

# Table of Contents

- 1. 功能简介.....4
  - 1.1 快速入门.....12
  - 1.2 更新日志索引.....33
    - 1.2.1 V1.15.0 更新日志.....42
    - 1.2.2 V1.14.0 更新日志.....47
    - 1.2.3 V1.13.0 更新日志.....50
    - 1.2.4 V1.12.0 更新日志.....53
    - 1.2.5 V1.11.0 更新日志.....58
    - 1.2.6 V1.10.0 更新日志.....62
    - 1.2.7 V1.9.0 更新日志.....65
    - 1.2.8 V1.8.0 更新日志.....70
    - 1.2.9 V1.7.0 更新日志.....75
    - 1.2.10 V1.6.0 更新日志.....78
    - 1.2.11 V1.5.9 更新日志.....82
    - 1.2.12 V1.5.7 更新日志.....87
    - 1.2.13 V1.5.6 更新日志.....92
  - 1.3 使用咨询&联系我们.....96
    - 1.3.1 帆软服务平台简介.....98
    - 1.3.2 合作认证操作说明.....102
    - 1.3.3 管理员管理方式.....107
    - 1.3.4 企业成员操作方式.....109
- 2. 运维平台场景案例.....111
  - 2.1 场景案例-应用宕机处理与分析.....116
  - 2.2 场景案例-应用配置修改与启停.....121
  - 2.3 场景案例-应用监控与告警.....125
  - 2.4 场景案例-应用资源分配.....132
  - 2.5 最佳实践-备份还原.....140
  - 2.6 最佳实践-应用更新.....149
- 3. 部署升级 FineOps.....152
  - 3.1 部署运维平台.....153
  - 3.2 登录运维平台.....166
  - 3.3 升级运维平台.....171
    - 3.3.1 外网升级运维平台.....172
    - 3.3.2 内网升级运维平台.....180
  - 3.4 常见问题.....187
    - 3.4.1 部署运维平台问题排查.....188
    - 3.4.2 端口占用检查与防火墙配置.....192
    - 3.4.3 docker 常用操作指南.....196
- 4. 部署迁移升级项目.....205
  - 4.1 部署新项目.....206
    - 4.1.1 部署新 FineReport 单机项目.....211
    - 4.1.2 部署新 FineReport 集群项目.....227
    - 4.1.3 部署新 FineBI 单机项目.....245
    - 4.1.4 部署新 FineBI 集群项目.....261
    - 4.1.5 部署新 FineDataLink 单机项目.....279
    - 4.1.6 部署新 FineDataLink 集群项目.....280
  - 4.2 接入已有项目.....297
    - 4.2.1 接入运维平台已部署的项目.....298
    - 4.2.2 接入非运维平台部署的项目.....304
  - 4.3 注册新项目.....317
  - 4.4 更新项目.....322
    - 4.4.1 外网升级运维项目.....323

4.4.2 内网升级运维项目 .....	330
4.5 迁移项目 .....	337
4.5.1 非容器化单机迁移容器化单机 .....	338
4.5.2 非容器化单机迁移容器化集群 .....	356
4.5.3 非容器化集群迁移容器化集群（仅迁移工程） .....	375
4.5.4 非容器化集群迁移容器化集群（迁移工程和集群组件） .....	390
4.5.5 容器化单机迁移容器化单机 .....	406
4.5.6 容器化单机迁移容器化集群 .....	415
4.5.7 容器化集群迁移容器化集群 .....	430
4.6 集群运维项目扩容 .....	440
4.7 常见问题 .....	441
4.7.1 运维平台项目异常问题排查 .....	442
4.7.2 MinIO 文件服务器简介 .....	446
5. 运维管理 .....	447
5.1 项目管理 .....	449
5.1.1 组件管理 .....	455
5.1.2 组件配置变更 .....	462
5.1.3 添加组件 .....	468
5.1.4 为运维项目开启 SSL .....	472
5.2 项目总览 .....	477
5.3 文件管理 .....	478
5.4 备份管理 .....	483
5.5 镜像管理 .....	492
5.6 运维组件 .....	498
5.7 为运维平台开启 SSL .....	506
6. 监控与告警 .....	510
6.1 监控 .....	512
6.1.1 系统资源监控 .....	513
6.1.2 系统健康观测 .....	520
6.1.3 服务器监控 .....	532
6.1.4 应用监控 .....	533
6.1.5 流量监控 .....	534
6.1.6 运维平台监控 .....	540
6.1.7 组件监控 .....	544
6.2 告警 .....	557
6.2.1 告警任务 .....	558
6.2.2 告警记录 .....	570
6.2.3 告警方式 .....	571
7. 业务工具 .....	591
7.1 健康巡检 .....	593
7.2 健康巡检内容 .....	599
7.3 宕机自动处理 .....	644
7.4 宕机处理策略 .....	648
7.5 资源清理 .....	653
7.6 会话管理 .....	661
7.7 资源优先级管理 .....	664
7.8 调度管理 .....	669
7.9 业务配置更改 .....	674
8. 日志管理 .....	701
8.1 运维日志 .....	703
8.2 性能堆栈 .....	709
8.3 dump 文件 .....	722
9. 实验性功能 .....	725
9.1 链路追踪 .....	726
9.1.1 场景案例-健康观测 .....	727

- 9.1.2 首屏链路 ..... 728
  - 9.1.3 调用链路 ..... 729
  - 9.1.4 非容器化部署项目接入链路追踪 ..... 740
- 9.2 资源隔离管理..... 745
  - 9.2.1 业务组管理配置 Nginx 组件 ..... 751

# 1. 功能简介

## 1. 概述

企业在管理帆软应用（FineReport、FineBI）的过程中，常常会遇到一些痛点和挑战：

- 1) **低效的运维流程：**应用管理员运维经验不足，专业运维人员又无法深入解决应用问题，传统运维流程耗时耗力，导致企业运维效率低下，影响业务的正常运转。
- 2) **不足的监控手段：**运维人员可能无法准确监控项目的性能和可用性，缺乏运维过程中必要的工具和技术。
- 3) **多项目管理混乱：**企业可能有多个项目运行，而这些项目由不同的部门或团队负责管理，导致版本不统一、部署环境各异，难以进行统一的运维管理。

因此帆软推出「FineOps 运维平台」，帮助企业更好地解决以上问题。

## 2. 功能定位

FineOps 运维平台是帆软软件推出的，为企业提供帆软应用稳定运行和全方位运维管理的解决方案。

导航	说明
<a href="#">官网指引</a>	用户可前往官网了解 FineOps 运维平台更多应用场景和相关资料
<a href="#">下载安装</a>	用户可前往下载页，获取 FineOps 运维平台安装包，在本地进行安装试用 安装步骤请参见： <a href="#">部署运维平台</a>
<a href="#">Demo 展示</a>	用户可前往 Demo 平台，使用帆软通行证登录，查看 FineOps 运维平台功能 出于安全考虑，该 Demo 演示系统屏蔽了部分功能，如需体验运维平台全功能，建议下载安装试用。 如出现使用问题，付费用户请联系技术支持
<a href="#">问题反馈</a>	技术支持联系方式：前往「 <a href="#">服务</a> 」，选择「在线支持」、电话「400-811-8890」 [helpvideo]5778[/helpvideo]

