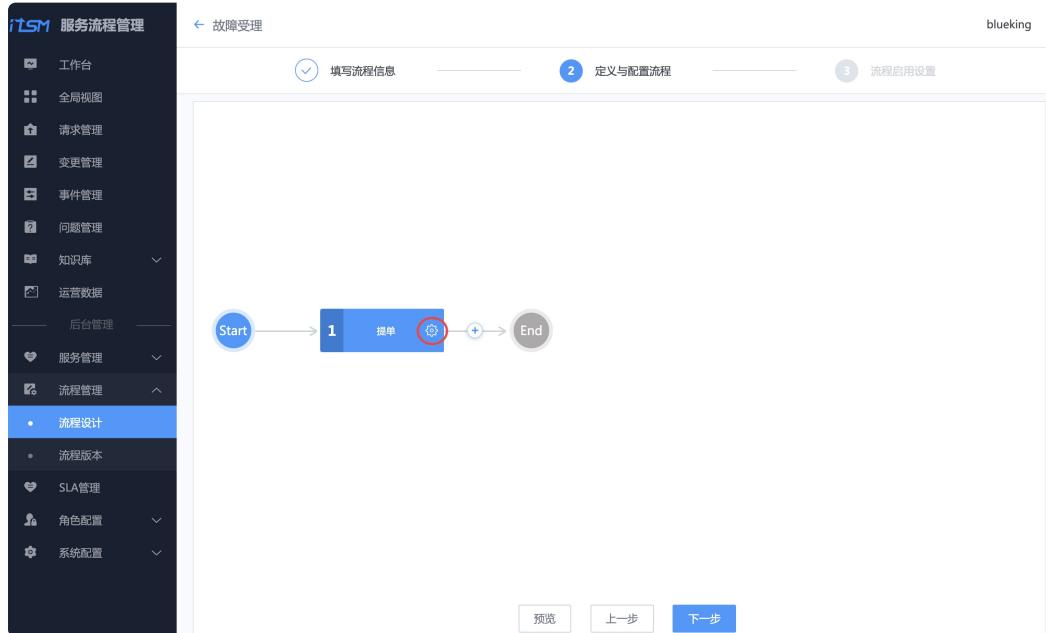
选择菜单【流程设计】，点击【新增】按钮，按提示填写流程信息。

The screenshot shows the 'Fill in Process Information' step of a three-step process. The left sidebar shows 'Workflow Management' with 'Process Design' selected. The main area has three tabs: 1. 填写流程信息 (selected), 2. 定义与配置流程, 3. 流程启用设置. The form fields are: 流程名称: 故障受理 (highlighted with a red box), 流程说明: 应对生产环境故障的流程., 至少输入5个字 12/100, 流程类型: 事件 (highlighted with a red box), 是否关联业务: 是 (radio button selected). Buttons at the bottom are 'Return' and 'Next Step'.

流程类型选择【事件】，需要【关联业务】，因为故障提报和业务相关，同时关联业务对应的角色：产品、运维。

点击【下一步】，进入【定义与配置流程】环节。

## 定义与配置流程



点击上图【流程画布】中的【齿轮】，配置【提单】流程节点的字段。

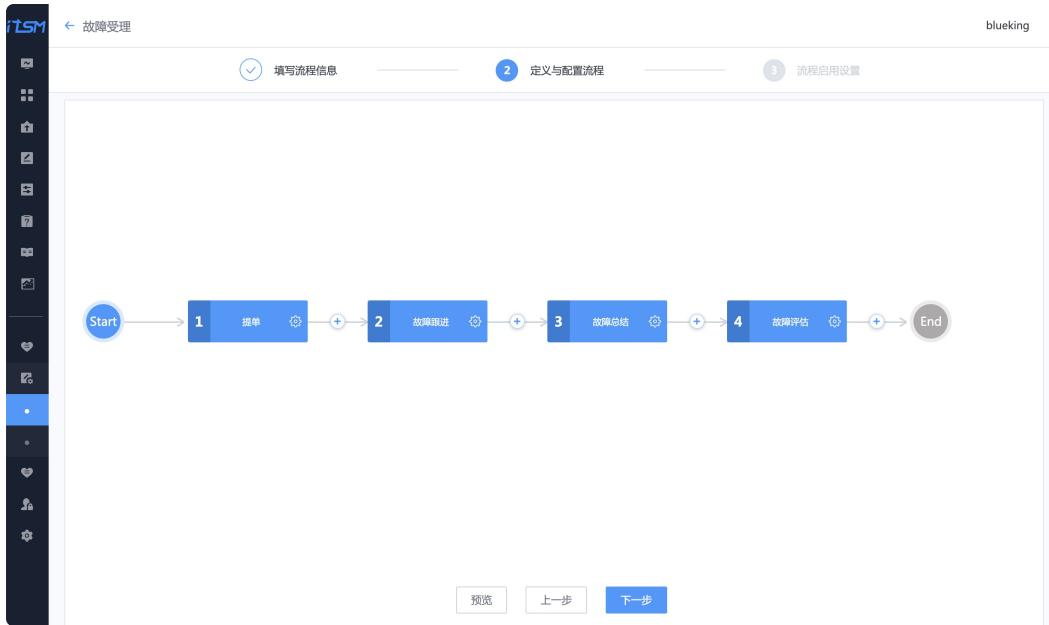
一般是客服提故障单据，所以操作角色选择【通用角色表】->【客服】(客服角色授权详见给 角色分配权限)。

The screenshot shows the 'ITSM 服务流程管理' (ITSM Service Flow Management) application. On the left, there's a sidebar with various management modules like Workstation, Global View, Request Management, Change Management, Event Management, Problem Management, Knowledge Base, and Operation Data. The '流程设计' (Process Design) module is currently selected. The main panel is titled '配置节点' (Configure Node). It has a '基本信息' (Basic Information) section where '节点名称' (Node Name) is set to '提单' and '操作角色' (Operational Role) is set to '通用角色表'. A red box highlights the '客服(1)' (Customer Service 1) option under '操作角色'. Below this is a table for '新增字段' (Add New Fields), which includes columns for '字段名' (Field Name), '字段类型' (Field Type), '字段值' (Field Value), '字段描述' (Field Description), '字段校验方式' (Field Validation Method), '布局要求' (Layout Requirements), '字段属性' (Field Properties), and '操作' (Operations). One row is shown with '标题' (Title) as the field name, '单行文本' (Single-line Text) as the type, and '必填' (Required) as the validation method. Buttons for '保存' (Save) and '字段预览' (Field Preview) are at the bottom.

点击【新增字段】，参照梳理故障提报流程，新增每个环节中需要的字段。

This screenshot shows the '编辑字段' (Edit Field) dialog box overlaid on the configuration interface. The dialog has several sections: '字段显示名' (Field Display Name), '影响类别' (Impact Category), '字段描述' (Field Description), '字段类型' (Field Type) set to '单选框' (Radio Button), '数据源' (Data Source) with '自定义数据' (Custom Data) selected, '自定义数据' (Custom Data) containing '登录注册' (Login/Registration), '页面访问' (Page Visit), and '六四' (Sixty-four), and '布局要求' (Layout Requirements) set to '半行' (Half-line). The main configuration interface on the left shows the same basic node information and a table for adding fields, with the '字段名' (Field Name) column showing '标题', '影响类别', '建单', '故障描述', and '故障来源'.

参照梳理故障提报流程，完成整个故障受理流程的配置。

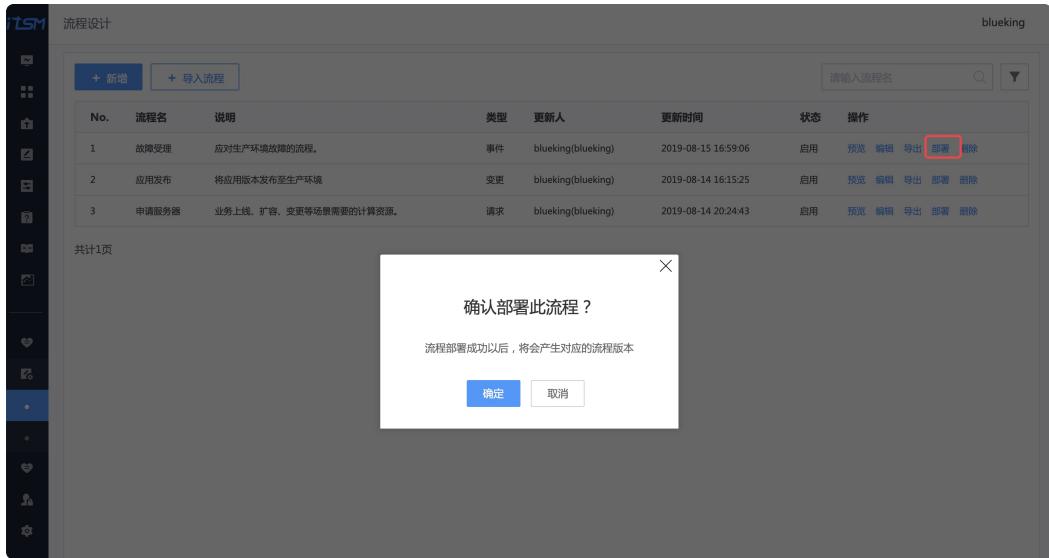


### 启用流程

【启用流程】，选择适合的通知策略，点击【提交】完成流程设计。

### 流程模板实例化

选择菜单【流程设计】，找到刚编辑的故障受理流程，点击【部署】，生成流程实例。

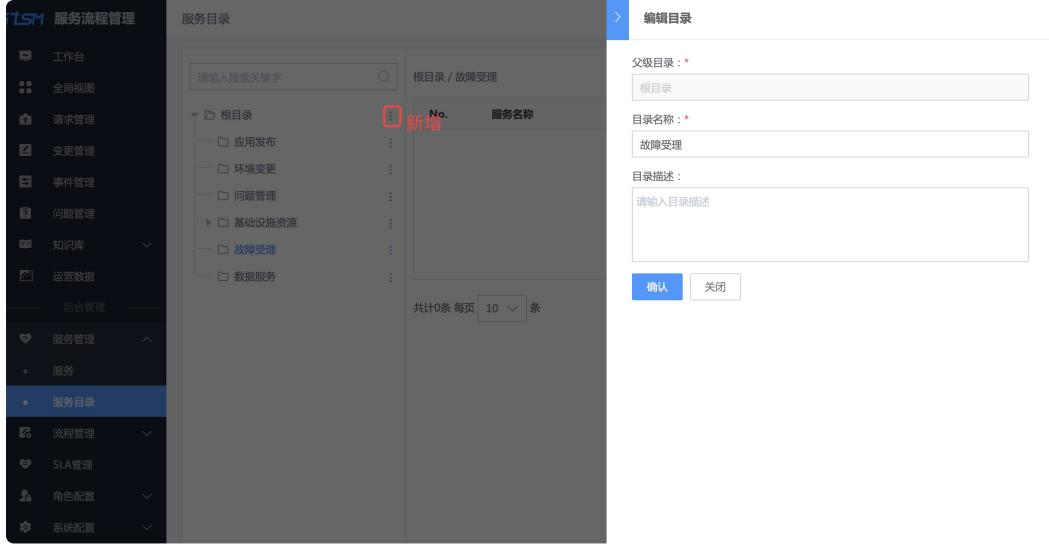


### 在服务目录中新增"故障受理"服务，并绑定流程

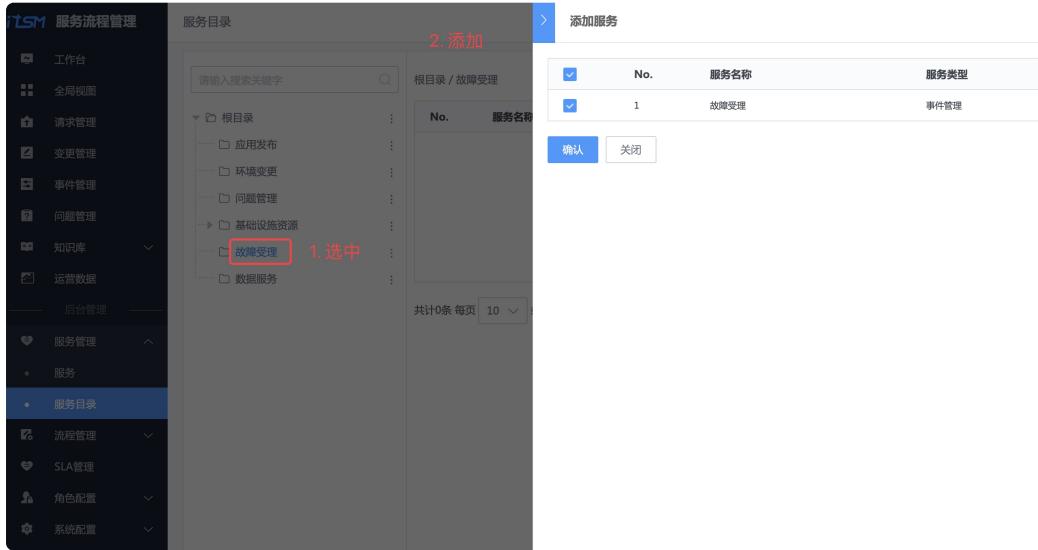
选择菜单【服务】，点击【新增】按钮，新增"故障受理"服务，并关联刚生成的流程实例。



选择菜单【服务目录】，选中【根目录】，点击右侧【：】，点击【新增】，按提示新增一个名为故障受理的服务目录。



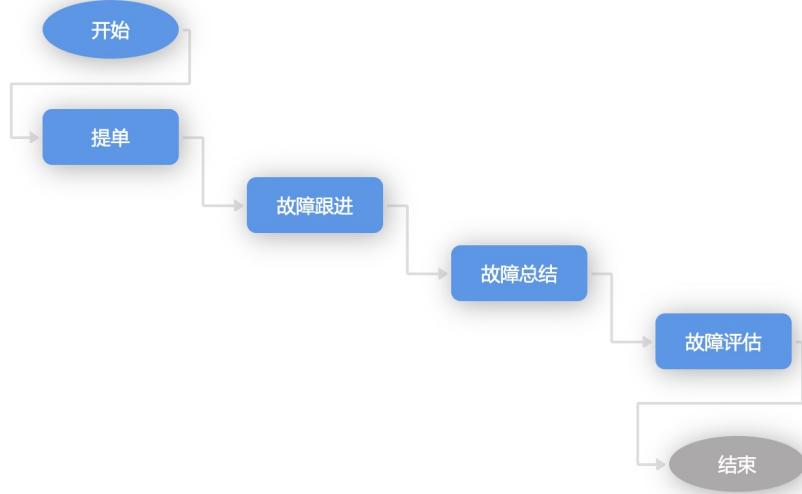
选中刚刚创建的服务目录【故障受理】，右侧会显示【添加】按钮，点击该按钮添加故障受理服务。



至此，流程设计和服务目录已新建好，接下来做一次故障提报演示。

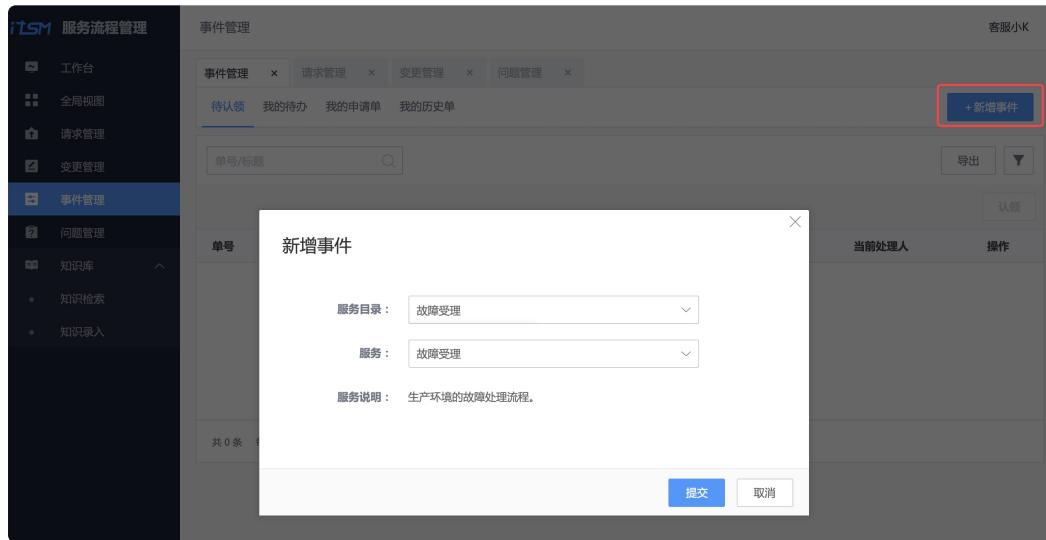
## 一次故障提报示例

## 流程预览



### 客服提报故障单

用 **客服** 账号登录流程服务，选择【事件管理】菜单，点击【新增事件】，选择【故障受理】服务，点击【提交】。



在【新增事件】界面，填写本次故障提报的关键信息，包括故障标题、截图以及故障处理团队等。

点击【提交】，提报故障。

ITSM 服务流程管理

客服小K

事件申请

是否使用模板： 使用模板可以快速填写字段信息

标题： 关联业务：

影响类别： 登录注册  页面访问  交易 建单： 建单  补单

故障描述：  
截图如下，疑似该运营商的DNS未解析A网站的域名

**无法访问此网站**

拒绝了我们的连接请求。  
请试试以下办法：  
 检查网络连接

Markdown WYSIWYG

故障来源： 客服反馈  运维监控  产品发现 故障预估级别： 一级  二级  三级

故障发现时间： 故障处理团队：

故障关注人：

**提交** **保存模板** **取消**

## 运维跟进故障处理

运维收到一封待处理的故障受理邮件。

**【ITSM】事件管理单【待处理】** ☆

发件人: **breaking** <[REDACTED]@tom>

时间: 2019年8月15日(星期四)下午5:20

收件人: [REDACTED]

### 【ITSM通知】事件管理

Hi,您好!