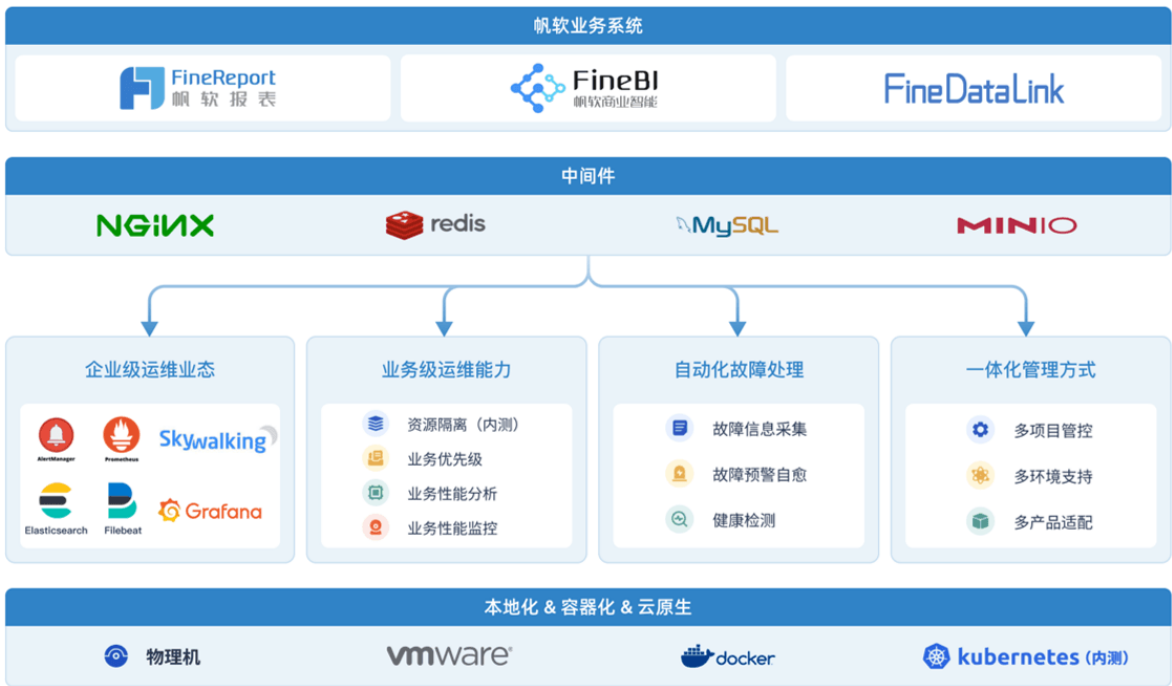
## 3. 功能架构

运维平台是基于多种开源工具和组件构建的监控和管理系统，包括 prometheus、

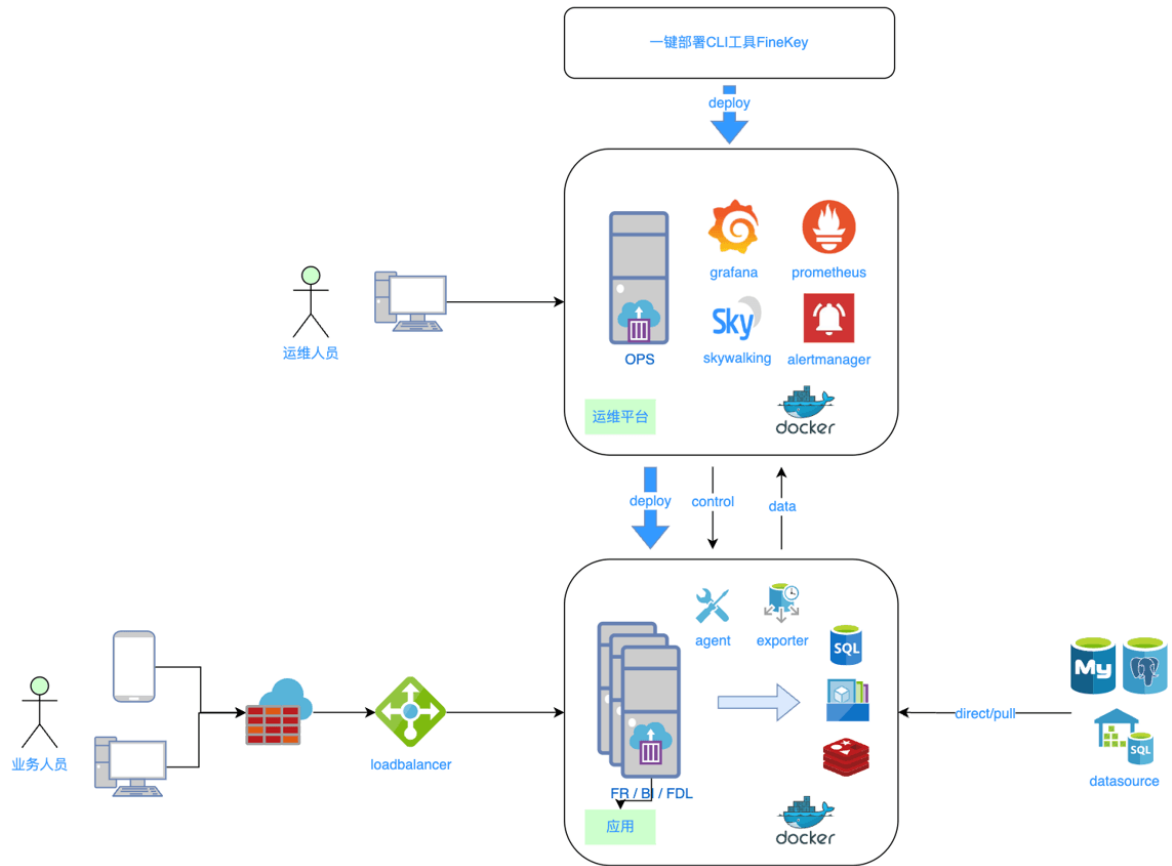


skywalking、grafana 等，共同协作帮助运维人员轻松地监控、管理和优化帆软应用。

运维矩阵：



技术架构：



4. 核心功能

核心功能	应用场景
项目管理	一个运维平台能够同时对接、部署、监控多个帆软项目。管理运维更统一、更高效。
系统监控	多视角、多维度、全环境的运维监控，让管理员纵观全局。

告警管理	高实时精准告警，比用户反馈更早一步通知系统异常。 企微、钉钉快捷对接，自带通知合并机制，避免消息轰炸。
资源优先级	重要业务报表、VIP 用户请求先行。适应于有额外服务器资源，需要打造VIP 环境的企业。
链路追踪	请求级别记录、还原用户报表访问的性能情况，逐层展开链路信息，快速定位性能问题。
备份管理	异地备份，适应于有灾备诉求的用户，可将工程相关文件异地备份到运维平台所在机器上。

5. 场景优势

- 部署运维平台：[详情请点击该链接](#)
- 容器化部署 FR/BI/FDL：[详情请点击该链接](#)

注：下表列出的所有场景，均面向在 Linux 系统上部署的帆软应用。

帆软推荐用户使用 Linux 系统服务器部署正式工程，原因请参见：[Linux 和 Windows 对比分析](#)

场景	非容器化部署	运维平台+非容器化部署	运维平台+容器化部署	功能导航
服务器资源监控	✗	✓	✓	支持监控 Linux 服务器磁盘、网络、C 盘、网络、CPU、内存等资源状态
				<a href="#">系统资源监控</a>
				<a href="#">系统健康观测</a>
				<a href="#">服务器监控</a>
应用状态监控		✗		不支持监控 Windows 服务器
				<a href="#">应用监控</a>
				<a href="#">流量监控</a>
	✓	✓	✓	<a href="#">运维平台监控</a>
	1. 内存	1. 内存	1. 内存	<a href="#">组件监控</a>
	2. CPU	2. CPU	2. CPU	
	3. 磁盘	3. 磁盘	3. 磁盘	
		4. 负载	4. 负载	

		4. 负载	5. 线程池	5. 线程池
		5. 监控时间短,	6. gc 状态	6. gc 状态
		不支持回溯	7. BI 资源监控	7. BI 资源监控
			8. 支持回溯历史时	8. 支持回溯历史时间资源状态
		间资源状态		
		✘	✔	✔
			集群组件	集群组件
			1. 配置数据库监	1. 配置数据库监
			控: MySQL	控: MySQL
			2. 状态服务器监	2. 状态服务器监
			控: Redis	控: Redis
			3. 负载均衡监控:	3. 负载均衡监控:
			Nginx	Nginx
			4. 文件服务监控:	4. 文件服务监控:
			MinIO	MinIO
		✔	✔	✔
		1. 负载异常预	1. 负载异常预警	1. 负载异常预警
		警	2. 磁盘空间预警	2. 磁盘空间预警
		2. 磁盘空间预	3. 宕机提醒预警	3. 宕机提醒预警
		警	4. 集群组件状态异	4. 集群组件状态异
			常预警	常预警
			5. 线程阻塞预警	5. 线程阻塞预警
			6. 打开文件数过多	6. 打开文件数过多
			预警	预警
			7. 自定义预警	7. 自定义预警
		✘	✘	✔
		需要上服务器后	需要上服务器后台	运维平台前端界面
		台操作	操作	化操作
				启停应用及集群组
告警	异常告警			<a href="#">告警任务</a>
				<a href="#">告警记录</a>
				<a href="#">告警方式</a>
日常运维	应用启停			<a href="#">组件管理</a>
				<a href="#">运维组件</a>