您有1条事件管理单【待处理】

标题: 北方地区的某运营商用户, 集中无法访问A网站

单号: NO2019081517062115

服务目录: 故障受理

发现时间: --

当前环节: 故障跟进

当前处理人: blueking(blueking),ops\_a(运维小A)

[点击查看详情](#)

如有任何问题, 可随时联系ITSM助手。

ITSM流程管理服务

2019年8月15日

使用运维账号登录流程服务, 在待办列表中, 找到待处理的故障单。

ITSM 服务流程管理

工作台

待认领列表 待办列表 1

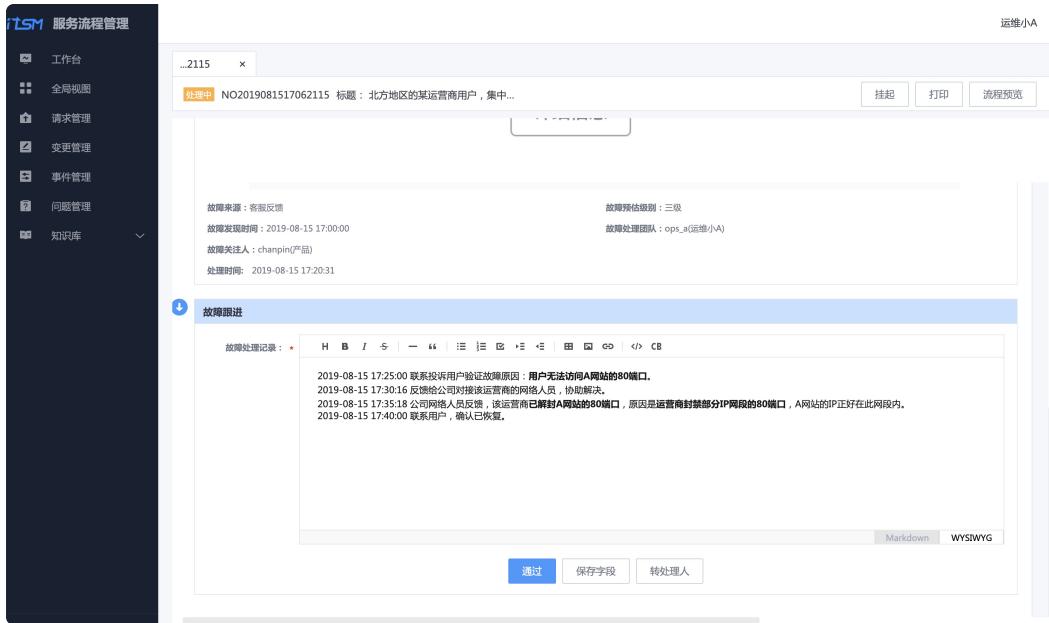
单号	标题	工单类型	提单人	提单时间	当前步骤
NO2019081517062115	北方地区的某运营商用户, 集...	事件管理	kefu(客服小K)	2019-08-15 17:06:21	故障跟进

运维小A

最新动态

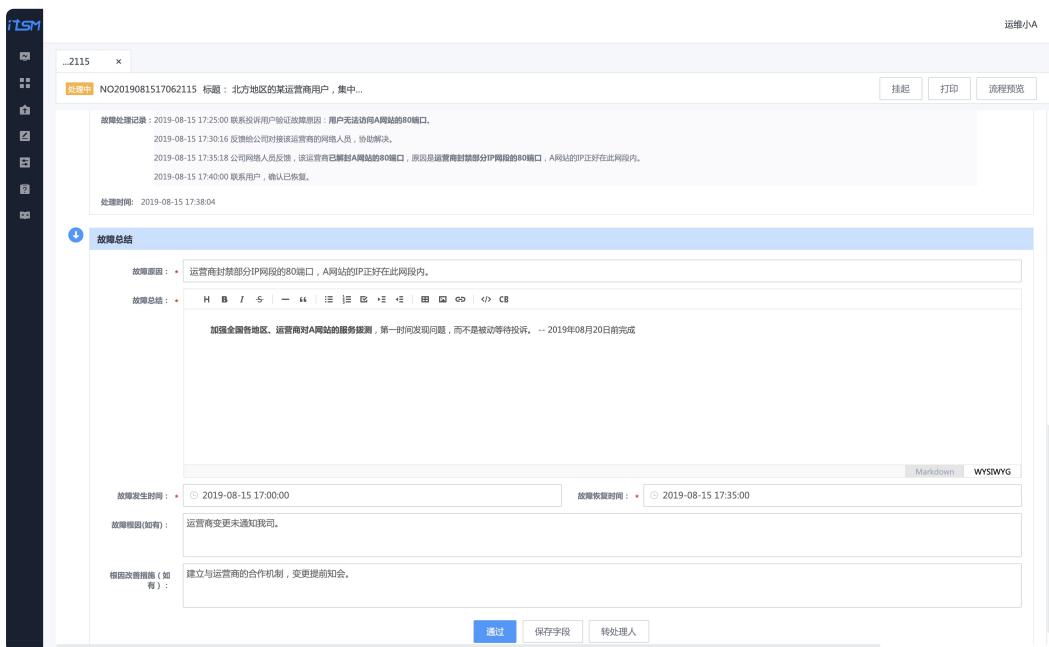
暂无动态  
您目前还没有新动态更新 !

点击链接, 填写故障处理记录, 点击【通过】, 完成本环节流程。



## 故障总结

处理完故障后，运维复盘本次故障，做故障总结。



## QC 做故障评估

QC（质量保障）人员在工作台的待办列表中，找到本次故障的单据。

工作台

QC小C

待办列表

单号 标题 工单类型 提单人 提单时间 当前步骤

NO2019081517062115 北方地区的某运营商用户，集中无法访... 事件管理 kefu(客服小K) 2019-08-15 17:06:21 故障评估

最新动态

1天前 NO2019081413015540 qc\_c(QC小C) 处理节点【QC发布评估】(通过)

QC 依据本次故障的影响范围、原因以及处理情况，做出故障评估。

...2115 x

处理中 NO2019081517062115 标题：北方地区的某运营商用户，集中...

处理时间：2019-08-15 17:38:04

故障总结 处理人：ops\_a(运维小A) 处理操作：通过

故障原因：运营商封禁部分IP网段的80端口，A网站的IP正好在此网段内。  
故障总结：加强全网各地区、运营商对A网站的服务探测，第一时间发现问题，而不是被动等待投诉。-- 2019年08月20日前完成  
故障发生时间：2019-08-15 17:00:00 故障恢复时间：2019-08-15 17:35:00  
故障根因(如有)：运营商变更未通知我司。  
根因改善措施（如有）：建立与运营商的合作机制，变更提前知会。  
处理时间：2019-08-15 17:44:56

故障评估

故障发现时长： 20分钟 故障持续时长： 35分钟  
故障定级：  一级  二级  三级 影响可用性： 0.01%  
用户投诉量： 6 故障类别： 网络环境  
故障责任组： IDC 故障响应评分：  达标  不达标  
故障处理评分：  达标  不达标 故障总结： 达标

通过 保存字段 转处理

至此，一次故障汇报和处理的流程结束。

...2115 x

已结束 NO2019081517062115 标题：北方地区的某运营商用户，集中...

2019-08-15 17:30:10 反馈给公司对谈该运营商的网络人员，协助解决。  
2019-08-15 17:35:18 公司网络人员反馈，该运营商已解封A网站的80端口，原因是运营商封禁部分IP网段的80端口，A网站的IP正好在此网段内。  
2019-08-15 17:40:00 联系用户，确认已恢复。  
处理时间：2019-08-15 17:38:04

故障总结 处理人：ops\_a(运维小A) 处理操作：通过

故障原因：运营商封禁部分IP网段的80端口，A网站的IP正好在此网段内。  
故障总结：加强全网各地区、运营商对A网站的服务探测，第一时间发现问题，而不是被动等待投诉。-- 2019年08月20日前完成  
故障发生时间：2019-08-15 17:00:00 故障恢复时间：2019-08-15 17:35:00  
故障根因(如有)：运营商变更未通知我司。  
根因改善措施（如有）：建立与运营商的合作机制，变更提前知会。  
处理时间：2019-08-15 17:44:56

故障评估 处理人：qc\_c(QC小C) 处理操作：通过

故障发现时长：20分钟 故障持续时长：35分钟  
故障定级：三级 影响可用性：0.01%  
用户投诉量：6 故障类别：网络环境  
故障责任组：IDC 故障响应评分：达标  
故障处理评分：达标 故障总结：达标

流程结束

# 进程与端口监控

当前进程和端口的监控依赖 CMDB 的配置，CMDB 配置好后就可以自动采集进程相关的数据和事件。

## 第一步：配置平台注册进程

配置路径：导航 → 业务 → 1)业务拓扑 → 2)模块设置 → 4)服务实例设置 → 5)进程设置 → 6)标签设置(可选)



进程名称	进程别名	进程启动参数	绑定IP	端口	工作路径	操作
java	java-broker	/data/mapleleaf/druid/con...	0.0.0.0	8082	--	编辑
java	java-broker	/data/mapleleaf/druid/con...	0.0.0.0	8082	--	编辑

- 进程信息配置
- 进程别名(必填): 对外显示的服务名，用户自定义，一般与进程实际服务名称对应，会在监控平台-主机监控-主机详情-进程服务中用于区分不同进程
- 进程名(必填): 程序的二进制名称，可以通过 `readlink -f /proc/${pid}/exe` 获取
- 启动参数匹配规则(非必填、重点): 实现精确匹配进程的重要规则，主要应用于如 Java 或 Python 等二进制名重复的进程，Agent 端采集器会通过该规则去识别进程 `cmdline` 符合参数的唯一进程。如果填入的参数不唯一就会识别到多个进程，如果进程二进制名本身唯一，可不填
- 绑定 IP(非必填): 进程实际监听的 IP 地址，可不填，如果填必须是准确对应 IP 的地址，如 IP 错误，采集器会无法匹配进程导致数据上报异常
- 端口(非必填): 同“绑定 IP”
- 协议(非必填): 同“绑定 IP”

如 druid-broker 这个 java 程序：

```
root@druid-public01:~# ps -ef | grep broker
root    4349  4191  0 11:07 pts/2    0:00:00 grep --color=auto broker
root    4367  26308  1 01  ?    04:10:   0 java -server -Xms24g -Xmx24g -XX:MaxDirectMemorySize=32g -XX:+ExitOnOutOfMemoryError -Duser.timezone=UTC -Dfile.encoding=UTF-8 -Djava.io.tmpdir=/tmp -Djava.util.logging.manager=java.util.logging.SimpleFormatter -Djava.logging.log4j.jul.LogManager -cp /data/mapleleaf/druid/conf/druid/cluter/query/broker /data/mapleleaf/druid/conf/druid/cluter/query/_common:/data/mapleleaf/druid/conf/druid/cluter/query/_common/hadoop-xml:/data/mapleleaf/druid/conf/druid/cluter/query/_common/hadoop-xml:/data/mapleleaf/druid/bin/../* org.apache.druid.Main server broker
root@druid-public01:~# ps -ef | grep broker
```

### ✓ 基础信息

进程名称 \* ⓘ

进程别名 \* ⓘ

进程启动参数 ⓘ

备注

### ✓ 监听信息

绑定IP

端口 ⓘ

协议

### > 进程管理信息

保存

取消

更多其他进程配置方法请查看[各种进程配置方法](#)

## 服务模版

“

服务实例：如果进程的端口，路径等信息都是一样的，建议使用服务模版，并且设置服务分类。方便后面的管理工作。如果是端口路径不一致也可以自动添加实现。

”

服务模板配置 服务模板实例

基本属性

模板名称：druid\_broker 服务分类：中间件 / broker

服务进程

+ 新建进程 模板中第一个进程默认为关键进程，服务实例化后的名称会包含此进程的基本信息

进程名称	进程别名	监听IP	端口	启动路径	启动用户	操作
java	java-broker	0.0.0.0	8082	--	--	<a href="#">编辑</a> <a href="#">删除</a>

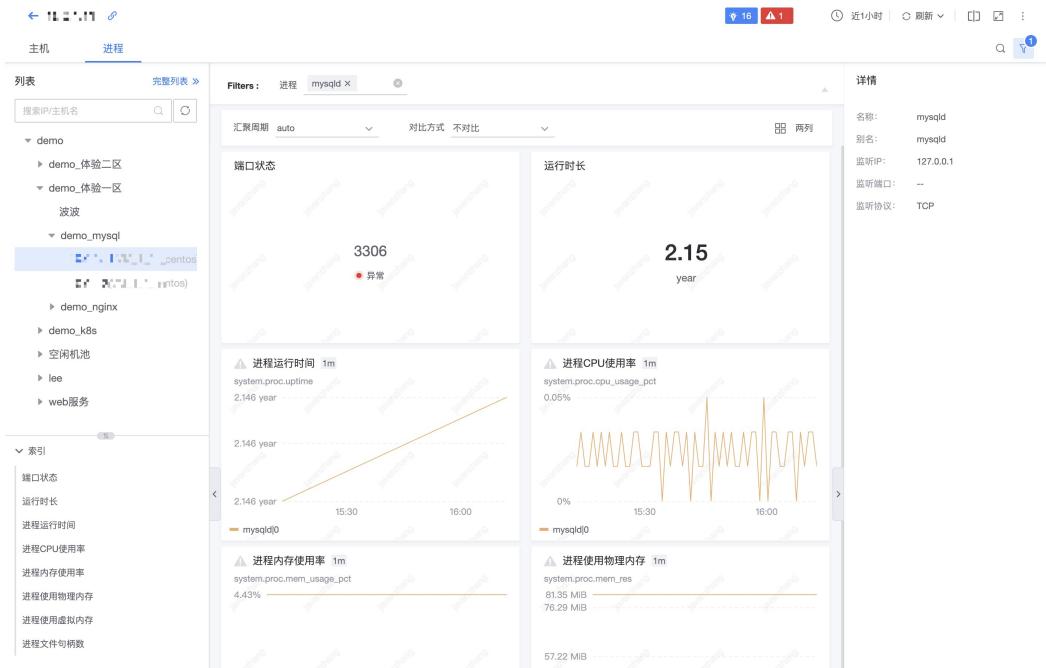
自动下发：

在配置平台上对进程的修改，会通过配置平台的事件推送功能自动下发至 Agent 上的 `/etc/gse/host/hostid`，监控的进程端口采集器会捕获文件变化，做异常检测，并且将内容更新到采集器中，下发时间预计在 2 分钟内，页面进程端口更新信息在 5 分钟内。

## 第二步：主机监控查看进程监控



- 进程：正常时标绿显示
- 端口：点击进程，正常时标绿显示，端口不存在以红色显示，被占用则是黄色，灰色说明采集器未上报数据，需要按照数据未上报流程检查
- 资源/性能：正常时图表有连续不断的数据
- 检查进程是否精确匹配：
- 如果是二进制为 Java 或 Python 的服务，检查进程端口资源是否只上报了配置的进程，如果配置的进程是唯一的，上报进程出现复数，说明精确匹配失败，需要确认匹配参数是否唯一



### 常见问题

1. 在CMDB创建了进程名，却告警进程名不存在：机器上不存在该进程，进程名错误
2. 已在CMDB中关闭了端口，但告警进程不存在：如在CMDB中的主机上配置了进程管理，则代表机器上应该存在该进程，必须准确。即使在端口已经false的情况下，仍会报进程不存在。
3. 当前进程绑定在ipv6的IP上，但由于CMDB未支持ipv6，会导致误告，待CMDB支持ipv6