配置更改	✕	✕	✓	件
	需要上服务器后台操作	需要上服务器后台操作	运维平台前端界面	<a href="#">组件管理</a>
			化操作	<a href="#">业务配置更改</a>
			1. 限制应用及集群	<a href="#">调度管理</a>
文件管理			组件资源	<a href="#">会话管理</a>
			2. 修改组件配置	<a href="#">资源清理</a>
				<a href="#">调度管理</a>
	✕	✓	✓	
更新升级	需要上服务器后台操作	运维平台前端界面	运维平台前端界面	
		化操作	化操作	<a href="#">文件管理</a>
		查看、上传、下载文件	查看、上传、下载文件	
		件	件	
（小版本）	✓	✓	✓	
	1. 服务器后台操作	1. 服务器后台操作	运维平台远程一键	<a href="#">组件管理</a>
	换 JAR	换 JAR	更新升级	<a href="#">运维组件</a>
	2. 数据决策系统在线更新升级	2. 数据决策系统在线更新升级		<a href="#">镜像管理</a>
异地备份还原	✕	✓	✓	
	需要自行手动实现异地备份还原	1. 备份至运维平台所在机器	1. 备份至运维平台所在机器	
		2. 不支持集群组件备份	2. 支持运维平台部署的集群组件整体备份，自己部署的集群组件需要自行处理	<a href="#">备份管理</a>
		3. 需手动还原	3. 支持集群整体还原，自己部署的集	

	群组件需要自行处理		
	✓	✓	✓
	巡检插件	巡检	1. 巡检
			2. 容器化标准环境
环境巡检			(包括环境要求变化时容器化更新自动覆盖)
日志获取	✓	✓	✓
	前端获取	1. 前端获取	1. 前端获取
		2. 前端触发生成堆栈等日志	2. 前端触发生成堆栈等日志
webshell 命令行执行	✗	✗	✓
	需要上服务器后台操作	需要上服务器后台操作	支持前端 webshell 至容器内执行命令
故障处理	✓	✓	✓
	1. 支持宕机自动恢复	支持 Linux 服务器中工程：	1. 支持宕机自动恢复
	2. 进程消失自动恢复覆盖范围优先	1. 宕机自动恢复	2. 支持进程消失自动恢复
		2. 进程消失自动恢复覆盖范围优先	
自动恢复		✗	
		不支持通过运维平台对 Windows 服务器中工程实现宕机自动恢复	
		但可依赖工程数据决策系统的宕机自	

[健康巡检](#)

[日志下载](#)

[组件管理](#)

[宕机自动处](#)

[理](#)

[宕机处理策](#)

[略](#)

		动恢复功能	
故障分析	×	✓	✓
	需要获取日志文	支持自动分析堆栈及 gc 日志	支持自动分析堆栈及 gc 日志
	件人工分析		<a href="#">日志分析</a>
部署	×	×	✓
	应用部署		1. 支持直接界面部署容器化应用 FR、BI, 配置端口
			2. 支持部署 Redis、Nginx、MinIO 等集群服务组件
其他	×	✓	✓
	资源优先级	支持定义高优先级资源, 资源紧张时	支持定义高优先级资源, 资源紧张时
		优先保障	优先保障
对接多项目	×	✓	✓
		支持对接多个项目	支持对接多个项目
	×	✓	✓
资源隔离		1. 支持实现集群内不同节点服务于不同用户	1. 支持实现集群内不同节点服务于不同用户
		2. 支持实现集群内不同节点服务于不同业务模板	2. 支持实现集群内不同节点服务于不同业务模板
	×	✓	✓
实验性功能		1. 支持通过链路追踪监控应用访问性能	1. 支持通过链路追踪监控应用访问性能
		2. 支持查询并分析链路	2. 支持查询并分析链路
			<a href="#">系统健康观测</a>
链路追踪			<a href="#">首屏链路</a>
			<a href="#">调用链路</a>



# 1.1 快速入门

## 1. 概述

本文以最简洁清晰的步骤拆解，带大家了解部署运维平台和帆软应用的流程，以及用户可使用运维平台进行的运维操作。

[helpvideo]6564[/helpvideo]

## 2. 准备服务器

容器化部署需要基于运维平台前端进行，用户需要先部署 1 个 FineOps 运维平台，再使用运维平台部署帆软应用。

因此用户需要准备**多台服务器**：

- 单节点工程准备 **2 台**服务器：1 台用于部署 FineOps，1 台用于部署帆软应用
- 多节点集群准备「**工程节点数+2 台**」服务器：1 台用于部署 FineOps，1 台用于部署集群组件，其他用于部署帆软应用工程节点（资源充足时，集群组件服务器可准备更多台）

本文介绍的是最简单的部署方案，因此对服务器配置要求较高，端口无法严格限制。

如无法提供全新服务器、网络端口限制较严格，请参考：[部署运维平台](#)、[部署新项目](#)

工程	服务器	推荐配置
单节点工程	运维平台服务器	1) 准备 <b>1 台</b> 全新服务器，该服务器不部署其他工程或内容，所有端口均未被占用
		2) 系统版本：CentOS 7.3~7.9 版本 / Ubuntu 18.04.4 及以上版本 / redhat 7.6 及以上版本
		3) 系统内核：3.10 及以上，最低 4 核
		4) 系统架构：X86_64
		5) 剩余空闲磁盘：使用命令「df -h」查询，/root/data 目录所在分区剩余空闲磁盘大于 200 G
		6) 剩余空闲物理内存：16G 以上，最低 12G
		7) 网络要求：对外开放 80 端口，确保用户可通过 80 端口访问运



维平台

8) 用户：使用 root 用户完成部署

1) 准备 1 台全新服务器，服务器不部署其他工程或内容，所有端口均未被占用

2) 系统版本：CentOS 7.3~7.9 版本 / Ubuntu 18.04.4 及以上版本 / redhat 7.6 及以上版本

3) 系统内核：3.10 及以上，fr/bi 最低 8 核，fdl 最低 4 核

4) 系统架构：X86\_64

5) 剩余空闲磁盘：使用命令「df -h」查询，服务器/root/data 目录所在分区剩余空闲磁盘大于：fr/fdl 50 G，bi 100G

工程节点服务器 6) 剩余空闲物理内存：fr 16G 以上、bi 32G 以上、fdl 8G 以上

7) 网络要求：

- 所有服务器与运维平台所在服务器内网互通
- 对外开放 8080 端口，确保用户可通过 8080 端口访问帆软应用

8) 用户：使用 root 用户完成部署

FDL 额外：

联系帆软技术支持，获取 FineDataLink、Kafka 相关镜像包

1) 准备 1 台全新服务器，该服务器不部署其他工程或内容，所有端口均未被占用

2) 系统版本：CentOS 7.3~7.9 版本 / Ubuntu 18.04.4 及以上版本 / redhat 7.6 及以上版本

3) 系统内核：3.10 及以上，最低 4 核

4) 系统架构：X86\_64

5) 剩余空闲磁盘：使用命令「df -h」查询，/root/data 目录所在分区剩余空闲磁盘大于 200 G

多节点运维平台  
集群 服务器

- 6) 剩余空闲物理内存：16G 以上，最低 12G
- 7) 网络要求：对外开放 80 端口，确保用户可通过 80 端口访问运维平台
- 8) 用户：使用 root 用户完成部署
- 1) 按照「**工程节点数**」准备全新服务器数量，服务器不部署其他工程或内容，所有端口均未被占用
- 2) 系统版本：CentOS 7.3~7.9 版本 / Ubuntu 18.04.4 及以上版本 / redhat 7.6 及以上版本
- 3) 系统内核：3.10 及以上，frbi 最低 8 核，fdl 最低 4 核
- 4) 系统架构：X86\_64
- 5) 剩余空闲磁盘：使用命令「df -h」查询，每台服务器/root/data 目录所在分区剩余空闲磁盘大于：