4. CMDB中配置了正确的进程端口，但仍然报告进程端口不存在，请查看/var/lib/gse/host/hostid 是否与CMDB中的一致，如不一样，是同步延迟导致

5. 相关配置位置

``` 进程相关配置文件

在CMDB创建进程后，相关配置文件会写到/var/lib/gse/host/hostid 以及 /usr/local/gse/plugins/etc/bkmonitorbeat.conf中，如下所示

/var/lib/gse/host/hostid  
windows的路径为c:/gse/data/host/hostid

/usr/local/gse/plugins/etc/processbeat.conf  
windows的路径为 c:/gse/plugins/etc/bkmonitorbeat.conf

进程采集插件

进程信息通过bkmonitorbeat插件采集，如进程不存在，则无法采集进行端口信息

**ps -ef |grep bkmonitorbeat**

./bkmonitorbeat -c ..etc/bkmonitorbeat.conf

**ps -ef |grep bkmonitorbeat**

./bkmonitorbeat -c ..etc/bkmonitorbeat.conf

...

### 第三步： 配置告警策略

如果监控进程的指标，默认会采集如下进程的指标**主机-进程-指标**

### 相关文档

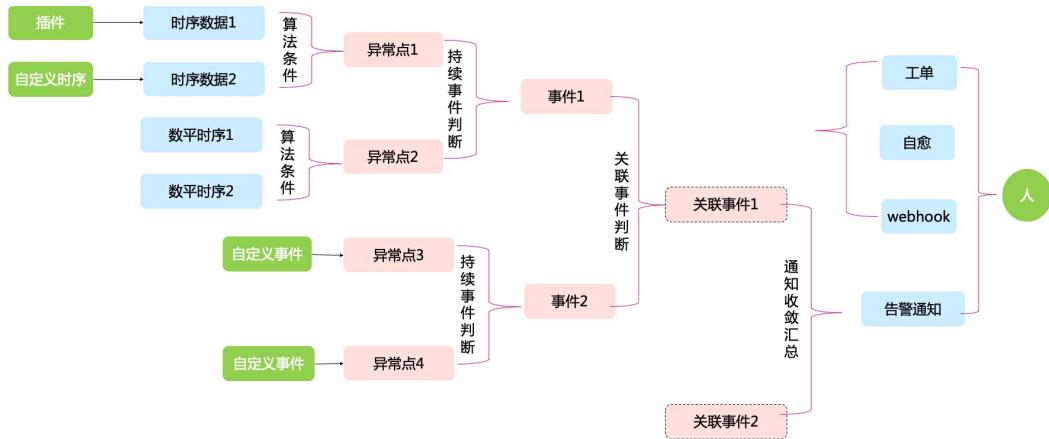
更多进程相关的采集查看场景 [如何实现多实例采集](#)

## 事件中心

事件中心记录的是所有的告警事件，及告警通知情况。可以在事件中心进行基本的问题定位和分析。

### 前置步骤

工作原理



了解概念：

- 异常点：时序数据通过算法检测判断有异常的点
- 告警事件：多个异常点为同一类为事件，一般具有持续性
- 告警通知：多个告警事件通过收敛规则，风暴抑制等手段汇总成告警通知
- 处理记录：是执行通知、执行处理套餐后的执行记录信息

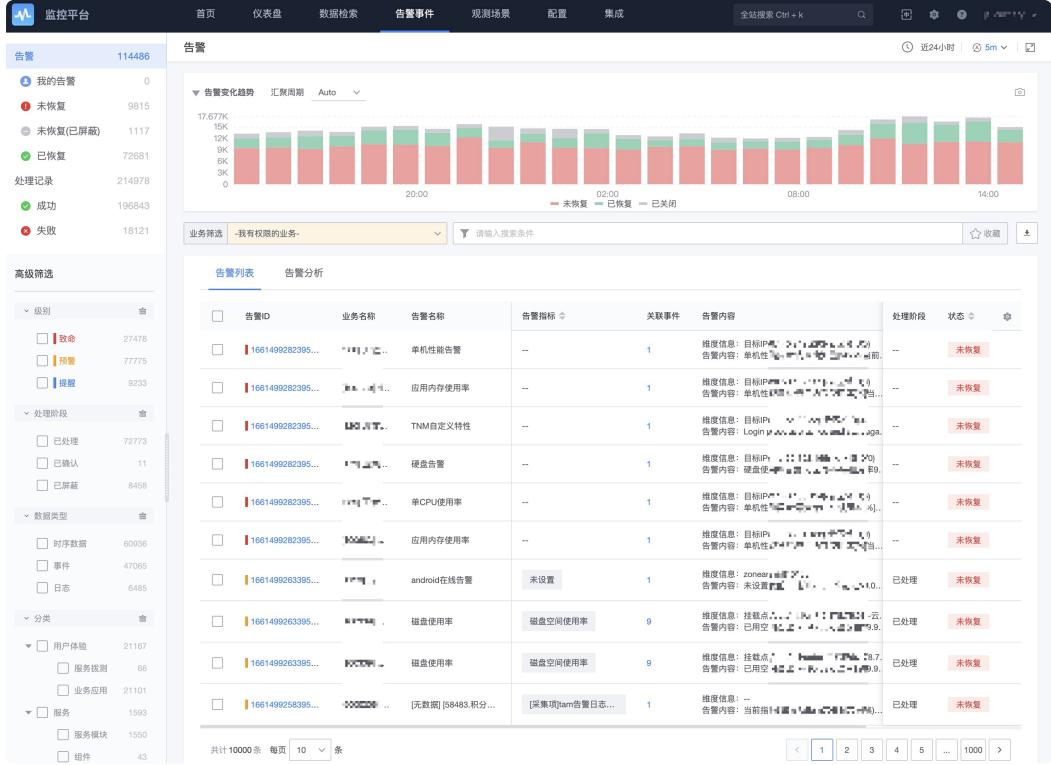
“

更多查看[术语解释](#)

”

## 主功能一览

展示所有的告警事件、关联事件、处理记录等，可以对事件进行查询，分析，告警确认等。



## 告警事件查看

### 告警事件列表

告警事件提供多种查询方法，帮助快速的定位到事件。

#### 通知状态：

- 成功：全部成功
- 部分失败：通知有部分失败
- 失败：通知全部失败
- 通知人为空：配置了告警组但是通知人为空的情况

#### 告警状态：

- 未恢复：事件一直在持续未恢复，需要重点关注
- 未恢复(已确认)：事件一直在持续未恢复，有人已经进行了告警确认
- 未恢复(已屏蔽)：事件一直在持续未恢复，有人已经进行了快捷屏蔽
- 未恢复(已抑制)：是监控的内部逻辑
- 已恢复：事件不再持续，算是历史情况，可以不再重点关注
- 已关闭：当策略删除的情况下，事件直接接入到关闭状态

## 告警处理

告警列表 告警分析

| 告警ID   | 告警名称          | 分类        | 告警指标                    | 关联事件  | 告警内容  | 关联信息                   | 处理阶段 | 状态 | 操作 |
|--|---------------|-----------|-------------------------|-------|---|------------------------|------|----|----|
| <input checked="" type="checkbox"/> 1637250125158931 | [saas]内存使用率告警 | 用户体验-业务应用 | mem_usage               | 10    | 维度信息: docker_name(bk_dataadmin)<br>告警内容: MAX(mem_usage) >= 90.0, 当前值93... | 已选择 2 条 批量确认 批量屏蔽 取消选择 | 已处理  |    |    |
| <input checked="" type="checkbox"/> 16359336501016   | consul服务异常    | 用户体验-业务应用 | Consul critical service | 21915 | 维度信息: --<br>告警内容: Consul critical service >= 1.0, 当前值...                  | 查看更多                   | 已处理  |    |    |

共计 2 条 每页 10 条 < 1 >

## 查询和收藏

蓝鲸

请输入搜索条件

— 未恢复 — 已恢复 — 已关闭

告警列表 告警分析

建议字段

- 告警ID (id)
- 告警名称 (alert\_name)
- 状态 (status)
- 告警内容 (description)
- 级别 (severity)
- 指标 (metric)
- 目标IP (ip)

最近搜索

- 告警名称: 进程
- 告警名称: “进程端口”
- 进程端口
- 告警名称: 进程端口
- 进程
- 重启

收藏

查看data1告警

恢复

恢复

1 >

## 告警分析

-我有权限的业务 -

tag.device\_name: "vda" OR tag.device\_name: "vda1" OR tag.device\_name: "vdb"

告警列表 告警分析 +新增

业务名称 (34)

数据平台(BEDEV) 63 100.00%

蓝鲸 0.00%

逍客晋升机考专用 0.00%

腾讯数据分析平台TQoS 0.00%

蓝鲸企业环境 0.00%

告警名称 (1)

磁盘I/O使用率 63 100.00%

指标ID (1)

I/O使用率 (pk\_monitor.system.io.util) 63 100.00%

持续时间 (3)

小于1h 10 15.87%

大于1h且小于1d 53 84.13%

大于1d 0 0.00%

目标IP (25)

6 9.52%

5 7.94%

3 4.76%

3 4.76%

3 4.76%

云区域ID (1)

0 63 100.00%

策略ID (1)

磁盘I/O使用率 63 100.00%

负责人 (2)

63 100.00%

63 100.00%

服务实例ID

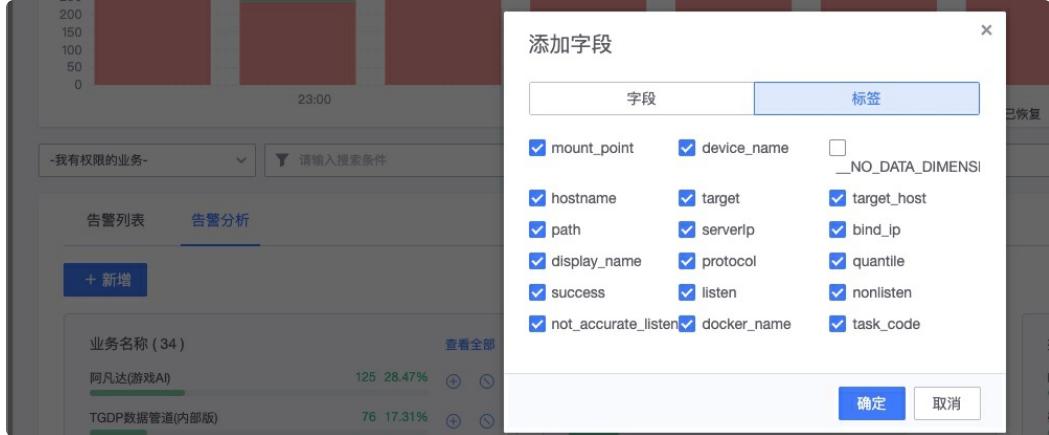
没有数据

device\_name (3)

vda 27 42.86%

vda1 27 42.86%

vdb 9 14.29%



## 告警事件详情

This screenshot shows the 'Alert Details' page. On the left, there's a sidebar with 'Event Center' and 'Advanced Selection' filters. The main area has tabs for 'Alerts' and 'Alert Analysis'. A red arrow points from the 'Alert Analysis' tab to the timeline chart on the right. The chart displays a green line for 'Current' data, a yellow line for '7 days ago', and a blue line for '1 day ago'. The chart shows a sharp rise in values starting around 23:30, with a peak around 23:45. Below the chart, there are sections for 'View Information', 'Handling Experience', 'Flow Record', and 'Associated Events'. A red circle with a '1' indicates an unhandled alert. A button for 'Quick Shield' and 'Alert Confirmation' is also visible.

- 【1】查看关联的告警策略，查看的是快照，有可能当前的策略已经被修改或者被删除
- 【2】告警处理
  - 快捷屏蔽：快捷的屏蔽该事件，后续再出现类似的事件在屏蔽时间内也不再进行通知
  - 告警确认：该告警事件快速确认，该事件恢复前不再进行通知。恢复后再出现相同的事件时还是会再进行通知
- 【3】通知状态详情：可以看到每次通知的情况
- 【4】视图信息：展示出现首次异常时间到结束时间点的数据状态。日志，事件类的展示方式不一样
- 【5】处理建议：简单的记录事件的处理解决方案，和指标进行关联
- 【6】流转记录：详细的事件流转过程，主要用于单一事件追踪定位

## 告警关联信息

平台会将告警的事件内容进行分析其关联的内容：所在的主机、关联的日志、关联的trace、关联的场景。可以满足快速的查看，也可以快速的进入跳转到具体内容进行查看。

告警详情 [更多](#)

新开页 [更多](#)

预警 [更多](#)

告警ID: 166162269939769813 告警名称: 磁盘IO使用率 [策略详情](#)

首次异常时间: 2022-08-28 01:46:00 处理状态: 1次(1次成功) [更多](#)

告警产生时间: 2022-08-28 01:51:39 (持续4m) 处理阶段: 已处理 [更多](#)

维度信息: 设备名(sda) - 目标IP: [更多](#)

告警内容: IO使用率 >= 80.0%, 当前值80.2747%

关联信息: 集群 [更多](#) \* 日志 [更多](#) \* 关联日志

[已恢复](#) [更多](#)

[快捷屏蔽](#)

视图信息 日志 主机 处理经验 流转记录 关联事件

IO使用率  
bk\_monitor.system.io.util

98%  
98% 捅值算法

0% 00:50 01:00 01:10 01:20 01:30 01:40 01:50 02:00

当前 1天前 7天前

问题定位

当通知没有收到的情况可以查看“通知状态详情”。

## 通知状态明细

X

通知次数 : 第4次 ( 2018-10-29 10:32:33 ) ▾

| 通知人                | 短信 | 邮件 | 企业微信 | 微信 | 钉钉 | 电话 |
|--------------------|----|----|------|----|----|----|
| root ( 运维团队 )      | ✗  | ✓  | ✓    | ✓  | ✓  | ✓  |
| admin ( 运维团队 )     | ✓  | ✓  | ✗    | ✓  | ✓  | ✓  |
| jingzheng ( 运维团队 ) | ✓  | ✓  | ✓    | ✓  | ✓  | ✓  |
| zhengzhu ( 运维团队 )  | ✓  | ✓  | ✓    | ✗  | ✓  | ✓  |
| root ( 运维团队 )      | ✓  | ✓  | ✓    | ✓  | ✓  | ✓  |
| lengtang ( 运维团队 )  | ✓  | ✓  | ✓    | ✓  | ✓  | ✓  |
| huayang ( 合作方 )    | ✓  | ✓  | ✗    | ✓  | ✓  | ✗  |
| user   合作方 )       | ✗  | ✓  | ✓    | ✓  | ✓  | ✓  |
| lengtang ( 合作方 )   | ✓  | ✓  | ✓    | ✓  | ✓  | ✓  |
| qihuna ( 合作方 )     | ✓  | ✓  | ✓    | ✓  | ✓  | ✓  |

报错信息...

告警通知被收敛了的细节可以查看流转记录。

| 视图信息  | 处理建议 | 流转记录 |
|---|------|------|
|   |      |      |
| ① 🔍 2020-01-21 13:58:44 至 2020-01-21 15:14:21<br>【告警收敛】 avg(load5) >= 0.0, 当前值2.6                   |      |      |
| 🔔 2020-01-21 13:58:41<br>【告警通知】 已达到发起告警通知的条件 (告警未恢复及未关闭, 每隔120分钟发送一次通知), 接收通知的告警组 bond, 通知状态 点击查看明细 |      |      |
| ⌚ 2020-01-21 13:58:41<br>【告警触发】 avg(load5) >= 0.0, 当前值2.45 达到了触发告警条件 (5分钟内满足1次检测算法)                 |      |      |
| ⊕ 🔍 2020-01-21 13:58:44 至 2020-01-21 13:58:44<br>【告警收敛】 avg(load5) >= 0.0, 当前值2.77                  |      |      |

## 单指标智能异常检测

AIOps相关的能力依赖蓝鲸AIOps平台，蓝鲸监控对接了AIOps平台的场景服务，在AIOps平台添加一个模型就可以直接在蓝鲸监控使用。

## 适用的数据

1. 数据类型：指标数据，支持多维度的指标数据
2. 数据来源：监控采集，默认上报的，计算平台tspider的可以在监控查看到的数据。自定义上报和日志的数据不包括在内。

## 配置方法

1. 监控数据中选择添加监控指标。



The screenshot shows the 'New Strategy' configuration interface. In the 'Monitoring Data' section, the 'Metrics' tab is selected and highlighted with a red box. Other tabs available are 'Events', 'Log Keywords', and 'Associated Alarms'. The 'Metrics' tab has sub-options like 'Time Series Data' and 'Multi-dimensional Metrics Matching'.

2. 模型名称中选择“通用智能异常检测模型”。



The screenshot shows the 'Monitoring Data' configuration interface. The 'Metrics' tab is selected. In the 'Detection Rules' section, the 'General Intelligent Abnormal Detection Model' is selected in the dropdown menu. A red box highlights this selection.

3. 配置希望检测的异常方向：向上、向下。

4. 配置敏感度，如果不清楚可以先不动，等有效果之后再根据告警情况进行调整，敏感度越高命中的告警更多，敏感度越低命中的告警越少。PS：配置完成后有一段时间的延迟，并且智能学习有一定学习过程。

### ▼ 检测规则 (通过查询的数据按检测规则判断是否需要进行告警)

同级别的各算法之间是 且 或 的关系

▲ 致命 智能异常检测

模型名称: 通用智能异常检测模型

下降异常:  ⓘ

上升异常:  ⓘ

敏感度:  较少告警 Ⓢ 较多告警

5. 接收到告警后, 如果不准确, 可以在告警详情中进行反馈, 模型会定期持续训练

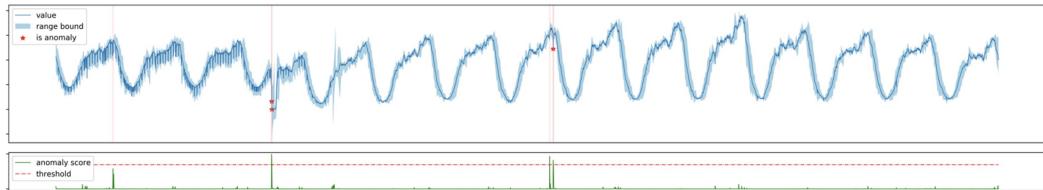
The screenshot shows the monitoring platform's alert management interface. On the left, there's a sidebar with various monitoring categories like Application, Data, and Metrics. The main area is divided into '事件' (Events) and '告警' (Alerts). The '告警' section lists several alerts, each with a unique ID, name, and status. One alert is highlighted with a red border. A red arrow points from this highlighted alert to its detailed view on the right. The detail view includes fields for告警ID (16527144031985882), 告警名称 (request\_cnt智能异常模型), 处理状态 (已处理), 处理时间 (2022-09-17 09:21:10), and告警内容 (request\_cnt智能异常模型识别异常, 异常类型: 上升异常, 异常分值: 0.999998, 具体值: 1.0). Below this, there are two line charts: 'request\_cnt' and '异常分值'. The 'request\_cnt' chart shows a sharp peak at approximately 09:21:10 on 2022-09-17. The '异常分值' chart shows a single data point at 1.000000 at the same time.

## 适用的数据场景

智能异常检测适用于大部分时序指标, 特别适用于以下三种场景。

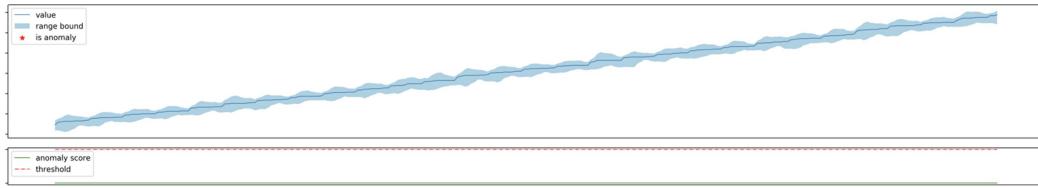
### 场景一：指标呈现周期性波动

当指标呈现周期性波动时, 难以设置合理的静态阈值来检出图中明显的异常点, 该场景下使用智能异常检测, 既能保证检测的准确性, 也能避免重复告警对用户的骚扰。



### 场景二：指标呈现趋势性增长或下降

当指标呈现合理的增长或下降趋势时, 使用静态阈值会在指标持续下降/上升期间被检测为异常; 该场景下使用智能异常检测, 能够自适应曲线的下降/上升趋势, 当且仅当指标变化幅度明显过大时才被检测为异常。



### 场景三：指标呈现突增或突降趋势

当指标呈现无法预期的突增或突降变化时，难以设置合理的静态阈值识别出曲线的突增点，该场景下使用动态异常检测，则可自动捕捉不同程度的突增或突降变化，对于指标变化幅度明显过大时会被检测为异常。



## 告警处理之处理套餐

在告警策略中，有告警的通知发送功能，同时也具有告警的处理功能。告警处理是由具体的处理套餐去执行动作的，通过HTTP回调，作业平台，标准运维，流程服务等系统，实现在产生告警后，对告警进行自动处理的能力。

使用告警处理功能，有2个步骤：

1. 定义处理套餐，表示具体的任务执行逻辑。
2. 在告警策略中使用套餐。

处理套餐的功能如下所示。

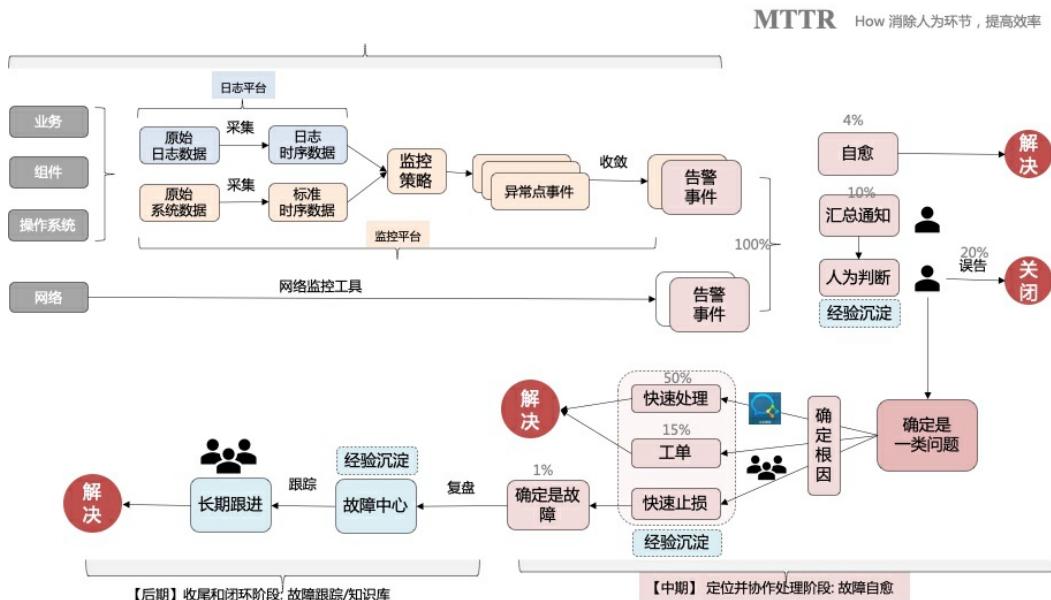
| ID     | 套餐名称         | 套餐类型   | 关联策略 | 触发次数(近7天) | 更新人   | 更新时间                  | 启/停                                 | 操作                                    |
|--------|--------------|--------|------|-----------|-------|-----------------------|-------------------------------------|---------------------------------------|
| #55290 | 磁盘空间自愈处理     | 作业平台   | --   | --        |       | 2022-08-17 10:02:4... | <input checked="" type="checkbox"/> | <a href="#">编辑</a> <a href="#">删除</a> |
| #49400 | icontest     | 一键拉群   | --   | --        | in    | 2022-05-20 16:56:3... | <input checked="" type="checkbox"/> | <a href="#">编辑</a> <a href="#">删除</a> |
| #48820 | 「快播」... [默认] | 一键拉群   | --   | --        | admin | 2022-05-20 10:13:1... | <input checked="" type="checkbox"/> | <a href="#">编辑</a> <a href="#">删除</a> |
| #36828 | 【快播套餐】微治发... | 标准运维   | --   | --        | admin | 2022-01-20 19:29:2... | <input checked="" type="checkbox"/> | <a href="#">编辑</a> <a href="#">删除</a> |
| #36827 | 【快播套餐】微治发... | 标准运维   | --   | --        | admin | 2022-01-20 19:29:2... | <input checked="" type="checkbox"/> | <a href="#">编辑</a> <a href="#">删除</a> |
| #36826 | 【快播套餐】转移主... | 标准运维   | --   | --        | admin | 2022-01-20 19:29:2... | <input checked="" type="checkbox"/> | <a href="#">编辑</a> <a href="#">删除</a> |
| #999   | 「快播」... [默认] | 内置授权套餐 | --   | --        | admin | 2021-12-22 10:34:2... | <input checked="" type="checkbox"/> | <a href="#">编辑</a> <a href="#">删除</a> |

告警策略中使用处理套餐。

## 故障自愈产品功能简介

故障自愈是故障生成和故障自动化处理的解决方案，提升企业的服务可用性和降低故障处理的人力投入，实现故障从“人工处理”到“无人值守”达到服务自愈的目的。

通过自动化处理节省人力投入，通过预定的恢复流程让恢复过程更可靠，通过并行分析达到更快的故障定位和恢复。



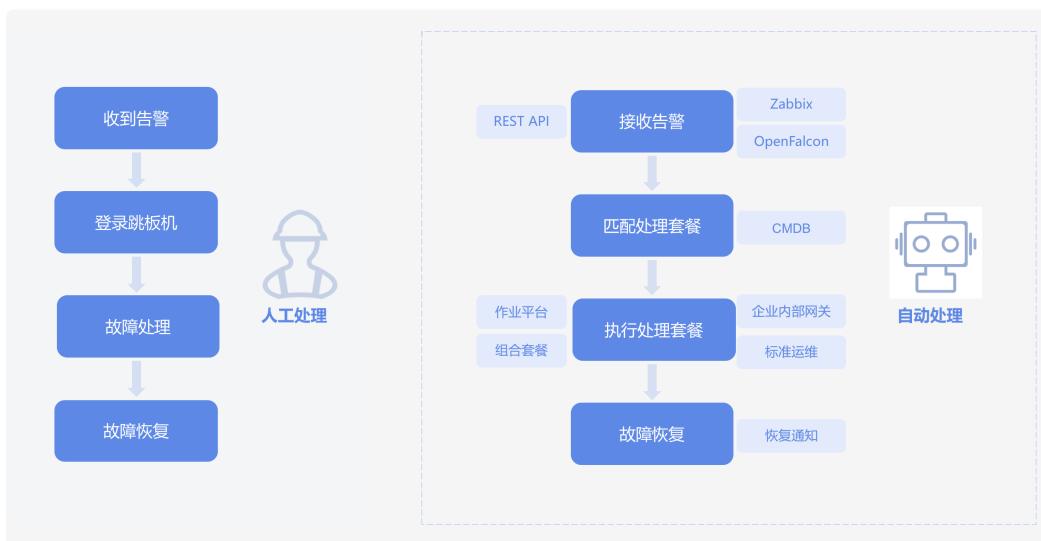
故障自愈是在解决运维中期的过程，目的是消除人为环节，降低MTTA和MTTR的过程。

## 特点及优势

1. 集成主流监控产品：告警源集成蓝鲸监控、4 款主流开源监控产品 Zabbix、Open-Falcon、Nagios、Icinga，及 AWS、邮件的告警接入，更能通过 REST API 拉取、推送告警。
2. 丰富的处理套餐：除支持作业平台、标准运维外、ITSM；还支持快捷套餐类(磁盘清理、汇总、检测 CPU 使用率 TOP10 等)；还支持自定义系统。
3. 告警生成：可以丰富告警事件，可以对告警事件进行识别进行去重生成最终告警，减少重复事件的干扰。
4. 灵活的事件中心：事件中心有告警，告警关联的事件，还有处理记录，可以帮助用户快速的回溯问题，协助发现定位问题。
5. 处理记录：记录每次执行动作，并且可以通过对处理记录进行分析。

## 引领行业故障处理新潮流

故障自愈重新定义故障处理流程，在运维领域系较早提出故障自动化理念并落地为产品。



## 事件处理流程引擎，实现无人值守自愈

获取监控告警发现异常，预诊断分析，调用预定义的处理流程，实现故障无干预自动处理。



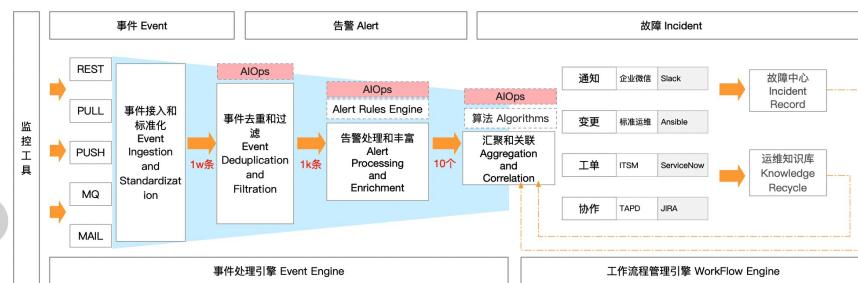
## 为企业节省人力及降低 MTTR

故障自愈通过自动化的故障处理流程，节省运维处理故障的人力成本。让运维把工作专注到企业服务的用户体验优化、数据分析，而不是基础运维服务。

自动化的故障处理流程，减少人工处理环节耗费的时间，降低故障处理时长。实现故障的无人值守，让故障处理不再依赖于人，提升企业 IT 服务的可用性。



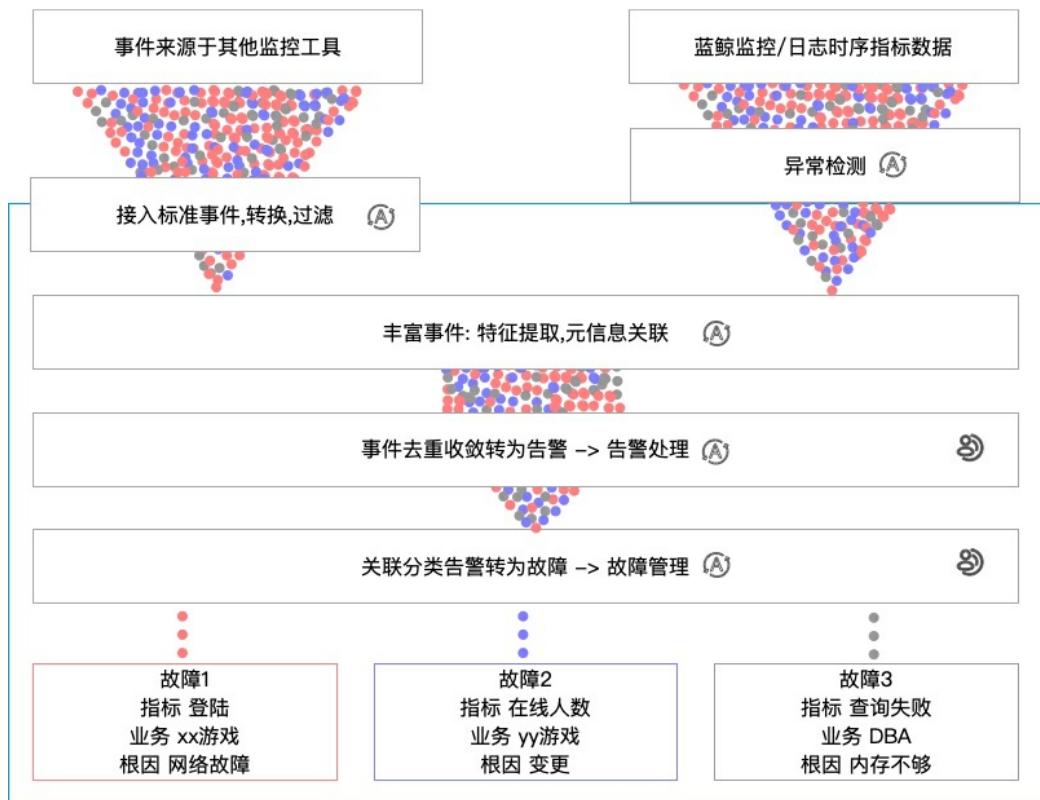
## 故障自愈产品架构图



故障的工作过程：

1. 事件接入：能够接收不同种类的告警事件，提供PUSH PULL等方式，支持主流的告警系统，也可以支持用户自定义接入扩展事件插件。
2. 事件处理引擎：接收到的事件会进行事件的标准化，丰富，去重等形成告警，通过对告警进行关联分析形成最后的故障，并且对故障进行根因分析进一步智能提供信息帮助运维决策。
3. 工作流管理引擎：对告警和故障可以通过工作流管理，联动周边的系统，可进行通知、JOB执行、标准运维流程、服务工单、告警回调等方式（还可以自定义服务）触发更多系统协作最终达到减少人为介入，甚至是服务自愈的目的。

## 蓝鲸监控与故障自愈的关系



故障自愈的能力是蓝鲸监控的能力的一部分，来自蓝鲸监控的数据对于分析和处理更有利，但并不需要指标数据和日志数据的处理时，只需要对接第三方告警系统，故障自愈也完全可以独立存在。

1. 部分功能模块是共用的，事件中心、处理套餐、告警组、屏蔽、告警策略（事件告警，关联告警）
2. 监控平台包含故障自愈的所有能力，故障自愈是监控平台的一个子系统
3. 权限是共用的，权限使用的是监控平台的权限。

## 开启 Kubernetes 监控

### 一、相关名词解释

- Kubernetes: 容器管理调度平台，参考什么是kubernetes。
- BCS: 蓝鲸容器服务平台（BCS, Blueking Container Service），是高度可扩展、灵活易用的容器管理服务平台。参考BCS使用文档。
- gse\_agent: 运行在主机（如CVM、物理机等）上的蓝鲸agent, 是监控数据上报的通道。
- bkmonitorbeat: 指标采集器，用于采集指标数据。bkmonitorbeat 部署会依赖 gse\_agent, 如主机上未安装 gse agent 会导致 bkmonitorbeat 启动失败（pod 处于 init 状态）。
- bkmonitor-operator: 用于监听 serviceMonitor、podMonitor、probeMonitor 等配置，并负责刷新各个 bkmonitorbeat 配置文件。
- dataid: 在监控平台内部用于标识数据的id。

## 二、什么是容器监控

容器监控，泛指在容器场景下，对容器环境的指标、事件等进行上报。

目前容器管理平台，基本被kubernetes所统一，此处所指的容器监控，其对象均指运行在kubernetes环境下的容器，对于不通过kubernetes纳管的容器监控场景暂不支持，后文不再单独说明。

容器环境下的监控特点：

- 监控目标是动态的，无法通过传统的方式对指定目标IP，目标端口进行监控。
- 容器会随时销毁，创建，监控目标变化非常快，没有监控无法溯源监测。
- 无需关心容器运行在什么机器上面。
- 容器的数量多，上报的指标量级多。

容器监控的核心诉求,主要分为4类:

1. 集群本身的运行状态监控，监控的目的是随时关注容量、及时发现异常，让集群的运维者能够快速修复集群问题。
2. 对kubernetes集群状态的监控，如master集群的etcd,kube-api, kube-scheduler,kube-proxy,kubelet等核心服务的监控。
3. 对node节点状态的监控，如CPU,内存，网卡等监控。
4. 对资源的统计需求，如cluster, NameSpace, node, pod数量的统计。
5. 对workload、service、pod、container运行指标和状态监控。
6. 容器内的事件、日志的监控。

容器的日志和事件采集，并能够配置告警。3. 自定义指标的监控，运行在容器中的服务，自身提供的服务指标是否正常，因为大多数情况下，运行环境的指标正常，并不等于服务本身就是正常的，所以需要上报服务本身的指标。

服务的指标，如接口的成功，失败，饱和度，错误率等，通常这类指标对应用开发来说，更具有实际参考价值，更能够对业务有帮助。4. HPA功能，对pod数量动态扩缩容。

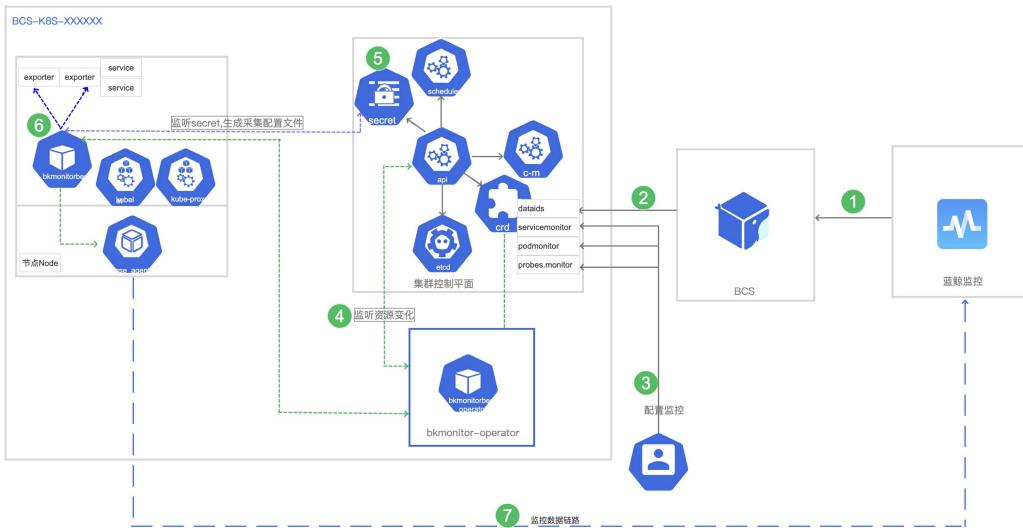
基于关键指标，对容器的pod数量进行增减，达到资源的有效利用。在云原生社区中，kubernetes各种监控方案尝试中，几乎都是推荐prometheus。原生的prometheus，在数据维度组合较小情况下，能够很好的工作，满足日常的使用需求，但在数据维度较多的情况下，即使用了thanos方案，依然效果不理想。具体表现是，在指标和维度组合数较大的情况下，占用内存比较高，查询速度会变慢，遇到服务重启，恢复时间较长，监控的稳定性受到极大挑战。

### 三、容器监控的方案

为了让大家统一集中管理监控，将容器监控与主机监控，业务监控等融为一体，监控平台推出了容器场景的监控能力。可以解决以下问题：

- 避免prometheus服务在高负载情况下OOM，查询无响应，可用性不高问题。
- 无需每个kubernetes集群独立部署prometheus。
- 解决容器监控和业务监控割裂使用的问题，集中式查看监控、告警，让监控数据在统一的地方消费使用，无需同时维护多套监控系统。

#### 数据采集流程



整个架构对比 prometheus-operator 来讲，我们少了 prometheus 组件，多了 bkmonitorbeat daemonset，因为我们的 bkmonitorbeat 可以替代 prometheus 的采集能力，而通过 gse 管道能力我们可以使用蓝鲸监控的存储和查询能力，也就是说，用户不需要在集群中部署Prometheus相关组件和prometheus-operator。

得益于蓝鲸监控中心化存储，bkmonitor-operator 较 prometheus-operator 方案，对集群内资源消耗更低。

#### 数据采集流程说明

1. 用户集群接入到蓝鲸监控后，生成 dataid 资源并应用到用户集群中。
2. operator dataid-watcher 监听到有 dataid 资源创建后，启动 serviceMonitor/podMonitor/Probe 检测器，实时检测集群中的监控对象。
3. operator 根据检测到的资源对象生成采集任务，并为每个节点创建一个 Secrets 资源记录采集任务内容。
4. 节点上的 bkmonitorbeat-reloader 监测到自己节点上的 Secrets 对象有变更，则由 reloader 给 bkmonitorbeat 发送 reload 信号，通知 bkmonitorbeat 重载配置。
5. bkmonitorbeat 感知到有采集任务，进行数据采集上报。

## 采集方式的区别

1. 原Prometheus是由Prometheus-server集中拉取，而蓝鲸监控是通过本机的node去采集，采集压力分摊到每个node，所以不存在集中式的采集任务压力。
2. 在每个node上部署bkmonitorbeat的daemonset，该daemonset采集本机的pod中的exporter数据
3. 集群中的数据，由kube-state-metrics，prometheus-node-exporter，kubelet原生提供，由本node的bkmonitorbeat采集
4. 支持多集群的采集，多个集群可以共用一套蓝鲸监控

## 四、如何接入容器监控

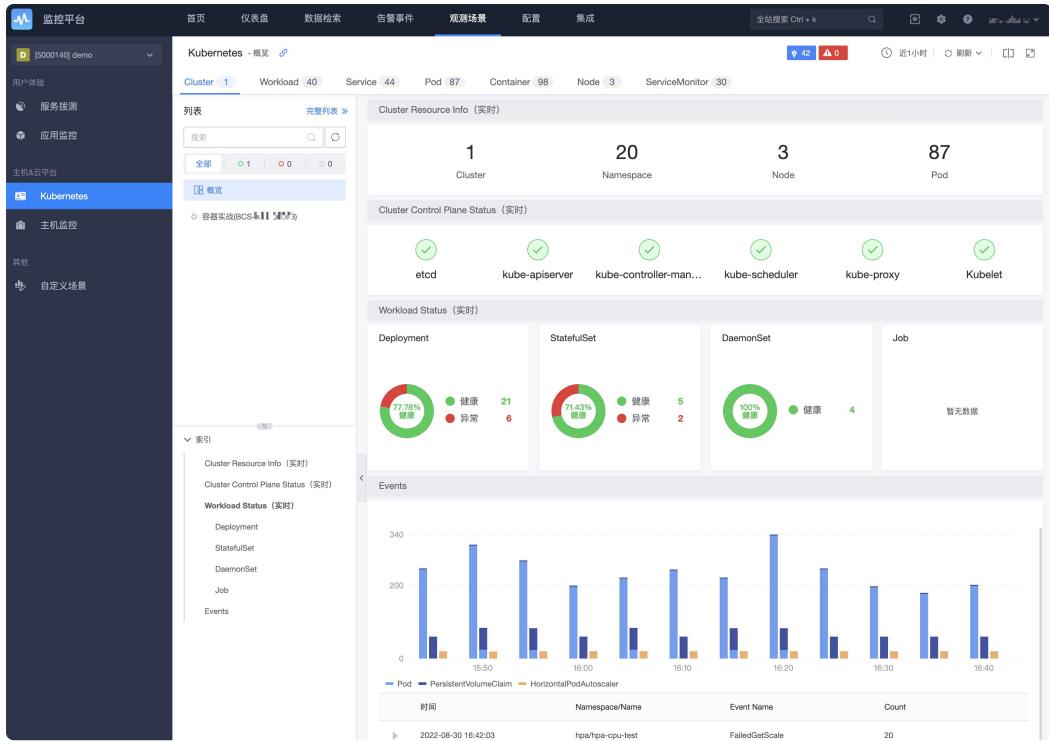
接入容器监控的集群，分2种情况

- 在BCS集群中已经接入的k8s集群，则可以直接在BCS的集群组件中安装bkmonitorbeat operator
- 集群未被BCS纳管，则需要将k8s集群托管，请参考[集群管理](#)，然后再安装bkmonitorbeat operator

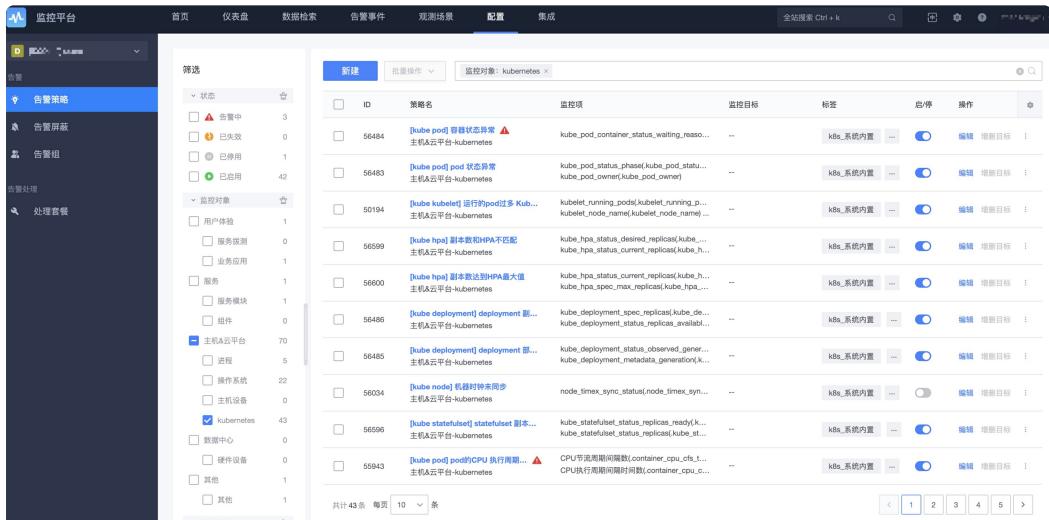
接入的过程，请参考[蓝鲸容器监控安装](#)

## 五、如何查看容器监控数据

观测场景 -> kubernetes



## 六、如何配置容器监控告警



有内置的一些策略，具体查看[容器默认策略说明](#)

## 七、servicemonitor、podmonitor数据如何接入

容器监控功能，兼容Prometheus的servicemonitor, podmonitor功能，采集目标的定义请参考[文档](#)

与Prometheus生态的区别是，不需要部署Prometheus-operator和Prometheus，数据上报链路统一走蓝鲸的数据管道。

每个node上面的数据，在本node上面的bkmonitor-operator的daemonset采集，而非集中式采集，因此，采集开销是在daemonset上面。

...

## 查看bkmonitor-operator