工程节点  
服务器

- fr/fdl 50 G，bi 100G

- 6) 剩余空闲物理内存：  
  
fr 16G 以上、bi 32G 以上、fdl 8G 以上
- 7) 网络要求：所有服务器与运维平台所在服务器、集群组件所在服务器内网互通
- 8) 用户：使用 root 用户完成部署

FDL 额外：

联系帆软技术支持，获取 FineDataLink、Kafka 相关镜像包

- 1) 准备 1 台全新服务器，服务器不部署其他工程或内容，所有端口均未被占用

集群组件  
服务器

- 2) 系统版本：CentOS 7.3~7.9 版本 / Ubuntu 18.04.4 及以上版本 / redhat 7.6 及以上版本
- 3) 系统内核：3.10 及以上，最低 4 核
- 4) 系统架构：X86\_64

5) 剩余空闲磁盘：使用命令「df -h」查询：

- 每台服务器/root/data 目录所在分区剩余空闲磁盘大于 50 G
- fdl 如需使用管道功能，集群组件服务器剩余空闲磁盘建议大于 1T

6) 剩余空闲物理内存：8G 以上

7) 网络要求：

- 服务器与运维平台所在服务器、工程节点服务器内网互通
- 对外开放 80 端口，确保用户可通过 nginx80 端口访问帆软应用

8) 用户：使用 root 用户完成部署

3. 部署运维平台

面向对象：部署 FineOps 运维平台。

本文介绍的是最简单的部署方案，因此对服务器配置要求较高，端口无法严格限制。

如无法提供全新服务器、网络端口限制较严格，请参考：[部署运维平台](#)

步骤	说明
	1) 准备一台全新服务器，该服务器不部署其他工程或内容，所有端口均未被占用、均可使用
	2) 系统版本：CentOS 7.3~7.9 版本 / Ubuntu 18.04.4 及以上版本/ redhat 7.6 及以上版本
1 准备服务器	3) 系统内核：3.10 及以上，最低 4 核
	4) 系统架构：X86_64
	5) 剩余空闲磁盘：使用命令「df -h」查询，/root/data 目录所在分区剩余空闲磁盘大于 200 G
	6) 剩余空闲物理内存：16G 以上，最低 12G

7) 网络要求：对外开放 80 端口，确保用户可通过 80 端口访问运维平台

8) 用户：使用 root 用户完成部署

- 内网：

1) 点击下载安装包：[FineOps 运维平台下载](#) 中的「全量版离线安装包」

外网：在线版安装包

2) 将安装包上传到服务器中

3) 终端连接服务器，进入安装包所在目录，解压安装包：tar zxvf finekey-operation-all.tar.gz

- 外网：

准备  
2 安装包

1) 点击下载安装包：[FineOps 运维平台下载](#) 中的「在线版安装包」

2) 将安装包上传到服务器中

3) 终端连接服务器，进入安装包所在目录，解压安装包：tar -xvf finekey-operation-online.tar

```
[root@localhost home]# cd /home
[root@localhost home]# ls
finekey-operation.tar.gz
[root@localhost home]# cd /home
[root@localhost home]# tar zxvf finekey-operation.tar.gz
finekey/
finekey/bin/
finekey/bin/finekey
finekey/bin/fine_net_test
finekey/conf/
finekey/conf/finekey.yaml
finekey/conf/finekey.push.yaml
finekey/resources/
finekey/resources/images/
```

1) 终端连接服务器，进入解压后的 finekey/bin 目录

2) 执行自动化部署：./finekey

3) 出现 deployment success! bye~ 提示时，表示已经成功部署

部署  
3 运维平台

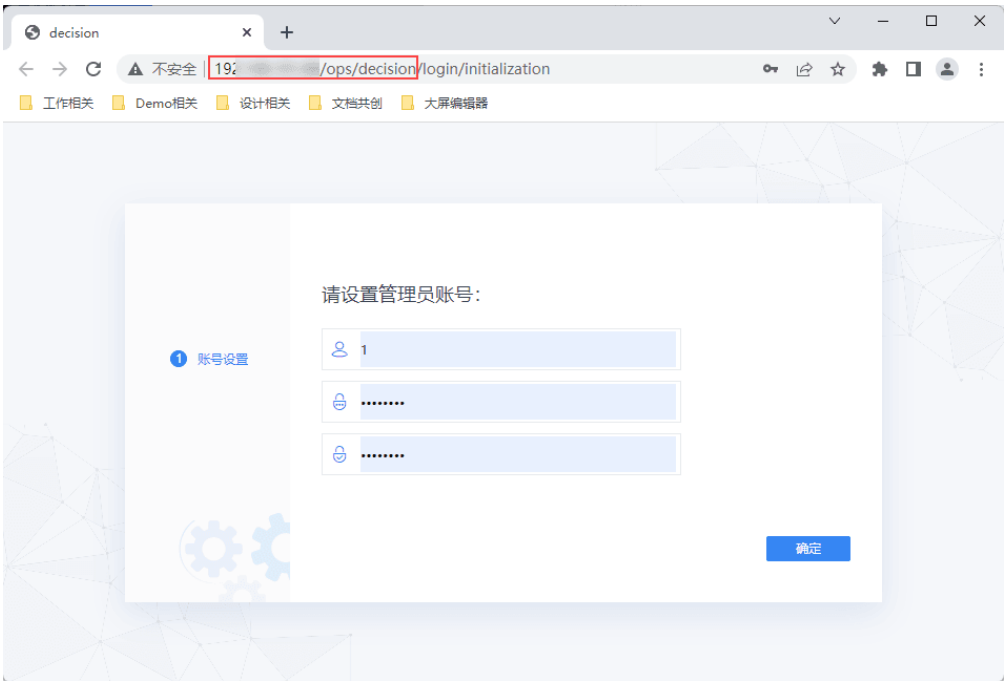
```
[root@localhost bin]# cd /home/finekey/bin/
[root@localhost bin]# ./finekey
INFO : 2023/03/28 11:33:05 begin install use config: /home/finekey/conf/finekey.yaml
INFO : 2023/03/28 11:33:05 use default ip:192.168.101.53
INFO : 2023/03/28 11:33:05 192.168.101.53 :use default user:root
INFO : 2023/03/28 11:33:05 deploymentType:2
INFO : 2023/03/28 11:33:05 begin check config valid!
INFO : 2023/03/28 11:33:05 The deployment mode is offline
INFO : 2023/03/28 11:33:05 192.168.101.53 :use local exec
INFO : 2023/03/28 11:33:05 over check config valid!

INFO : 2023/03/28 11:33:05 begin check net and port~
INFO : 2023/03/28 11:33:05 begin start server and bind ports in all nodes~
INFO : 2023/03/28 11:33:06 192.168.101.53 :time: 2023-03-28/11:33:06 +0800
INFO : 2023/03/28 11:33:06 No migration related configuration

INFO : 2023/03/28 11:33:06 192.168.101.53 :begin install docker~
INFO : 2023/03/28 11:33:06 finekey home:/home/finekey
INFO : 2023/03/28 11:33:08 unzip docker success!
INFO : 2023/03/28 11:33:09 copy docker to /usr/bin success!
INFO : 2023/03/28 11:33:10 docker daemon is starting~
INFO : 2023/03/28 11:33:13 docker daemon start success! docker ps:
```

- 1) 在浏览器地址栏输入：http://{ip}/ops/decision (请把{ip}换成自己的服务器 IP)
- 2) 回车，即可访问 FineOps 运维平台

访问  
4 运维  
平台



4. 部署新帆软项目（FR/BI/FDL）

面向对象：先部署好了运维平台，此时可以让运维平台帮助管理员部署并自动接入 FR/BI 应用。

本文介绍的是最简单的部署方案，因此对服务器配置要求较高，端口无法严格限制。

如无法提供全新服务器、网络端口限制较严格，请参考：[部署新项目](#)

步骤	说明
1 准备服务器	<p>单机</p> <p>1) 准备 1 台全新服务器，服务器不部署其他工程或内容，所有端口均未被占用</p> <p>2) 系统版本：CentOS 7.3~7.9 版本 / Ubuntu 18.04.4 及以上版本/ redhat 7.6 及以上版本</p> <p>3) 系统内核：3.10 及以上，fr/bi 最低 8 核，fdl 最低 4 核</p> <p>4) 系统架构：X86_64</p> <p>5) 剩余空闲磁盘：使用命令「df -h」查询，服务器/root/data 目录所在分区剩余空闲磁盘大于：fr/fdl 50 G，bi 100G</p> <p>6) 剩余空闲物理内存：fr 16G 以上、bi 32G 以上、fdl 8G 以上</p>

7) 网络要求:

- 所有服务器与运维平台所在服务器内网互通
- 对外开放 8080 端口, 确保用户可通过 8080 端口访问帆软应用

8) 用户: 使用 root 用户完成部署

集群

1) 按照「工程节点数+1」准备全新服务器数量, 服务器不部署其他工程或内容, 所有端口均未被占用

2) 系统版本: CentOS 7.3~7.9 版本 / Ubuntu 18.04.4 及以上版本/ redhat 7.6 及以上版本

3) 系统内核: 3.10 及以上, frbi 最低 8 核, fdl 最低 4 核, 集群组件服务器最低 4 核

4) 系统架构: X86\_64

5) 剩余空闲磁盘: 使用命令「df -h」查询, 每台服务器/root/data 目录所在分区剩余空闲磁盘大于:

- fr/fdl 50 G, bi 100G, 集群组件服务器 50 G
- fdl 如需使用管道功能, 集群组件服务器剩余空闲磁盘建议大于 1T

6) 剩余空闲物理内存:

fr 16G 以上、bi 32G 以上、fdl 8G 以上、集群组件服务器 8G 以上

7) 网络要求:

- 所有服务器与运维平台所在服务器内网互通
- 集群组件服务器对外开放 80 端口, 确保用户可通过 80 端口访问帆软应用

8) 用户: 使用 root 用户完成部署

FDL 额外：

联系帆软技术支持，获取 FineDataLink、Kafka 相关镜像包

1) 管理员登录 FineOps 运维平台，点击「运维管理>项目管理」，点击「部署新项目」



2) 进入「项目设置」界面，设置以下信息，点击「下一步」

自定义项目名称、选择应用类型、全选自定义部署组件、其他无需变更

部署帆软项目

2



3) 进入「节点配置」

单机新增 1 个主应用节点，集群按照工程数量新增主应用节点和 1 个组件节点。

输入主机、端口、用户名密码，挂载路径无需修改。



4) 点击「开始部署」，进入部署过程。



通过运维平台成功部署的新项目， 自动接入运维平台的项目管理列表中

其中的访问地址， 即为该帆软项目的访问地址， 在浏览器新窗口访问该地址，  
即可进入工程初始化配置界面

访问帆软项目



### 5. 接入已有项目

面向对象：先部署好了 FR/BI 应用， 后部署了运维平台。为了让运维平台管理 FR/BI 应用， 需要让运维平台接入部署好的应用。

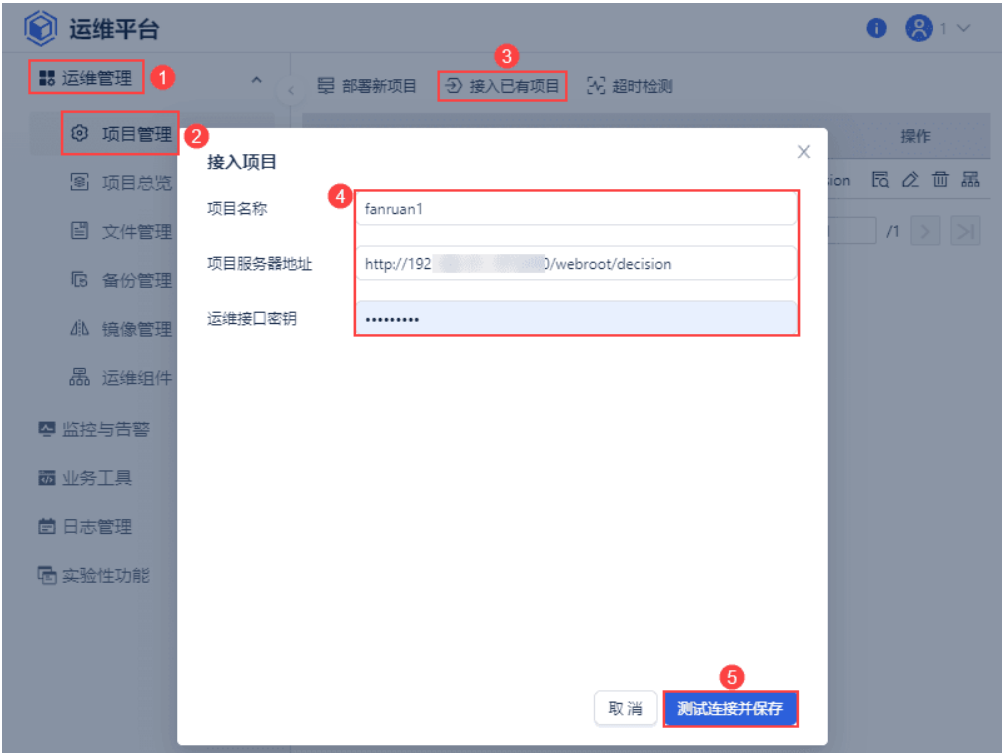


由于容器化部署和非容器化部署的项目，在接入时的步骤不完全相同，因此第一步需要判断工程的部署方式，然后再选择不同的接入方案。

5.1 接入容器化部署的项目

详情请参见：[接入已有容器化项目](#)

步骤	说明
检查是	
1 否支持 运维平台和需要被管理的 FR/BI 应用，所在的服务器必须内网互通	
接入	
	1) 获取 FR/BI 应用的访问地址，形如： <code>http://ip:port/webroot/decision</code>
	2) 获取 FR/BI 应用的运维接口密钥
	管理员登录 FineReport/FineBI 应用，点击「管理系统>智能运维>健康巡检>运维工具」。设置一个运维接口密钥，点击「保存」。
2 准备材料	
	1) 管理员登录运维平台，点击「运维管理>项目管理」，点击「接入已有项目」按钮。
3 接入已有项目	2) 自定义项目名称，填入第二步准备的信息。
	3) 点击「测试连接并保存」。



连接成功后提示「接入项目成功」，在项目管理列表中出现新接入的项目，即代表接入成功



## 5.2 接入非容器化部署的项目

详情请参见：[接入已有非容器化项目](#)

步骤		说明
检查是否支持接入	1	运维平台仅支持接入 FineReport11.0.6/FineBI5.1.22 及以上版本工程
	2	运维平台和需要被管理的 FR/BI 应用，所在的服务器必须内网互通
	3	请确保应用部署在 Linux/Windows 系统的 Tomcat 中间件中。
	4	请确保 FR/BI 应用安装了「系统运维」插件。建议将插件升级到最新版本。
项目信息设置	是否必要前提准备	必要步骤
	帮助运维平台与 FR/BI 应用连接	1) 获取 FR/BI 应用的访问地址，形如：http://ip:port/webroot/decision 2) 获取 FR/BI 应用的运维接口密钥

管理员登录 FineReport/FineBI 应用，点击「管理系统>智能运维>健康巡检>运维工具」。设置一个运维接口密钥，点击「保存」。



- 1) 管理员登录运维平台，点击「运维管理>项目管理」，点击「接入已有项目」按钮。
- 2) 自定义项目名称，填入上一步准备的信息。
- 3) 点击「测试连接并保存」。



非必要步骤

组件信息设置  
3 是否必要  
在运维平台所在服务器安装 exporter 组件，为该工程的外接数据库、负载均衡、状态服务器进行日志采集  
可安装的 exporter 组件包括：mysqld\_exporter、redis\_exporter、nginx\_exporter  
如不安装，可直接点击「跳过」进入下一步，不影响运维平台与工程的对

接，但后续运维平台将无法监控这些组件的状态并进行日志采集

- 1) 确认 FR/BI 应用中是否有符合要求的外接数据库、负载均衡、状态服务  
器组件

前提  
准备

- 如果没有这些组件，就完全可以跳过这个步骤
- 2) 确认 MySQL、Redis、Nginx 组件的端口可与运维平台内网联通
- 3) 准备 Nginx 组件的服务器地址、端口、SSL 信息
- 4) 为 FR/BI 应用安装 Nginx 状态监控模块