```
kubectl get pods -n bkmonitor-operator -o wide ````
```

| NAME  | READY | STATUS  | RESTARTS | AGE |
|---|-------|---------|----------|-----|
| bkmonitor-operator-stack-bkmonitorbeat-daemonset-fxtxq①                   | 2/2   | Running | 0        | 83m |
| bkmonitor-operator-stack-bkmonitorbeat-daemonset-kkskq②采集本机/2ode的数据       | 2/2   | Running | 0        | 83m |
| bkmonitor-operator-stack-bkmonitorbeat-daemonset-pvpmh③                   | 2/2   | Running | 0        | 83m |
| bkmonitor-operator-stack-kube-state-metrics-84fd994fc4-9fr75④采集/3s数据      | 2/2   | Running | 0        | 83m |
| bkmonitor-operator-stack-operator-deployment-b96f69d8f-x9ptl⑤kmd/1or-oper | 1/1   | Running | 0        | 83m |
| bkmonitor-operator-stack-prometheus-node-exporter-9dltx⑥                  | 1/1   | Running | 0        | 83m |
| bkmonitor-operator-stack-prometheus-node-exporter-hcpcz⑦                  | 1/1   | Running | 0        | 83m |
| bkmonitor-operator-stack-prometheus-node-exporter-t4sjl⑧                  | 1/1   | Running | 0        | 83m |
| bkmonitor-operator-stack-system-collect-fb6c88b87-h974s⑨事件采集              | 3/3   | Running | 0        | 83m |

...

## 查看daemonset的配置