- 1) 检查自动设置的 MySQL 和 Redis 组件信息
- 2) 手动填写 Nginx 组件信息

操作  
步骤

接入项目

✓ 项目设置

2 组件设置

3 服务器设置

提示：非运维平台部署的项目可选填组件、服务器设置

▼ 外接数据库组件

MySQL组件名称

mysql

MySQL地址

192.168.101.218

MySQL端口

3306

用户名

root

▼ Redis组件

Redis组件名称

redis

Redis地址

192.168.101.201

端口

17001

取消

跳过

测试连接并继续

非必要步骤

在 FR/BI 应用所在服务器安装 node\_exporter 组件，为该工程进行服务器信  
息采集

是否  
必要  
4 服务器  
信息  
设置

可以为应用的每一个节点、每一个组件所在服务器都安装一份 node\_expor  
ter 组件

如不安装上述组件，可直接点击「跳过」进入下一步，不影响运维平台与  
工程的对接，但后续运维平台将无法监控服务器状态并进行日志采集

前提  
准备

1) 准备所有需要安装组件的节点服务器的主机 IP、端口、root 用户密码

2) 在所有需要安装组件的节点服务器，都准备好一个目录，用于存放 ex

24

porter 日志

- 2) 准备 node\_exporter 需要占用的端口可用性
- 3) 确保 node\_exporter 端口可与运维平台内网联通

将上一步准备的所有服务器的相关信息填入

操作  
步  
骤

接入项目

✓ 项目设置

✓ 组件设置

3 服务器设置

+

新增服务器

▼ 服务器1

主机

192.168.5.12

端口

22

用户名

root

验证方法

密码

密码

•

安装路径

/opt/Katherine/fr11

▼ 服务器2

主机

192.168.5.12

端口

22

用户名

root

验证方法

密码

密码

•

安装路径

/opt/Katherine/fr11

注：服务器信息配置用于安装日志采集和服务端信息采集组件，不希望安装可以选择跳过，此时将无法监控服务器状态并进行项目日志采集

取消

上一步

跳过并保存

测试连接并保存

检测以上步骤是否有问题， 没问题的情况下会成功接入

在「项目管理」界面会出现新接入的 FR/BI 项目

5 接入成功

接入项目

✓ 项目设置

✓ 组件设置

3 服务器设置

+

新增服务器

▼ 服务器1

主机

192.168.5.12

端口

22

用户名

root

验证方法

密码

密码

•

安装路径

/opt/Katherine/fr11

▼ 服务器2

主机

192.168.5.12

端口

22

用户名

root

验证方法

密码

密码

•

安装路径

/opt/Katherine/fr11

注：服务器信息配置用于安装日志采集和服务端信息采集组件，不希望安装可以选择跳过，此时将无法监控服务器状态并进行项目日志采集

取消

上一步

跳过并保存

测试连接并保存

6. 首次使用运维平台管理项目时需要做什么？

本节分为两类操作：

一类是管理员首次需要配置好的、让运维平台定期自动执行的操作。

一类是管理员需要手动定期为项目进行检查的一些内容。

6.1 配置自动化规则

建议操作	说明
	<p><b>应用场景：</b></p> <p>定期自动对项目进行整体异机备份，防止工程服务器出现意外，工程文件出现损毁后有办法快速恢复</p> <p><b>功能简介：</b></p> <p>在「运维管理&gt;备份管理」中选择运维项目，开启自动备份，设置备份频率</p> <p>设置自动备份</p> <p><b>参考文档：</b> <a href="#">备份管理</a></p>



**应用场景：**

运维平台监控到项目出现异常时，可以通过 email、webhook、企业微信通知管理员

因此管理员需要先设置好告警对象，方便运维平台知道给谁发消息

设置告警对象	<p><b>功能简介：</b></p> <p>在「监控与告警&gt;告警管理&gt;告警通知设置」中设置告警对象，支持「email、webhook、企业微信、企业微信机器人、钉钉机器人、飞书机器人」多种方式</p> <p><b>参考文档：</b> <a href="#">告警方式</a></p>
--------	------------------------------------------------------------------------------------------------------------------------------------------------------



应用场景：

健康巡检，可以理解为定期为系统做体检，确保及时发现系统的问题并进行调整。

因此管理员需要设定好体检的周期，让运维平台自动定期对系统进行监控巡检。

功能简介：

在「业务工具>健康巡检」中开启系统自动检查，设置执行周期

参考文档：[健康诊断](#)

设置自动健康巡检



应用场景：

设置宕机处理策略

系统宕机之后：

需要第一时间进行重启，需要第一时间恢复进程， 以免影响业务用户使用

需要导出宕机日志，帮助管理员事后查看和分析宕机原因和优化方案

功能简介：

在「业务工具>宕机自动处理>宕机处理策略」中选择运维项目，开启宕机自动处理，设置宕机相关配置项

参考文档：[宕机处理策略](#)