```
kubectl get ds -n bkmonitor-operator bkmonitor-operator-stack-bkmonitorbeat-daemonset -oyaml ````
```

```
imagePullPolicy: IfNotPresent
name: bkmonitorbeat
resources:
  limits:
    cpu: "1"
    memory: 1Gi
  requests:
    cpu: 250m
    memory: 256Mi
terminationMessagePath: /dev/termination-log
terminationMessagePolicy: File
volumeMounts:
- mountPath: /data/bkmonitorbeat/config
  name: bkmonitorbeat-global-config
```

最高使用CPU1核，内存1G，如果需要采集的数据多，可以调整此配置。

## 八、如何绘制仪表盘-统一监控视角

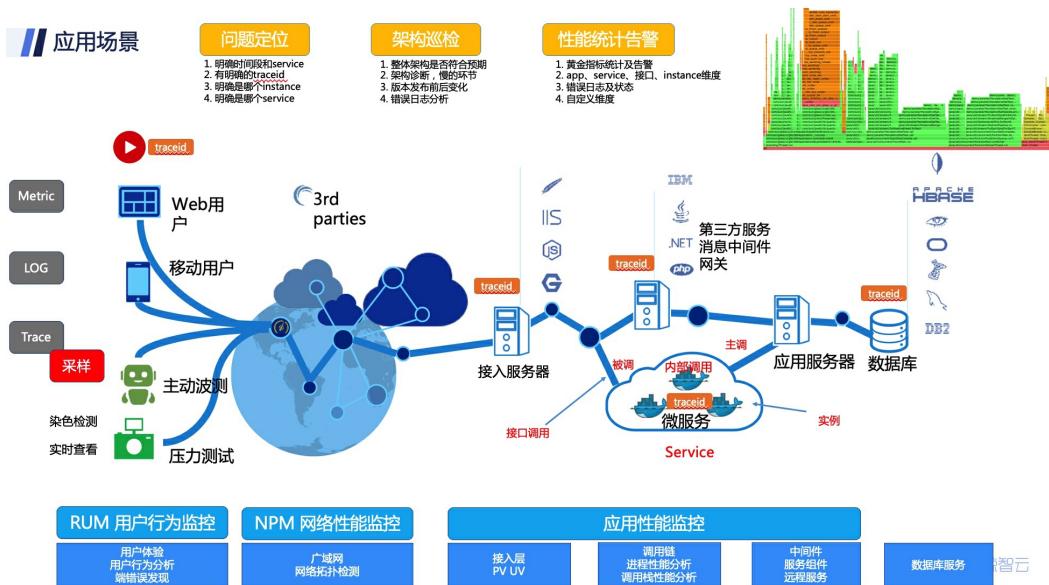
容器的监控指标，可以在仪表盘中进行展示和分析，可以自定义视图查看

## 开启APM监控

观测场景-应用监控

### 什么是APM

APM (Application Performance Monitoring) 即应用性能监控，通过应用软件之间的服务调用来分析软件的问题，可以更细粒度的监控和满足用户更好的体验。APM具体要解决三方面的问题



### 用户排障流程

用户已经明确知道有故障，进行精准的问题复现和定位。

- 第一种：只知道有问题的时间段和service，需要进行范围缩小和上下游查找。调用链查询方式。  
(内部错误的筛选项)

- 第二种：已经拿到明确的tracid进行具体的查看，有可能拿到的traceid断链了，或者没有采集上来。
  - 需要进行染色（明确哪个traceid必需采集上来，或者具体的某个userid，vip类型必需采集不丢弃）
  - 实时进行分析（不采样，不丢数据5分钟）
- 第三种：明确是哪个instance有问题，开始进行反向查看，具体问题在哪？
- 第四种：明确是知道哪个service有问题，再对service进行查看包括instance

## 常规巡检流程

用户还不知道自己的服务状况，相关监控还未完善，或者监控告警之后需要进行整体的查看。

- 查看整体的情况，看是否符合预期。拓扑、黄金指标、TOP的数据。
- 对架构的诊断：Application拓扑、慢的环节
  - SQL分析
    - 热key、大key(response的大小)
- 版本发布后查看版本的变化
- 报表（用户定义用户关心的内容）
- 错误日志统计、日志聚类、日志分析

## 主动告之问题

用户接入还未有人投诉，通过指标数据和智能监控数据统计主动告之用户系统状况

- application维度
- service维度
- 接口维度
- instance维度
- 错误日志
- 自定义维度

## 相关概念

### 业务 BUSINESS

管理和资源的物理隔离，配置平台中最基本命名空间

### 应用 APPLICATION

应用一般是拥有独立的站点，由多个Service共同组成提供完整的功能，拥有独立的软件架构。从技术层

面来说是Trace数据的存储隔离，在同一个应用内的数据将进行统计和观测。

## 服务 SERVICE

微服务中独立的模块，独立完成一个功能，一般可以认为相同的程序构成的为一个Service，那么构成的每个进程为服务实例。

## 接口 RESOURCES

应用程序中处理一次独立的访问请求的过程，包括了HTTP的请求，DB的查询，中间件操作等等。

## 错误 ERRORS

当一个异常的发生导致了此次的接口请求为错误，在字段中会记录错误的状态，`span.status_code`，所以Span中有日志不代表一定是错误，也不是Status是错误就一定有日志，两者没有绝对关系，主要看输出的设置。不过建议按规范使用status\_code，否则平台无法统计出相应的内容。

## TRACE & SPAN

核心的两个概念和数据，就是基于TracelD和Span的数据信息进行的汇总统计。

- Trace：记录经过分布式系统的请求活动，一个trace是spans的有向无环图
- Span：一个trace中表示一个命名的，基于时间的操作。Spans嵌套形成trace树。每个trace包含一个根span，描述了端到端的延迟，其子操作也可能拥有一个或多个子spans。

## OPENTELEMETRY

OpenTelemetry合并了OpenTracing和OpenCensus项目，提供了一组API和库来标准化遥测数据的采集和传输。OpenTelemetry提供了一个安全，厂商中立的工具，这样就可以按照需要将数据发往不同的后端。蓝鲸监控的APM的SDK完全兼容Opentelemetry的SDK即能力。

## 关系

- 一个业务有多个应用
- 应用之间有数据隔离
- 一个应用包括了N个Service
- 一个Service包括了N个服务实例

## APM应用创建

1. 需要创建一个APM应用 并设置应用名称以及应用英文名

## 新建应用

×

### 什么是应用?

应用一般是拥有独立的站点，由多个Service共同组成，提供完整的产品功能，拥有独立的软件架构。从技术方面来说应用是Trace数据的存储隔离，在同一个应用内的数据将进行统计和观测。更多请查看产品文档。

应用名称 \*

请输入

英文名 \*

请输入

描述

请输入

确定

取消

2. 选择插件和对应的语言环境，提供相应的帮助文档

3. 选择存储并进行存储设置

## SDK上报数据

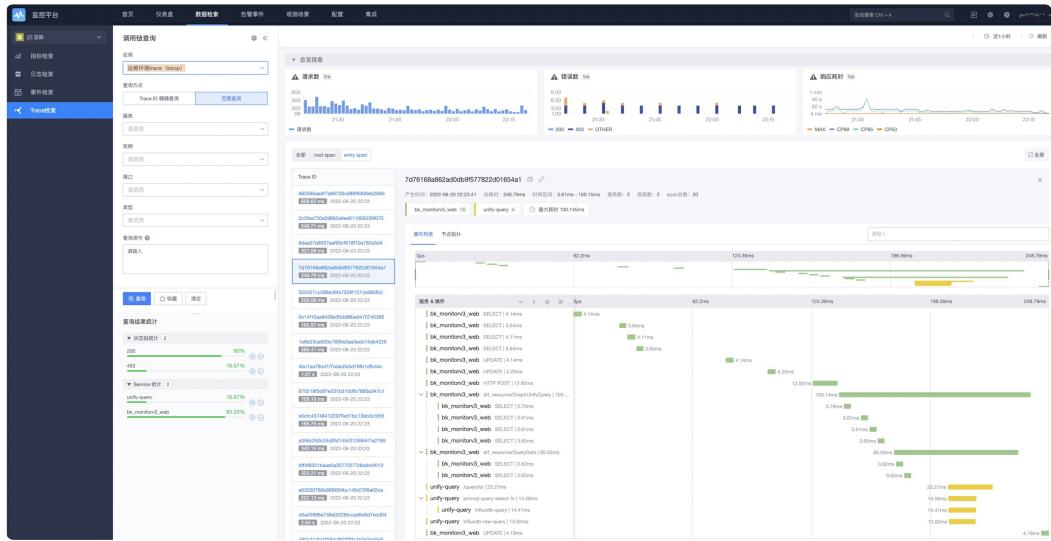
创建完应用后，拿着生成的Token信息，按上报地址，基于不同的语言接入进行相应数据上报。

具体查看各语言的[SDK上报](#)

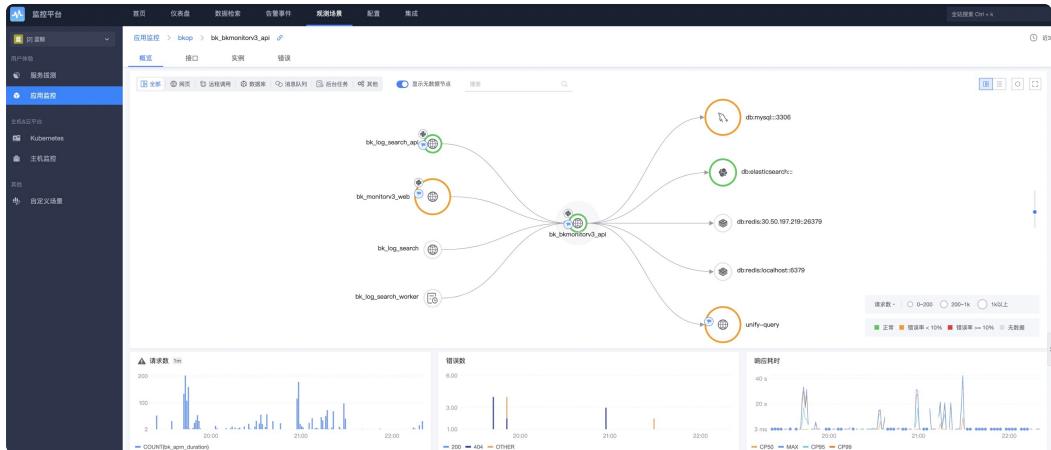
## 查看数据

数据上报成功后，在 数据检索-> Trace检索 最快可以看到数据， 观测场景 -> 应用监控 需要10分钟的统计周期以上才会有相应的数据查看。

## Trace调用链查看



## 应用拓扑查看



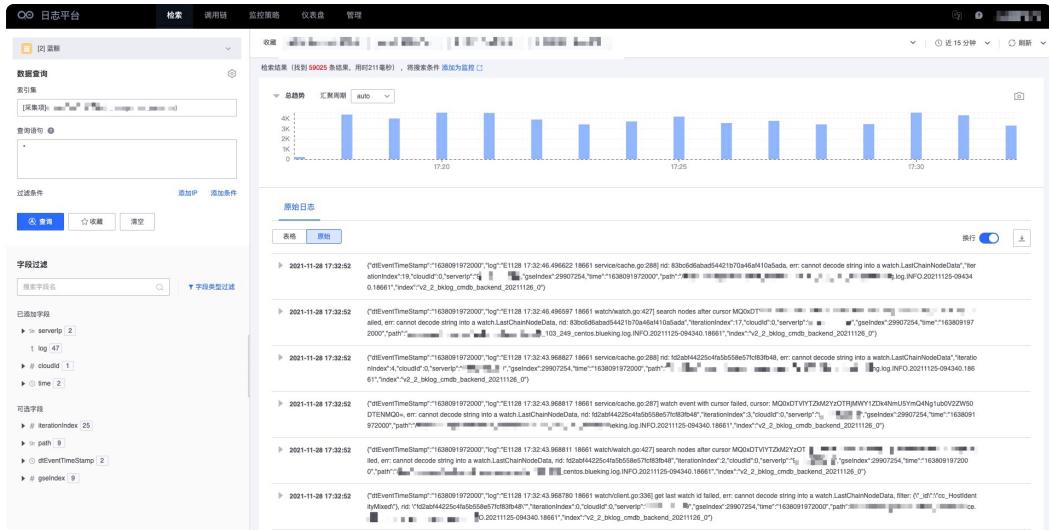
## 日志检索

日志检索主要是用来快速定位问题，避免在服务器端进行日志的查询，优点是性能高效和工具便捷。

### 工作原理

能够进行日志的检索，需要已经有数据源的接入并且形成索引集，才可以用来检索和监控。

### 检索



## 查询语句

点击搜索框，会列出当前日志的字段名

The screenshot shows a search interface with a search bar containing '请输入' (Please input). Below it is a dropdown menu listing various log fields:

- dtEventTimeStamp**: 筛选包含 dtEventTimeStamp 的结果
- log**: 筛选包含 log 的结果
- iterationIndex**: 筛选包含 iterationIndex 的结果
- cloudId**: 筛选包含 cloudId 的结果
- serverIp**: 筛选包含 serverIp 的结果
- gseIndex**: 筛选包含 gseIndex 的结果
- time**: 筛选包含 time 的结果
- path**: 筛选包含 path 的结果

支持 QueryString 语法和正则匹配。具体的查询语法查看[query string 详解](#)。

``` 精确匹配(支持AND、OR) author:"John Smith" AND age:20

字段名匹配(\*代表通配符): status:active title:(quick brown)

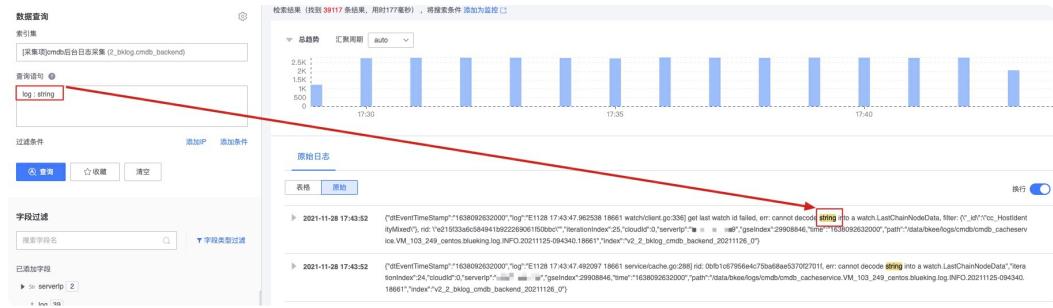
字段名模糊匹配: vers\*on:(quick brown)

通配符匹配: qu?ck bro\*

正则匹配： name:/joh?n(ath[oa]n/

范围匹配(数值类型): count:[1 TO 5] count:[1 TO 5} count:[10 TO \*] ...

点击log字段，输入关键字string，检索结果如下



## 过滤条件

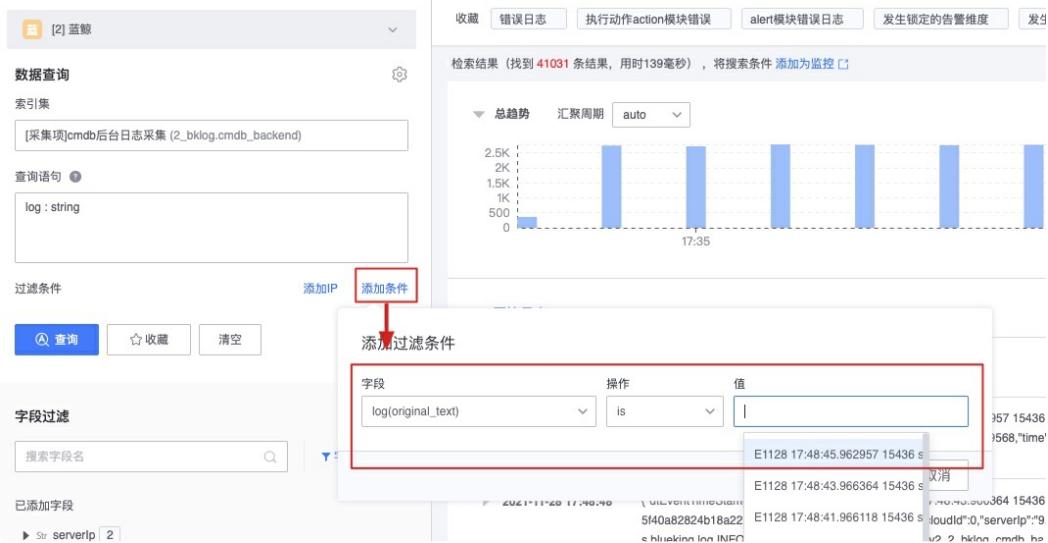
添加过滤条件可以更精确的定位到日志内容。

IP 快选

**IP 快选**通过关联 CMDB 的业务拓扑，控制日志检索范围。



## 添加条件



注意：

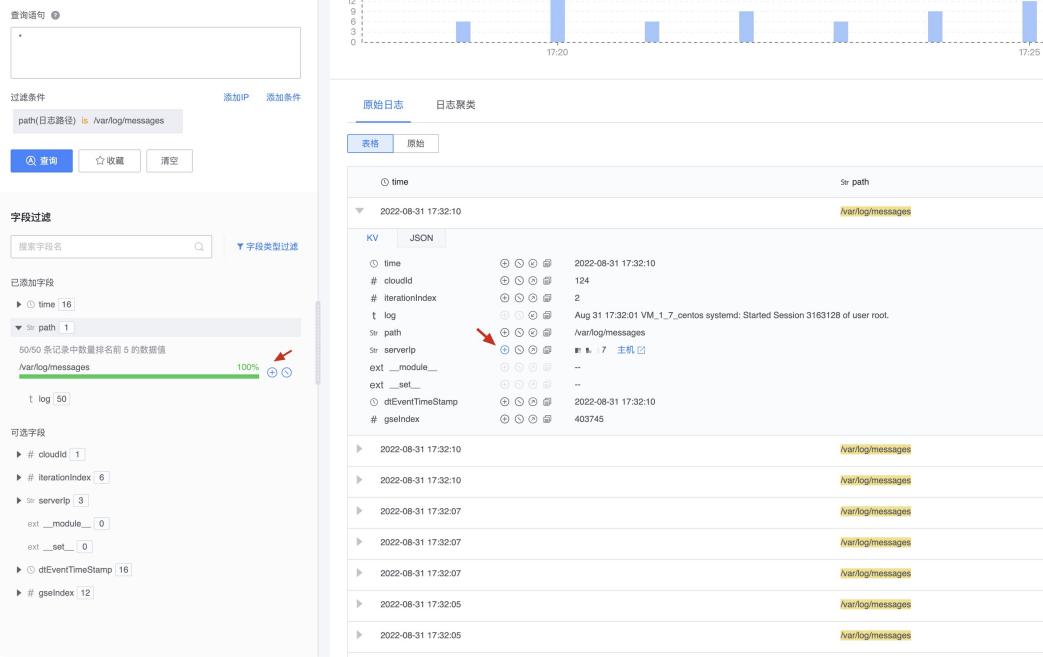
- 值列出的是当前已检索出日志中的值，默认是列出50条日志。如果当前日志中不存在需要的值，则需要手动输入。
- 从添加条件中检索，和输入查询ES的语句其实是一样的效果。