6.2 定期手动检查

建议操作说明

应用场景：

为满足客户的新需求以及完善之前版本某些功能的不足之处，FineReport/FineBI 在不断地更新迭代，管理员需要定期检查项目组件是否正常运行、是否需要更新

功能简介：

该功能仅面向容器化部署的项目

在「运维管理>项目管理>组件管理」中可以检查

检查组件状态与更新





应用场景：

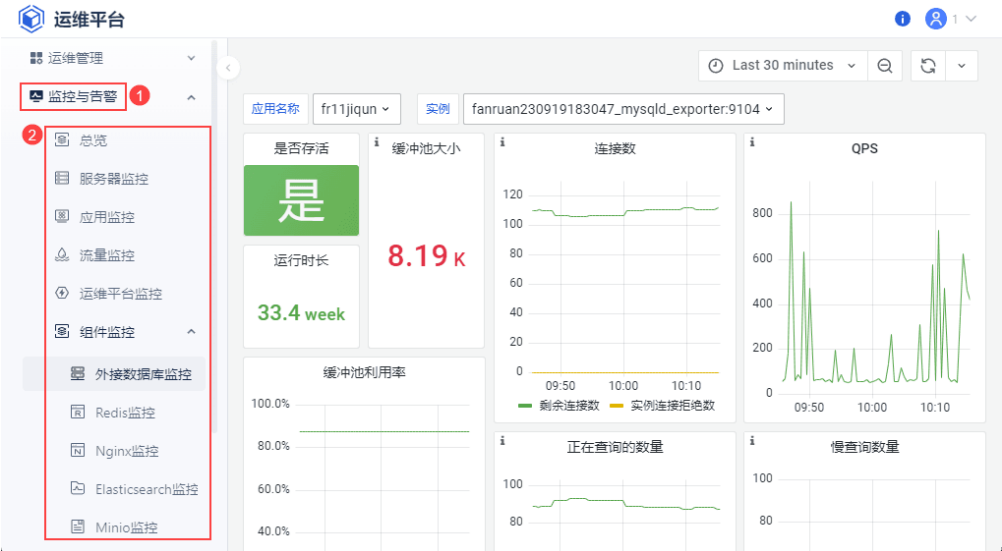
对于服务器、工程和工程中的组件，运维平台提供对应监控看板，监控各个指标情况

当指标出现异常时，会标黄/标红显示

功能简介：

在「监控与告警」中，可以查看各个监控看板，查看异常指标项

检查各类  
监控仪表  
板



应用场景：

运维平台会定期对 FR/BI 工程进行体检，管理员要根据体检报告改善工程配置

功能简介：

在「业务工具>健康巡检」中，查看是否存在异常项目，根据修改建议优化工程

检查健康  
巡检异常  
项



6. 项目出现异常时，运维平台可以做什么？

问题	解决方案
当系统运行出现异常时，会触发系统的告警规则	
当触发告警后，会通过告警方式通知管理员	
管理员根据告警内容，优化问题，提前解决问题	



当系统宕机之后

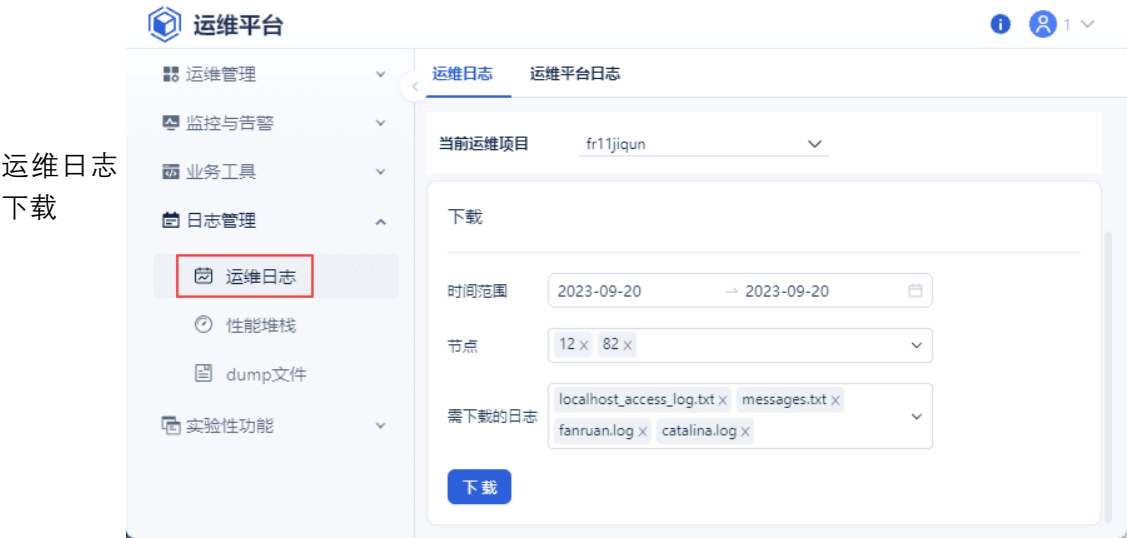
- 1) 运维平台可根据宕机处理策略，自动重启工程，恢复进程
- 2) 运维平台提供此次宕机的原因和处理方案，可根据方案优化