## 操作符

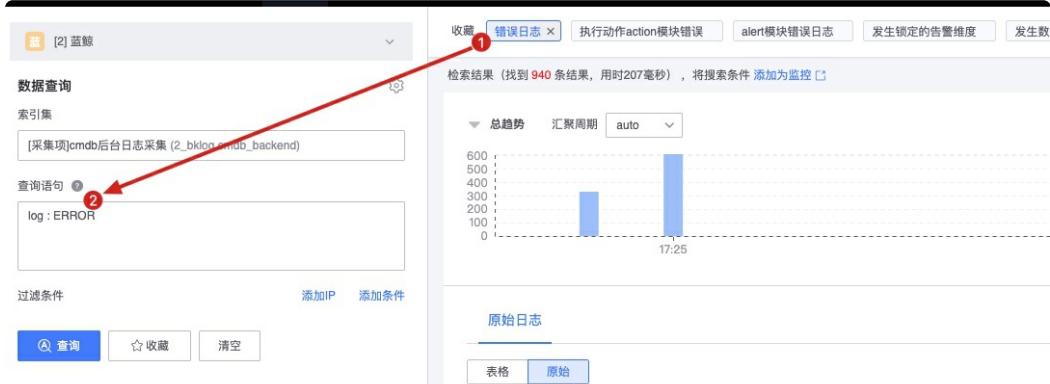
is is one of is not is not one of 大于 小于 等于 大于等于 小于等于 exists does not exists

## 快捷添加条件

- **+** 号 添加 key is value
- **-** 号 添加 key is not value



## 收藏



## 自动查询

数据查询

索引集 [采集项]cmdb后台日志采集 (2\_bklog.cmdb\_backend)

是否开启自动查询

默认自动查询，输入查询语句后，自动会进行检索，但有时习惯上想输入完整的查询语句之后，再点击检索来查询会更明确，就可以关闭自动查询

## 日志格式展示

原始日志

表格 原始 执行  上

2021-11-28 17:48:48 {"dtEventTimeStamp": "1638092928000", "log": "E1128 17:48:45.962957 15436 service/watch.go:38"} id: 69e97a30335e4ddcafe91166d383de, err: cannot decode string into a watch.LastChainNodeData", "iterat ionIndex": 9, "cloudId": 0, "serverIp": "10.10.10.10", "gsIndex": 29909568, "time": "1638092928000", "path": "/data/logs/cmdb/cmdb\_eventserver.VM\_103\_249\_centos.blueking.log.INFO.20211108-203006.15436", "index": "v2\_2\_bklog\_cmdb\_backend\_20211128\_0"}

| KV                 | JSON                                                                                       |
|--------------------|--------------------------------------------------------------------------------------------|
| ○ dtEventTimeStamp | 1638092928000                                                                              |
| t log              | E1128 17:48:45.962957 15436 service/watch.go:38                                            |
| # iterationIndex   | 9                                                                                          |
| # cloudId          | 0                                                                                          |
| Str serverIp       | 10.10.10.10                                                                                |
| # gsIndex          | 29909568                                                                                   |
| ○ time             | 1638092928000                                                                              |
| Str path           | /data/logs/cmdb/cmdb_eventserver.VM_103_249_centos.blueking.log.INFO.20211108-203006.15436 |

内容会联动：

- 主机IP， 监控的IP详情
- traceID， 会联动对应的调用链

原始日志 日志聚类

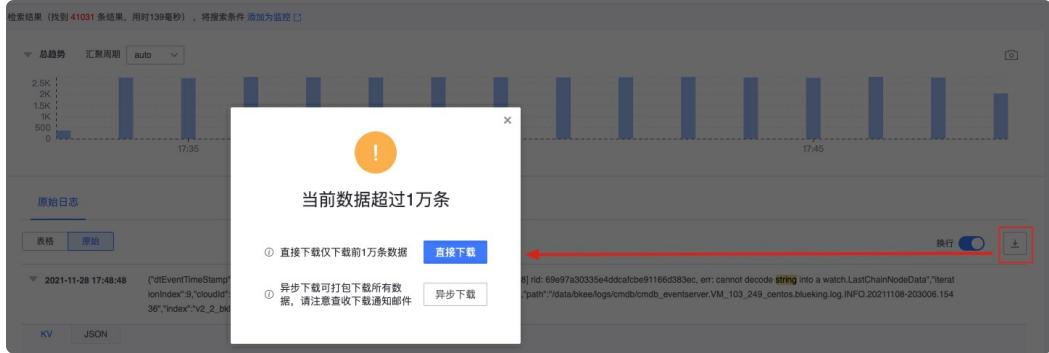
表格 原始 执行  上

2022-08-31 17:41:47 {"\*\*": "\*\*"}  
展开全部

2022-08-31 17:41:47 {"\*\*": "\*\*"}  
展开全部

日志聚类具体查看[./log\_reduce.md]

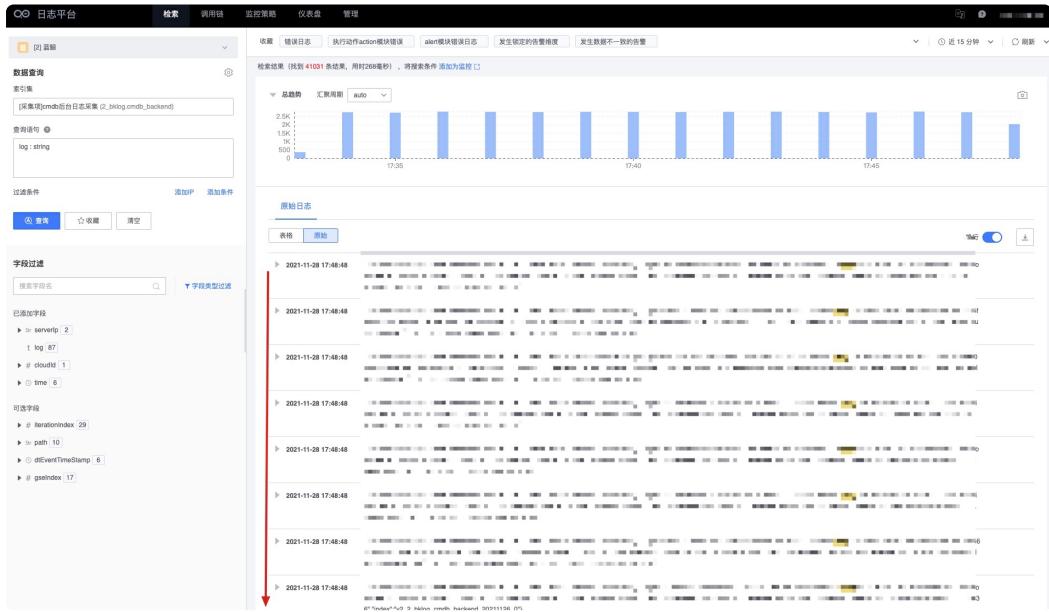
## 日志导出



默认导出是1万条，如需导出更多日志，则可以使用异步下载功能，通过邮件发送下载链接，最大支持100万条日志。

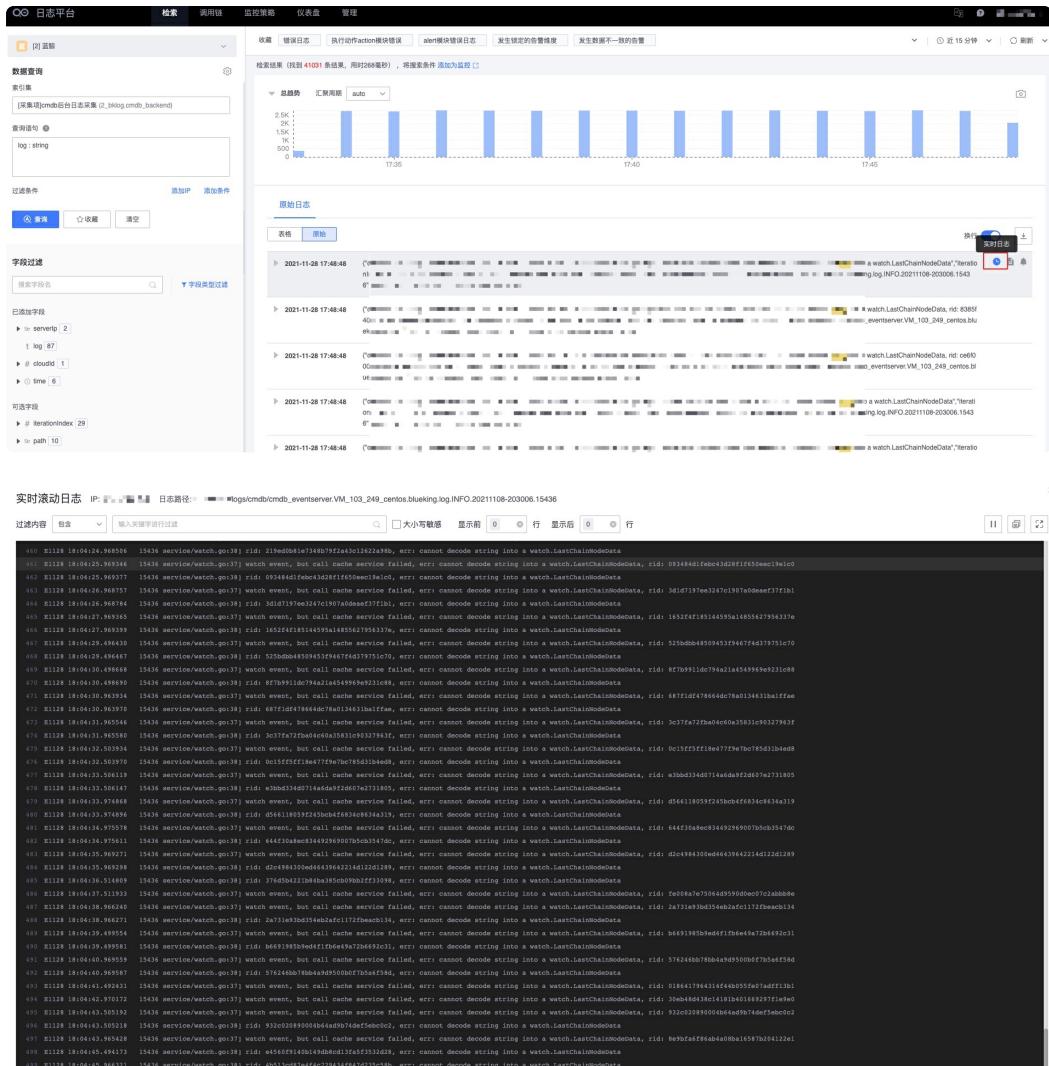
## 加载更多日志

向下滑动，则可以继续加载更多日志



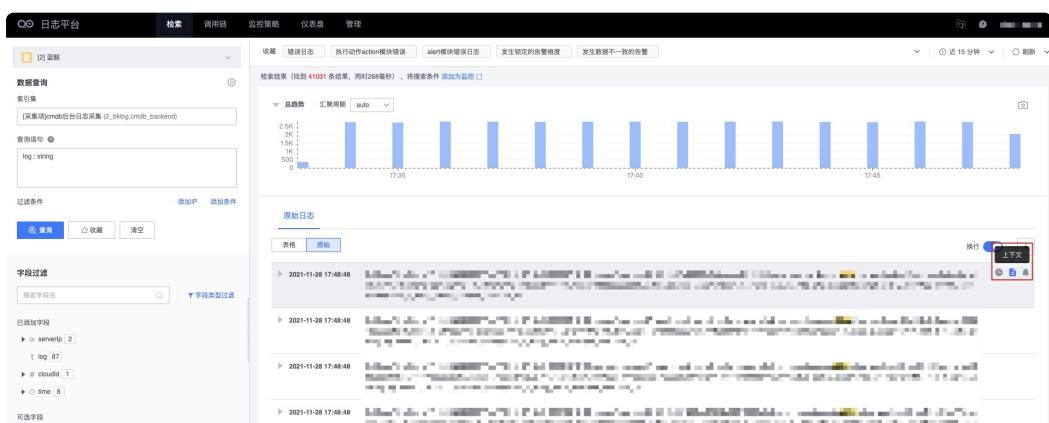
## 实时日志

实时日志滚动。



## 上下文查看

上下文日志，一般是查询到感兴趣的关键词后，需要对产生该日志的上下文进行深究的场景，如下图所示



# 容器 webConsole

如果采集的日志来自BCS容器，会有WebConsole可以直接打开查看

## 字段显示和排序

字段显示和顺序，还有多列排序功能。

| 设置显示字段           |  | 设置排序权重 | 显示别名 |
|------------------|--|--------|------|
| 待选项列表(7)         |  | 全部添加   | 清空   |
| cloudId          |  |        | 删除   |
| iterationIndex   |  |        | 删除   |
| serverIp         |  |        | 删除   |
| __module__       |  |        |      |
| __set__          |  |        |      |
| dtEventTimeStamp |  |        |      |
| gseIndex         |  |        |      |
|                  |  |        |      |
| 已选项列表(3)         |  | i      |      |
| time             |  |        | 删除   |
| path             |  |        | 删除   |
| log              |  |        | 删除   |

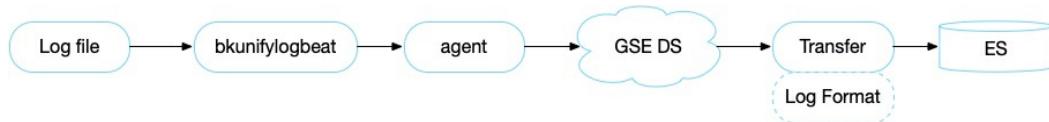
# 如何监控日志平台的数据

日志是发现问题和定位问题非常有用的一类数据，基本上任何监控对象都有日志，操作系统，进程模块，服务模块等会有相应的日志数据。

有效的使用日志数据不仅可以达到监控的目的也可以满足问题定位的需求。

## 前置步骤

工作原理：



日志采集是监听日志文件内容变化进行实时的日志流采集，采集过程中可以进行“日志提取”格式化日志内容。

日志采集和处理基于日志平台服务。可以直接“日志平台”进行采集和配置，也可以直接在监控平台进行。

采集导航路径：日志平台 → 管理 → 日志采集

策略配置导航路径：导航 → 监控配置 → 策略 → 新建 → 日志关键字/监控项(来源日志平台)

问题定位导航路径：导航 → 日志检索

视图配置导航路径：导航 → 仪表盘 → 新建 → 指标选择(来源日志平台)

## 日志采集&字段提取

具体查看日志平台[采集接入](#)

## 日志监控策略

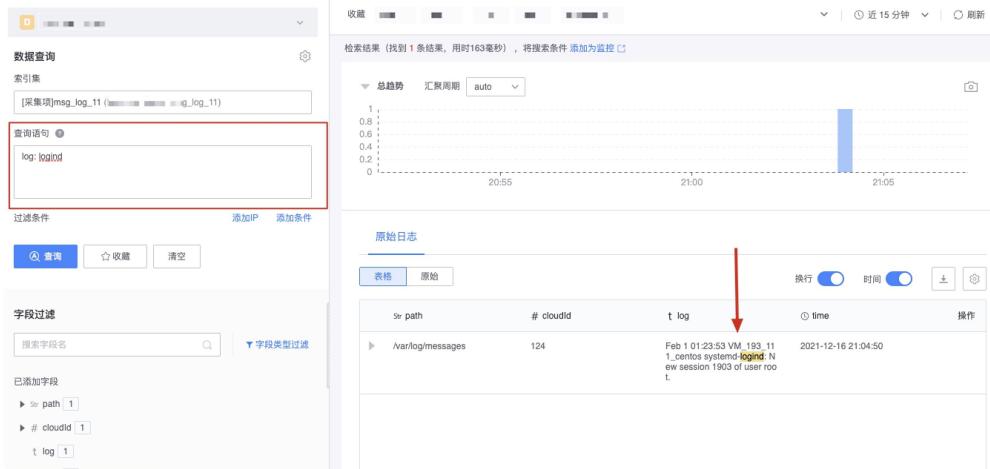
日志监控有两类：

1. 日志关键字监控，最常见也是最常用的监控方法
2. 日志转指标时序数据监控，就是将日志转化为时序数据进行监控

### 日志关键字监控

两个入口

1. 可以直接在日志检索，在检索后的结果点击上方的“添加为监控”



## 2. 在监控平台的策略配置里面直接选择"日志关键字"策略。

The screenshot shows the 'Log Keyword' strategy selected in the monitoring platform's policy configuration interface. Other options like '指标' (Metrics), '事件' (Events), and '关联告警' (Associated Alarms) are also shown.

The screenshot shows the 'Log Keyword' strategy selected in the monitoring platform's policy configuration interface. Other options like '指标' (Metrics), '事件' (Events), and '关联告警' (Associated Alarms) are also shown.

所使用的查询语句都是一样的

“ 注意：当前的日志检索语句是基于 ES 的分词语法。具体查看[query string](#)。 ”

## 日志转指标时序数据监控

经过字段提取的数据会进入到监控的指标选择器中。

- 指标：聚合类型并且是数值类型的，如 int/long。
- 维度：聚合类型并且为 string 类型的。



注意：当不需要进行监控，也不需要进行整体查询的时候可以使用分词。可以节省存储空间，提高效率。



设置&预览

重置 查看标准字段

| 字段名        | 字段说明(选填) | 类型     | 聚合                                  | 分词                                  | 时间                       | 预览(值)                                                    |
|------------|----------|--------|-------------------------------------|-------------------------------------|--------------------------|----------------------------------------------------------|
| request_ip | 请求主机     | string | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> | 10.0.1.10                                                |
| datetime   | 请求时间     | string | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | 25/Jan/2020:19:54:55 +0800                               |
| request    | 请求方式     | string | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | POST                                                     |
| url        | 请求URL    | string | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | /api/v3/hosts/favorites/search                           |
| protocol   | 请求协议     | string | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> | HTTP/1.0                                                 |
| code       | 返回码      | int    | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | 200                                                      |
| sendbytes  | 传输字节     | long   | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | 140                                                      |
| referrer   | 引荐网页     | string | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | http://www.baidu.com/s?wd=...&tn...                      |
| useragent  | 用户端      | string | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_2) Apple... |

在监控的指标选择器中将出现两个指标项 `code` 和 `sendbytes`。

其他都会变成维度

## 数据可视化

数据除了在策略中可以使用之外也可以在仪表盘、数据检索中查询到。

# 快速构建 Nginx 集群

## 情景

传统的 Nginx 集群要先部署多个 Nginx 节点，然后通过 `upstream` 统一一个入口提供给用户访问。

该过程操作繁琐，接下来看 BCS（容器管理平台）如何通过 容器调度 快速构建 Nginx 集群。

## 前提条件

- K8S 基本概念，包含 Deployment、Services。
- 完成 BCS 部署
- 准备 2 台云主机：4 核 8 G，不低于 CentOS 7，K8s Master 和 Node 各 1 台
- 完成上述 2 台云主机的 Agent 安装，并分配至 CMDB 业务下

## 操作步骤

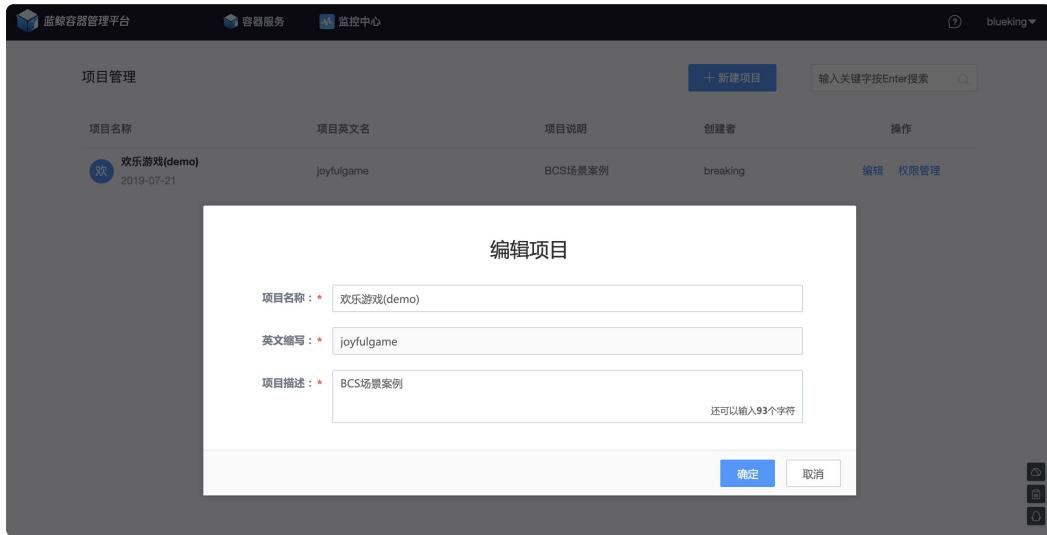
### 1. 新建集群

## 2. BCS 快速构建 Nginx 集群

### 新建集群

#### 启用容器服务

在 BCS 首页，点击 新建项目，如 欢乐游戏(demo)。



然后选择容器编排类型为 Kubernetes，关联 前提条件 中提到的 CMDB 业务，点击 启用容器服务。



### 新建集群

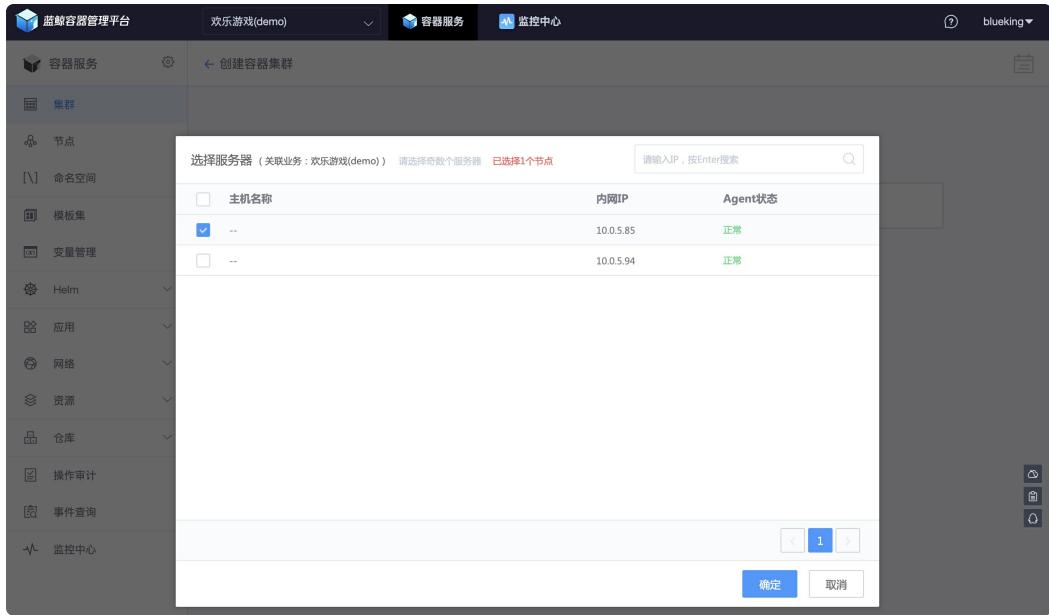
启用容器服务后，进入容器服务欢迎页，点击 创建容器集群。

Copyright © 2012–2019 Tencent BlueKing. All Rights Reserved

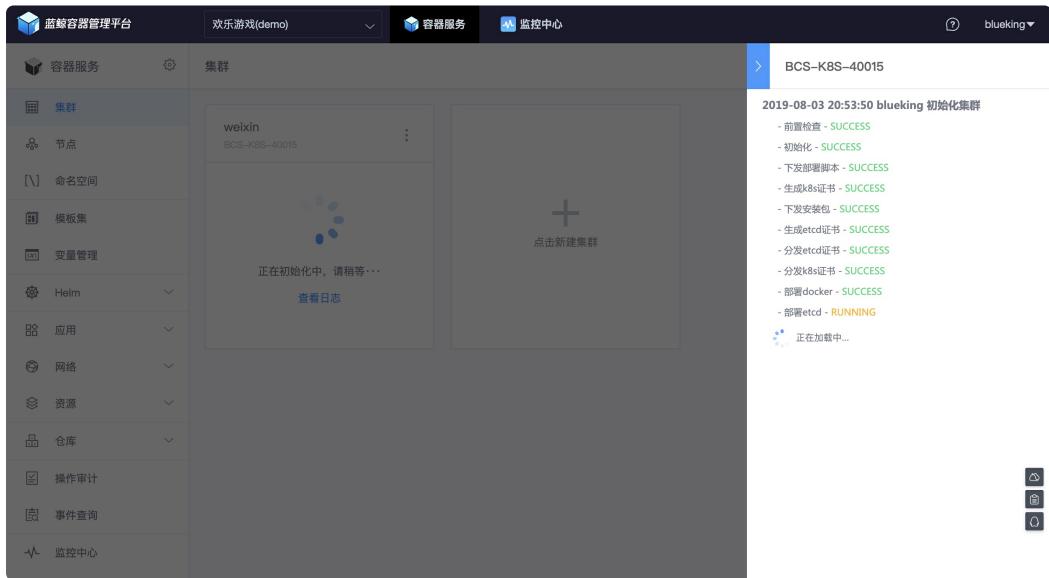
按提示填写集群的基本信息。

容器服务的集群划分和 **传统单体应用在 CMDB 中的集群划分**很类似，可以按照 **地域（如华北区）** 或者 **完全独立的应用集合（微信区）** 来划分。

选择 1 台云主机作为 Master。



点击 **确定** 后，集群开始初始化。



点击 **节点管理**

The screenshot shows the BlueKing Container Management Platform interface. At the top, there are tabs for '容器服务' (Container Service), '欢乐游戏(demo)' (selected), '容器服务', and '监控中心'. On the left, a sidebar menu includes '集群' (Cluster), '节点' (Nodes), '命名空间' (Namespace), '模板集' (Template Set), '变量管理' (Variable Management), 'Helm', '应用' (Application), '网络', '资源', '仓库', '操作审计', '事件查询', and '监控中心'. The main area displays a cluster named 'weixin' (BCS-K8S-40015). A context menu is open over the cluster details, with '节点管理' (Node Management) highlighted by a red box. Other options in the menu include '总览' (Overview), '集群信息' (Cluster Information), and '删除' (Delete). Below the cluster details, there is a message '点击新建集群' (Click to create a new cluster) with a plus sign icon. On the right side of the main area, there are three small icons: a circular arrow, a refresh symbol, and a circular arrow with a minus sign.

点击 **添加节点**，按提示节点添加。

This screenshot shows the '节点管理' (Node Management) page for the 'weixin' cluster. The top navigation bar includes '容器服务', '欢乐游戏(demo)', '容器服务', '监控中心', and user information 'blueking'. The left sidebar is identical to the previous screenshot. The main content area shows the 'weixin' cluster details again. Below the cluster details, there are tabs for '总览' (Overview), '节点管理' (Node Management) (which is selected and highlighted by a red box), and '集群信息' (Cluster Information). A large '添加节点' (Add Node) button is prominently displayed with a red box around it. Below this button, there is a search bar and a '批量操作' (Batch Operation) dropdown. The main table lists one node: '10.0.5.94' which is '初始化中' (Initializing). There is a '查看日志' (View Log) link next to it. At the bottom, there is a pagination control showing '共计1条 每页 5 条' (Total 1 item per page 5 items) and a blue '1' button indicating the current page. On the right side of the main area, there are three small icons: a circular arrow, a refresh symbol, and a circular arrow with a minus sign.

至此，新建集群完毕。可以看到集群的基础信息。

The screenshot shows the BlueKing Container Management Platform interface. At the top, there are tabs for '容器服务' (Container Service) and '监控中心' (Monitoring Center). The left sidebar is titled '容器服务' and includes sections for '集群' (Cluster), '节点' (Nodes), '命名空间' (Namespace), '模板集' (Template Set), '变量管理' (Variable Management), 'Helm', '应用' (Application), '网络' (Network), '资源' (Resources), '仓库' (Repository), '操作审计' (Audit Log), '事件查询' (Event Query), and '监控中心' (Monitoring Center). The main content area displays a cluster named 'weixin' (BCS-K8S-40015). It shows resource usage metrics: CPU 使用率 (CPU Usage Rate) at 29.13%, 内存使用率 (Memory Usage Rate) at 14.76%, and 磁盘使用率 (Disk Usage Rate) at 8.53%. A large button labeled '点击新建集群' (Click to Create New Cluster) is present. On the right side, there are three small icons.

另外，在集群的设置(:)下拉菜单中，可以看到集群主要性能指标。

This screenshot shows the detailed performance metrics for the 'weixin' cluster. The left sidebar is identical to the previous screenshot. The main area has tabs for '概览' (Overview), '节点管理' (Node Management), and '集群信息' (Cluster Information). The '概览' tab is selected, displaying three charts: 'CPU 使用率' (CPU Usage Rate) at 29.13% (1.17 of 4.00 Shares), '内存使用率' (Memory Usage Rate) at 14.75% (1.10GB of 7.46 GB), and '磁盘使用率' (Disk Usage Rate) at 8.53% (26.43GB of 309.81 GB). Below the charts, there are two circular progress indicators: one for '节点' (Nodes) at 100% and one for '命名空间' (Namespaces) at 0. A 'WebConsole' button is located at the bottom right.

## BCS 快速构建 Nginx 集群

### 新建命名空间

新建命名空间 `dev`。

## 新建模板集

模板集，可以类比为 K8S 中 Helm 的 Charts，在 K8S 编排中，是 K8S 对象的集合：`Deployment`（无状态）、`StatefulSet`（有状态）、`DaemonSet`（守护进程集）、`Job`（定时任务）、`Configmap`（配置项）`Secret`（保密字典），具体参见 [模板集使用介绍](#)。

打开菜单 [模板集]，新建模板集 `web-nginx`。

按提示，填写 `Deployment`



## 填写 Service

**关联应用:** web-nginx **1. 关联Deployment**

**Service类型:** NodePort **2. 访问Service的方式。NodePort会占用Node的端口资源以提供外部访问。**  
如果想通过LoadBalancer将服务提供外部访问，需要二次开发做集成

**端口映射:** http 8088 TCP **3. Service暴露的端口** **4. Deployment中定义容器监听的端口** **5. Node上监听的端口**

**标签管理:** app = web-nginx

## 实例化

## 检查部署效果

在菜单 **网络** -> **Services** 中，找到刚实例化的 Service **web-nginx**。

在菜单 **[应用]** -> **[Deployment]** 中可以找到 **web-nginx**。

以及其运行指标：

通过访问 `Node+NodePort`，可以查看刚刚部署 Nginx 集群的版本号。

```
bash [root@ip-10-0-5-94-n-bcs-k8s-40015 ~]# curl 10.0.5.94:30008 -I HTTP/1.1 200 OK
Server: nginx/1.12.2 Date: Thu, 08 Aug 2019 09:11:42 GMT
```

通过访问 `Service IP + Port`，也可以查看刚部署 Nginx 的版本号。

```
bash [root@ip-10-0-5-94-n-bcs-k8s-40015 ~]# curl 10.254.11.4:8088 -I HTTP/1.1 200 OK
Server: nginx/1.12.2 Date: Thu, 08 Aug 2019 09:12:33 GMT Content-Type: text/html
Content-Length: 612 Last-Modified: Tue, 11 Jul 2017 13:29:18 GMT Connection: keep-
alive ETag: "5964d2ae-264" Accept-Ranges: bytes
```

## 在 K8S 中部署 WordPress

### 情景

WordPress 是流行的开源博客程序，Helm 官方维护了 WorePress 的 Chart，接下来看在 BCS 中如何部署 WordPress，开始你的博客之旅。

### 前提条件

- 了解 Helm 的使用方法
- 集成 K8S 存储，例如 将 NFS 作为 K8S PV Provisioner
- Git Clone Helm Charts
- 新增 LoadBalancer

### 操作步骤

1. 上传 WordPress Chart 到仓库
2. 部署 WordPress
3. 访问测试

## 上传 WordPress Chart 到仓库

进入 `Charts` 本地仓库的 `wordpress` 目录。

```
```bash
```

## cd charts/stable/wordpress/

||

```
总用量 84 -rw-r--r-- 1 root root 454 9月 12 11:01 Chart.yaml -rw-r--r-- 1 root root 186 9月 12 11:01
OWNERS -rw-r--r-- 1 root root 29059 9月 12 11:01 README.md -rw-r--r-- 1 root root 233 9月 12 11:01
requirements.lock -rw-r--r-- 1 root root 173 8月 27 17:00 requirements.yaml drwxr-xr-x 3 root root 4096 9月
12 11:01 templates -rw-r--r-- 1 root root 13278 9月 12 11:01 values-production.yaml -rw-r--r-- 1 root root
12925 9月 12 11:01 values.yaml ````
```

可以看到存在 `requirements.yaml` 文件，了解 WordPress 依赖 mariadb 的 Chart。

```
```bash
```

## cat requirements.yaml

```
dependencies: - name: mariadb version: 6.x.x repository: https://kubernetes-
charts.storage.googleapis.com/ condition: mariadb.enabled tags: - wordpress-database ````
```

在当前目录使用 Helm 命令下载依赖包。

```
```bash
```

## helm dep build

```
Hang tight while we grab the latest from your chart repositories... ...Successfully got an update from the
"joyfulgame" chart repository ...Successfully got an update from the "stable" chart repository Update
Complete. □ Happy Helming! □ Saving 1 charts Downloading mariadb from repo https://kubernetes-
charts.storage.googleapis.com/ Deleting outdated charts ````
```

推送 Chart 到仓库

```
```bash
```

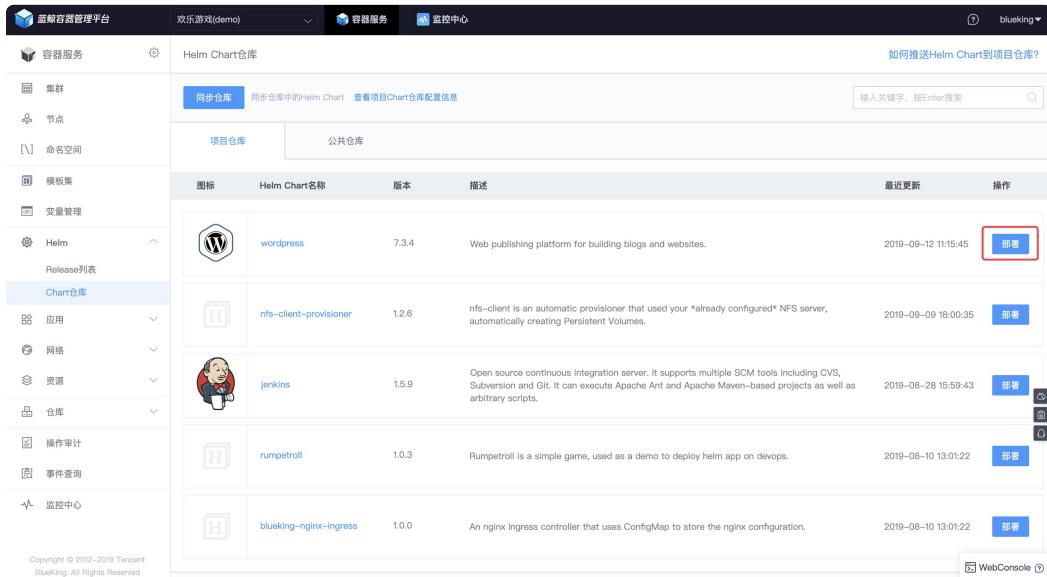
# helm push . joyfulgame

Pushing wordpress-7.3.4.tgz to joyfulgame... Done. ````

在 BCS 【Chart 仓库】菜单中，点击【同步仓库】，将刚刚上传的 Chart 从仓库同步到 BCS 的界面中。

## 部署 WordPress

在 BCS 【Chart 仓库】菜单中，找到刚刚上传的 WordPress Chart，点击【部署】。



图标	Helm Chart名称	版本	描述	最近更新	操作
	wordpress	7.3.4	Web publishing platform for building blogs and websites.	2019-09-12 11:15:45	<span style="border: 2px solid red; padding: 2px;">部署</span>
	nfs-client-provisioner	1.2.6	nfs-client is an automatic provisioner that uses your *already configured* NFS server, automatically creating Persistent Volumes.	2019-09-09 18:00:35	<span style="border: 2px solid blue; padding: 2px;">部署</span>
	jenkins	1.5.9	Open source continuous integration server. It supports multiple SCM tools including CVS, Subversion and Git. It can execute Apache Ant and Apache Maven-based projects as well as arbitrary scripts.	2019-08-28 15:59:43	<span style="border: 2px solid blue; padding: 2px;">部署</span>
	rumpetroll	1.0.3	Rumpetroll is a simple game, used as a demo to deploy helm app on devops.	2019-08-10 13:01:22	<span style="border: 2px solid blue; padding: 2px;">部署</span>
	blueking-ingress	1.0.0	An nginx Ingress controller that uses ConfigMap to store the nginx configuration.	2019-08-10 13:01:22	<span style="border: 2px solid blue; padding: 2px;">部署</span>

可以修改 Helm 参数，例如 WordPress 管理员账号、密码等，这里重点提一下用户访问用到的 Service 和 Ingress：

- Service

使用 ClusterIP 即可，因为用户访问使用 Ingress。

```
yaml service: type: ClusterIP # HTTP Port port: 80
```

- Ingress

此处启用 Ingress，并填写绑定的 主机名 和 路径。类似 Nginx 配置文件中 Server Section 部分的 server\_name。

The screenshot shows the Helm parameters configuration for the WordPress chart. The 'YAML模式' tab is selected. The code block displays the rendered Kubernetes YAML, which includes annotations for Ingress, secrets for MySQL, and a StatefulSet for MySQL. A specific line of code is highlighted: `- name: wordpress.bk.tencent.com`, indicating the custom domain mapping.

```

    250 ## 
251- ingress:
252   ## Set to true to enable ingress record generation
253   enabled: true
254
255   ## Set this to true in order to add the corresponding annotations for cert-manager
256   certManager: false
257
258   ## Ingress annotations done as key:value pairs
259   ## For a full list of possible ingress annotations, please see
260   ## ref: https://github.com/kubernetes/ingress-nginx/blob/master/docs/annotations.md
261
262   ## If tls is set to true, annotation kubernetes.io/secure-backends: "true" will automatically be set
263   ## If certManager is set to true, annotation kubernetes.io/tls-acme: "true" will automatically be set
264   annotations:
265     # kubernetes.io/ingress.class: nginx
266
267   ## The list of hostnames to be covered with this ingress record.
268   ## Most likely this will be just one host, but in the event more hosts are needed, this is an array
269   hosts:
270     - name: wordpress.bk.tencent.com
271       port: 80
272
273   ## This is configuration for the ingress
  
```

点击【预览】，可以看到 BCS 将 Charts 通过 `helm template` 命令渲染为 K8S 的对象描述文件。

The screenshot shows the preview of the rendered Kubernetes YAML configuration. The code block displays the detailed object definitions, including the StatefulSet for the MySQL database, the Deployment for the WordPress application, and the Service for external traffic routing. The MySQL StatefulSet includes labels and annotations for BCS integration.

```

1  apiVersion: apps/v1beta1
2  kind: StatefulSet
3  metadata:
4    annotations:
5      io.tencent.paa.creator: blueking
6      io.tencent.paa.updator: blueking
7    labels:
8      app: mariadb
9      chart: mariadb-6.8.7
10     component: master
11     heritage: tiller
12     io.tencent.bcs.app.appid: '6'
13     io.tencent.bcs.controller: BCS-K8S-40015
14     io.tencent.bcs.containerLabel: wordpress-1909121112-mariadb
15     io.tencent.bcs.controller.type: StatefulSet
16     io.tencent.bcs.kind: Kubernetes
17     io.tencent.bcs.monitor.level: general
18     io.tencent.bcs.namespace: dev
19     io.tencent.bkdata.baseall.dataid: '6566'
20     io.tencent.bkdata.container.stateful.dataid: '0'
21     io.tencent.bkdata.container.stateful.id: '106168461940c2952b97790b967'
22     io.tencent.paa.source_type: helm
23     io.tencent.paa.version: 7.3.4
24     io.tencent.bcs.cluster: BCS-K8S-40015
25     release: wordpress-1909121112
26     name: wordpress-1909121112-mariadb
27   spec:
28     replicas: 1
29     selector:
30       matchLabels:
31         app: mariadb
32         component: master
33       release: wordpress-1909121112
34     serviceName: wordpress-1909121112-mariadb
35     template:
36       metadata:
37         labels:
38           app: mariadb
39           chart: mariadb-6.8.7
40           component: master
41           io.tencent.bcs.app.appid: '6'
  
```

点击【部署】即可。

在【Release 列表】菜单中可以查看部署状态。

名称	类型	Pods
wordpress-1909130255	Deployment	1/1
wordpress-1909130255-mariadb	StatefulSet	1/1
wordpress-1909130255-mariadb	Service	-/-
wordpress-1909130255	Service	-/-
wordpress-1909130255	Ingress	-/-
wordpress-1909130255-mariadb	ConfigMap	-/-
wordpress-1909130255-mariadb-tests	ConfigMap	-/-
wordpress-1909130255-mariadb	Secret	-/-
wordpress-1909130255	Secret	-/-
wordpress-1909130255	PersistentVolumeClaim	-/-
wordpress-1909130255-mariadb-test-sqtpg	Pod	-/-
wordpress-1909130255-credentials-test	Pod	-/-

部署成功，接下来测试访问。

## 访问测试

修改域名解析或 PC 上 hosts 文件（Mac 下路径为 /etc/hosts），将 Ingress 中配置的主机名指向到 LoadBalancer 中节点的外网 IP，然后打开浏览器访问，可以看到 WordPress 首页。

User's Blog! — Just another WordPress site

## Hello world!

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

user September 13, 2019 Uncategorized 1 Comment

Search

### Recent Posts

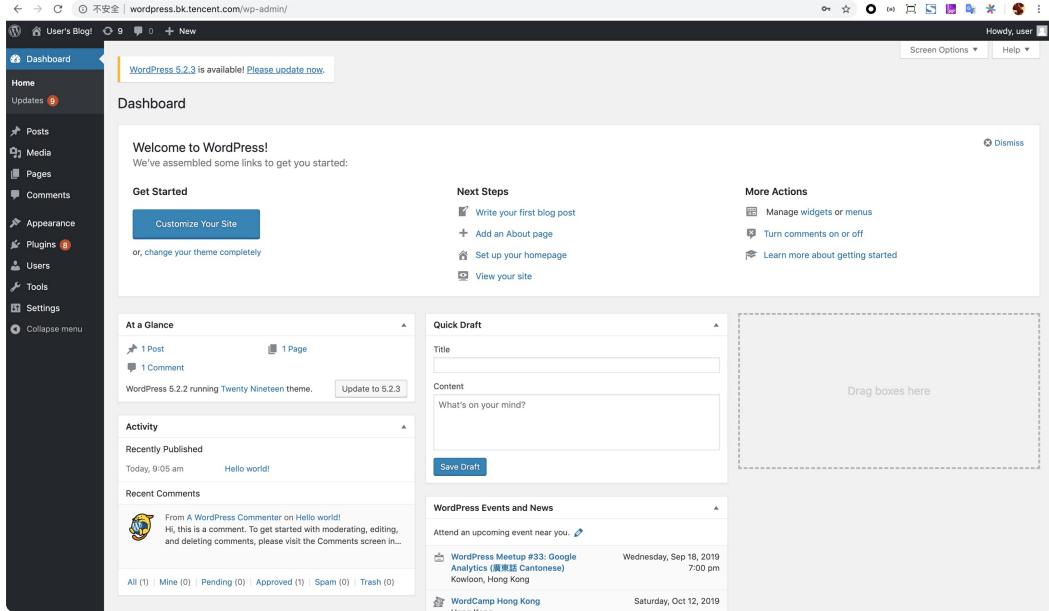
[Hello world!](#)

---

### Recent Comments

### Archives

输入用户名（默认为 user）和密码登录 Wordpress 后台。



如果没有在 Chart 参数中没有设置密码，可以通过命令获取 WordPress 的 Secret。

```
```yaml
```

## kubectl get secret -n dev

```
NAME TYPE DATA AGE wordpress-1909130255 Opaque 1 4h37m
```

## kubectl get secret/wordpress-1909130255 -n dev -o yaml

```
apiVersion: v1
data:
  wordpress-password: bFgzTmZvSHJIVg==
kind: Secret
creationTimestamp: 2019-09-13T07:01:33Z
name: wordpress-1909130255
namespace: dev
type: Opaque
```

```
echo "bFgzTmZvSHJIVg==" | base64 --decode
```

```
IX3NfoHrHV ````
```

BCS 部署应用，如此简单。

## Java Maven Demo

本篇文章将指导你如何在 bk-ci 编译Maven工程。

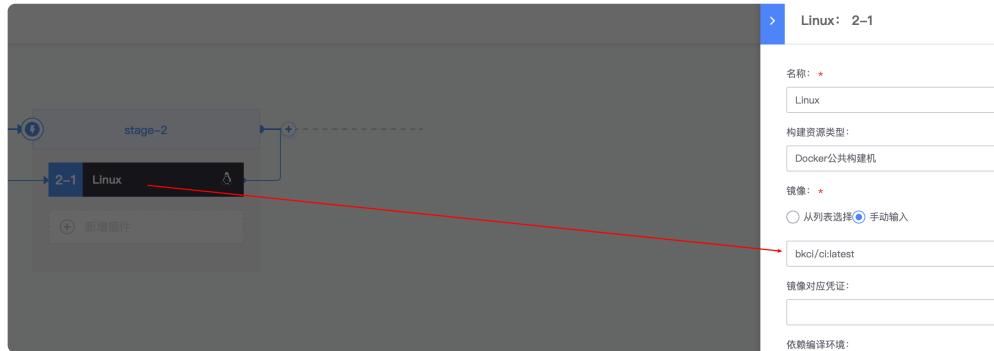
### 准备材料

- 一个 maven 工程: <https://gitlab.com/bk-ci/gs-maven.git>

- 一个包含 mvn 命令的 CI 镜像: <https://hub.docker.com/r/bkci/ci>

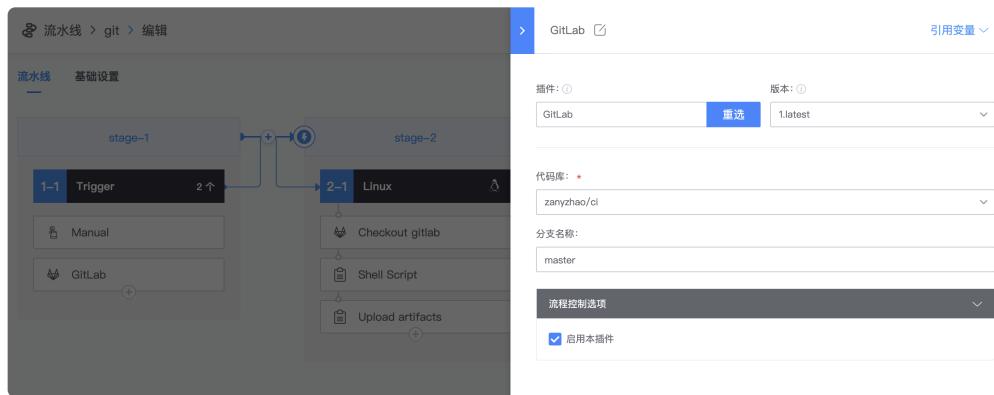
## 详细步骤

- 将准备好的 gitlab 代码库关联至 bk-ci, 请参考
- 创建一条空白流水线
- 将 Linux 构建环境添加到 Job2-1, 镜像地址填写: bkci/ci:latest



- 依次添加如下 4 个插件:

### 5. Checkout Gitlab



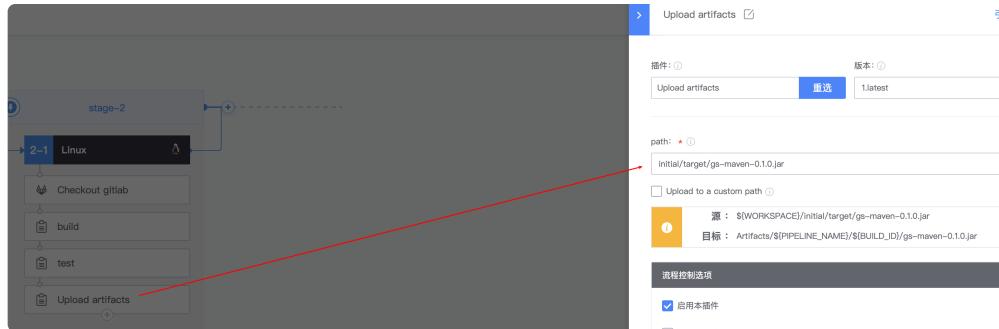
### 6. Shell Scripts

```
bash #!/usr/bin/env bash cd initial mvn install
```

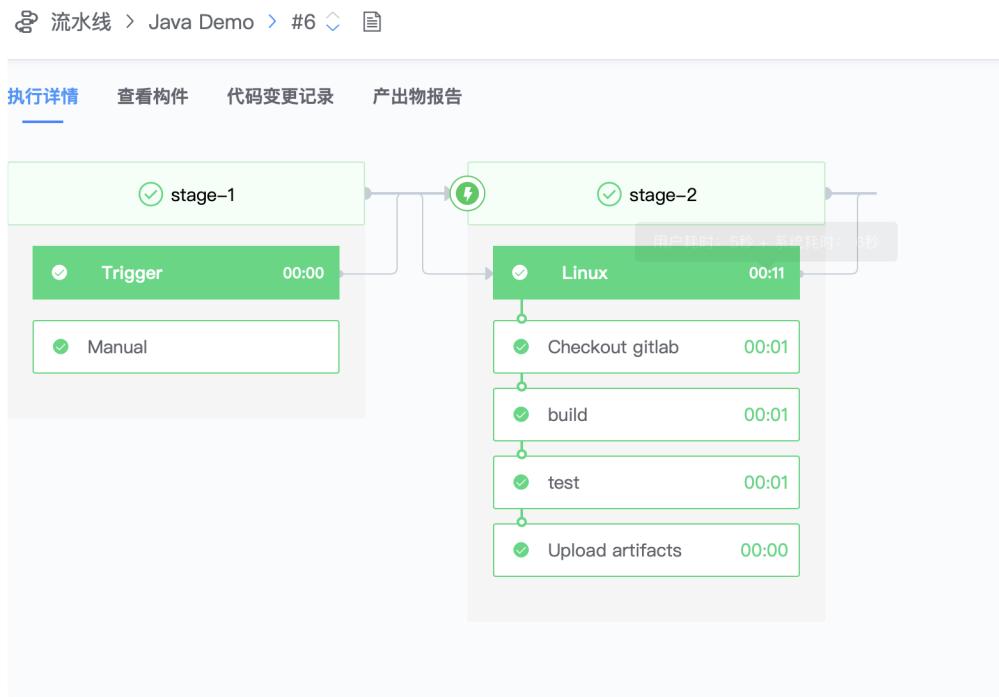
### 7. Shell Scripts

```
bash #!/usr/bin/env bash cd initial mvn test
```

### 8. Upload artifacts



## 9. 运行流水线，观察结果



# NodeJS Demo

本篇文章将指导你如何在 bk-ci 编译**NodeJS**工程。

## 准备材料

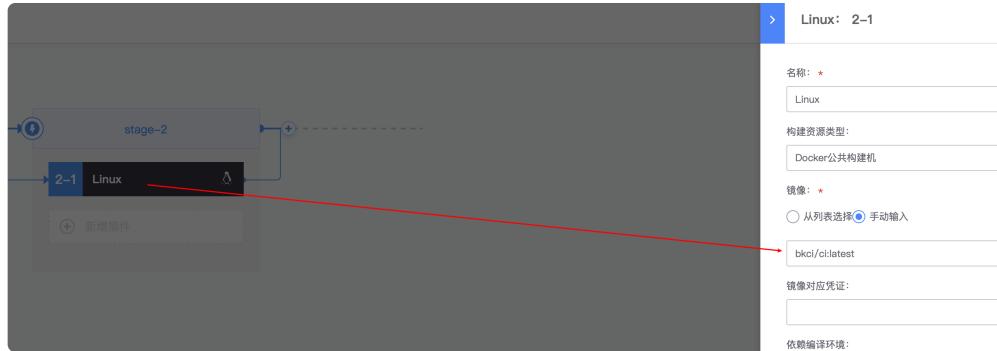
- 一个 NodeJS 工程: <https://gitlab.com/bk-ci/gs-maven.git>
- 一个包含 npm 命令的 CI 镜像: <https://hub.docker.com/r/bkci/ci>

## 详细步骤

1. 将准备好的 gitlab 代码库关联至 bk-ci, [请参考](#)

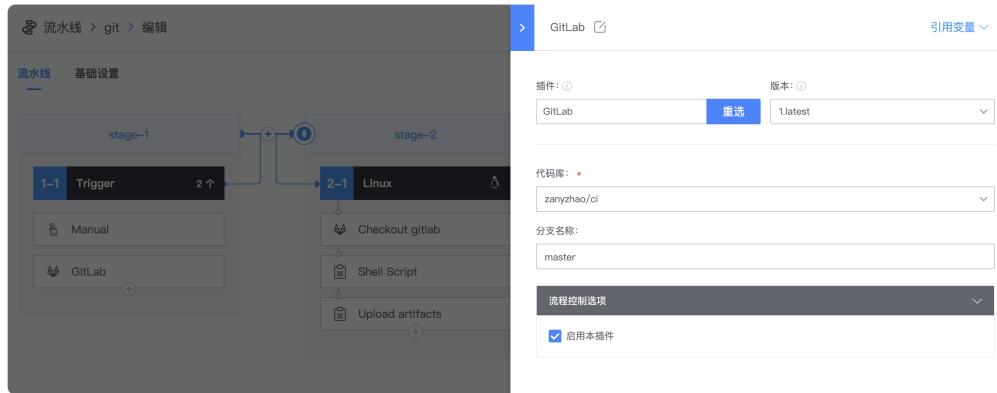
2. 创建一条空白流水线

3. 将 Linux 构建环境添加到 Job2-1，镜像地址填写：bkci/ci:latest



4. 依次添加如下 3 个插件：

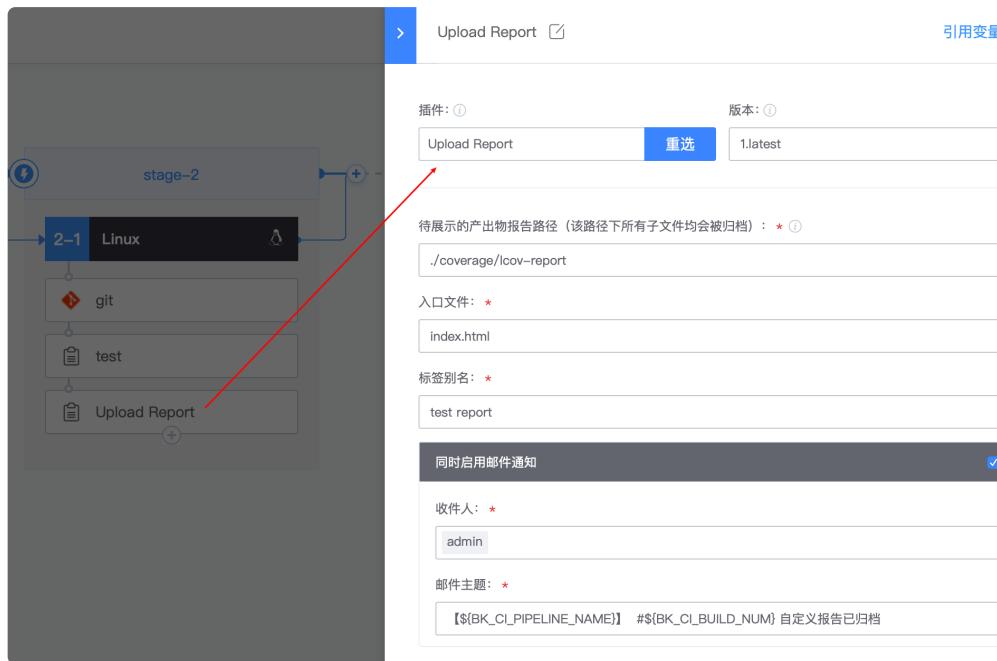
5. Checkout Gitlab



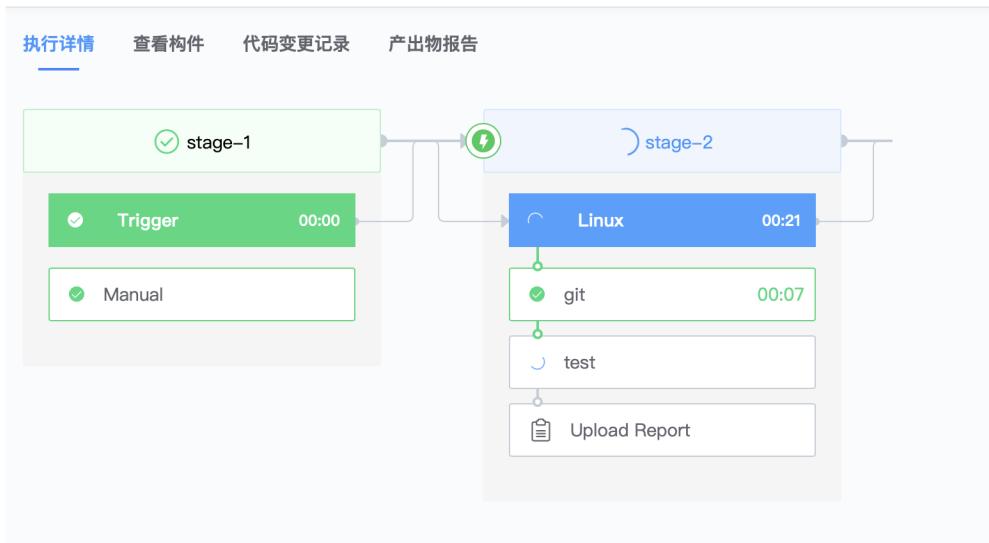
6. Shell Scripts

```
bash #!/usr/bin/env bash npm run test
```

7. Upload artifacts



8. 运行流水线，观察结果



## 测试环境自动更新

### 情景

开发同学完成功能开发、本地测试通过后，运维通过自动化工具或人工登录服务器更新，由于开发和运维存在 **沟通成本**，导致持续集成的测试环节效率低下，影响整体研发效率。

接下来看蓝盾（BK-CI）是如何做到 **测试环境自动更新**。

### 前提条件

- 部署蓝盾（BK-CI）

### 操作步骤

- 配置自动构建流水线
- 提交代码（Git Commit）验证

#### 配置自动构建流水线



## 提交代码 (Git Commit) 验证

下面为一次自动更新记录，从 提交代码到更新测试环境，不到 1 分钟。

```
{% video %}assets/bk-ci-demo-HD.mp4{% endvideo %}
```