- 3) 宕机后自动导出堆栈日志，运维平台可对日志进行分析，可根据分析报告优化



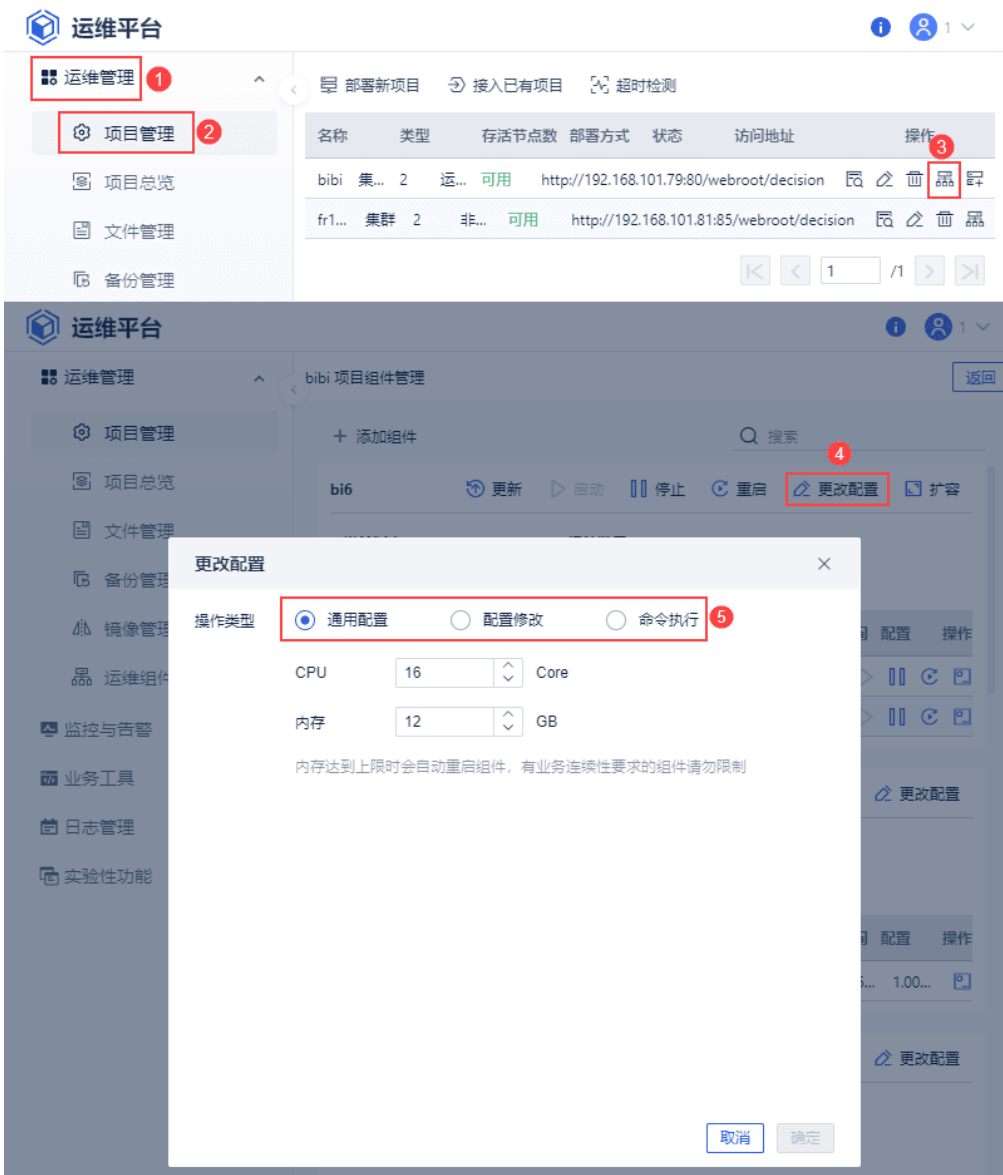
出现任何问题，需要分析工程日志时，可通过运维平台快速下载日志到本地进行分析



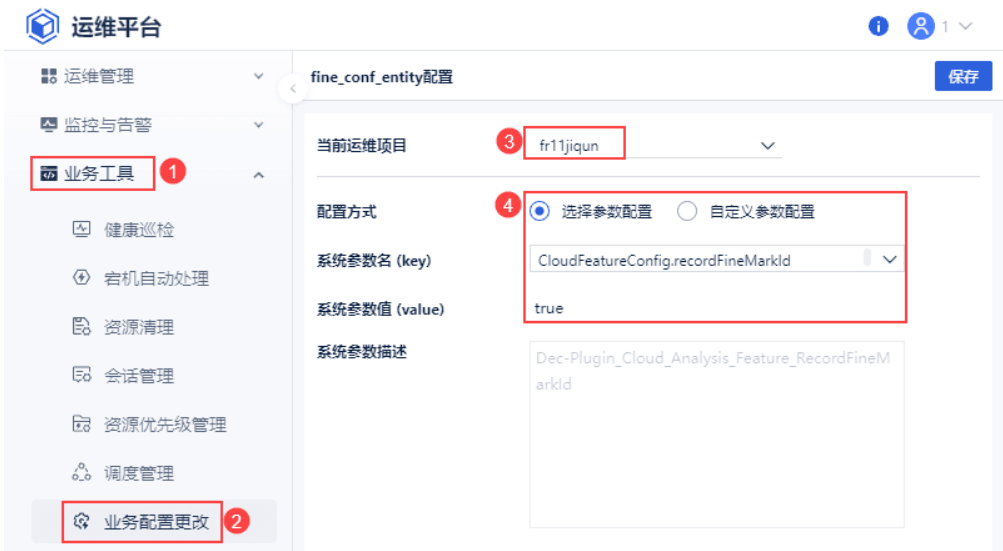
如果检测到是 FR/BI 工程或组件的配置不合理，需要进行调整时：

配置更新 1) 对于 FR/BI 项目组件，可通过组件管理功能对 Nginx/Redis/配置库进行与启停

配置修改



2) 对于 FR/BI 工程，可通过「业务配置更改」功能修改工程配置，确保工程平稳运行



## 1.2 更新日志索引

### V1.15.0

#### 部署运维平台

优化：部署运维平台时，调整 node\_exporter、ops\_agent 组件默认端口，防止与运维项目冲突

#### 部署运维项目

新增：支持对运维平台部署的项目，界面化上传授权镜像，便捷地进行容器私有云注册

优化：部署运维项目时，支持填写外网 IP，以适配不支持内网访问、仅可外网映射的服务器

优化：部署运维项目时，支持「中止部署」，自动停止本次部署并回退已有操作

优化：部署 FineBI 项目时，bi 组件默认 Xmx 限制提升，降低 oomkill 风险

#### 日志管理

优化：下载性能堆栈时，支持选择日志类型，定向下载指定内容

#### 实验室功能

优化：支持自定义链路追踪数据留存天数，满足个性化分析诉求

优化：支持手动清空运维项目链路数据，防止磁盘不足引起异常

**2023-12-14 发布，详情请参见：[V1.15.0 更新日志](#)**

### V1.14.0

#### 部署升级

优化：部署运维平台时，若检测到挂载磁盘空间不足，可快速更换部署路径

优化：部署运维项目时，部署的配置库组件，由 MySQL5 调整为 MySQL8

#### 监控与告警&日志管理

优化：告警任务触发逻辑与宕机自动生成性能堆栈逻辑匹配，防止造成理解误差

#### 实验室功能

优化：调用链路展示详情优化，支持筛选 span，支持查看自助数据集信息

**2023-12-01 发布，详情请参见：[V1.14.0 更新日志](#)**

V1.13.0

**运维管理**

新增：运维平台部署的项目，FineOps 支持界面化为项目 Nginx 组件配置 SSL

优化：运维平台 SSL 配置入口优化，调整到运维组件-Nginx 组件中，方便查找

**日志管理**

优化：项目运维日志新增覆盖多种 FineBI 日志，便于精确分析性能问题

优化：运维平台日志新增覆盖多种运维组件日志，方便用户获取 FineOps 自身运行情况

**2023-11-17 发布，详情请参见：[V1.13.0 更新日志](#)**

V1.12.0

**部署升级**

新增：部署运维平台成功后，提示工程地址，方便用户访问

**运维管理**

新增：组件管理支持导出容器化部署信息，方便后期运维

优化：文件管理支持新建文件夹、批量上传、上传.fvs 和.mfvs 格式文件、新增同名覆盖提示

优化：备份还原磁盘空间检查优化，升级和还原前备份检查优化，防止备份占用磁盘过高

**日志管理**

新增：支持手动上传本地性能堆栈，至运维平台进行分析

优化：支持展示历史性能堆栈生成记录

优化：性能堆栈使用顺序优化，变更为：分析>下载>生成

**实验室功能**

新增：若某个调用链路所在 session 有相关首屏链路数据，则支持联动跳转「查看首屏链路」

优化：系统健康观测联动跳转链路详情时，部分跳转项从「首屏链路」调整为「调用

链路」

2023-11-01 发布，详情请参见：[V1.12.0 更新日志](#)

V1.11.0

部署升级

新增：新增「版本更新」提醒，帮助用户及时享受更多功能和更优性能

新增：新增浏览器版本校验，避免使用低版本浏览器，导致运维平台界面异常

运维管理

优化：部署/接入项目时，项目名称限制放宽，新增支持英文大写、中文

优化：部署新项目时，访问路径支持为空，方便配置工程短链接

优化：部署新集群项目时，支持使用外部共享目录作为文件服务器

监控与告警

新增：组件监控新增「加速引擎监控」，帮助管理员监控管理 BI 大数据引擎组件

优化：新增默认告警「运维平台挂载磁盘预警」，覆盖通用运维场景

实验室功能

新增：调用链路详情新增「分析汇总」视图，帮助用户快速洞察耗时分布

优化：调用链路详情新增「链路信息」按钮，汇总展示链路关键信息

优化：调用链路列表新增「FGC」标识，标记链路时间范围内是否发生 FullGC

2023-10-17 发布，详情请参见：[V1.11.0 更新日志](#)

V1.10.0

监控与告警

系统健康监测：界面分区优化，新增全局控件：模块切换按钮、请求类型筛选按钮，交互更加合理

应用监控：新增观察指标，帮助用户排查线程泄露场景：总线程使用量监控、线程类别 Top10

告警管理：新增三种告警通知方式，满足用户多样化需求：企业微信机器人、钉钉机器人、飞书机器人

业务工具

健康巡检：集群运维项目，支持使用健康巡检快速修复功能

**2023-09-28 发布，详情请参见：**[V1.10.0 更新日志](#)

## V1.9.0

### 部署升级

升级 FineOps：升级 FineOps 时检测并升级 docker 引擎版本，防止出现后续使用 bug

### 运维管理

项目管理：部署新项目 and 接入已有项目交互界面优化，拆分默认部署组件和自定义部署组件

项目管理：部署新项目失败再次部署，自动填充上次部署的项目设置和节点配置，节约用户时间

项目管理：接入已有容器化项目步骤精简，删除冗余操作步骤：组件信息设置、服务器信息设置

项目管理：支持从组件管理处添加应用节点，集群项目扩容操作步骤更流畅

### 监控与告警

告警管理：默认告警「负载过高告警——节点无响应」更名为「应用卡顿告警——节点无响应」，降低用户理解成本

### 业务工具

健康巡检：为降低用户优化工程的难度，健康巡检部分检查项支持快速修复

**2023-09-07 发布，详情请参见：**[V1.9.0 更新日志](#)

## V1.8.0

### 运维管理

项目管理：支持容器化部署全新的 FineDataLink 集群项目

项目管理：支持编辑项目访问地址，将自动生成的访问地址修改为实际项目访问地址

项目管理：部署新项目、接入已有项目入口位置调整，更醒目

项目总览：新增项目对比功能，支持展示两个工程中 JAR 包和插件的不一致清单

### 业务工具



健康巡检：新增检测项 1026 密码强度限制检测

## 日志管理

性能堆栈：完善自动触发堆栈日志的场景，分析记录列表新增展示「触发类型」

## 实验室功能

调用链路：查询界面新增筛选项：端点、状态、实例

**2023-08-23 发布，详情请参见：**[V1.8.0 更新日志](#)

## V1.7.0

### 监控与告警

告警管理：支持自定义告警，支持自定义异常触发条件和达成告警的持续时间

告警管理：支持查看触发的告警任务的记录，帮助管理员快速识别常见问题、确定趋势和模式，并采取相应的措施来解决或预防潜在的故障或风险

告警管理：告警管理界面和交互展示完全重构，拆分为：告警任务、告警记录、告警通知设置三个模块

### 实验室功能

链路追踪：对于使用 elasticsearch 套件获取和存储日志的运维项目，在查看链路详情时，支持获取日志到本地

**2023-08-07 发布，详情请参见：**[V1.7.0 更新日志](#)

## V1.6.0

### 部署升级

项目管理：容器化部署集群项目时，必须安装 elasticsearch 套件，确保集群日志存储的稳定性

项目管理：项目管理列表中，部署方式措辞优化，展示为：非运维平台部署/运维平台部署

项目总览：新增组件拓扑视图，以拓扑形式展示项目内各服务、组件间的网络结构及联通性

运维组件：支持界面化操作，为运维平台的 Nginx 配置 SSL 证书，以满足使用 HTTPS 访问的需求

## 监控与告警

流量监控：新增流量监控，以拓扑视图展示项目内各服务各节点间请求流量关系，排查性能、联通性等问题

## 业务工具

宕机自动处理：内存堆栈导出记录，新增「触发类型」，记录自动导出堆栈的触发原因，方便后续分析

**2023-07-21 发布，详情请参见：[V1.6.0 更新日志](#)**

## V1.5.9

### 项目管理

项目扩容：提供「添加节点」和「工程扩容」功能，支持为容器化集群运维项目，前端界面化扩充工程节点

### 运维管理

项目总览：提炼出用户最关心的信息和最常用的操作，组合成了「项目总览」界面

## 业务工具

健康巡检：磁盘检查新增检测项「0509 顺序读取(非缓冲池)、0510 顺序写入(非缓冲池)」，删除检测项「0503 随机读」

## 监控与告警

系统资源监控：精选新增高频关注指标，剔除部分冗余指标，重新划分为：全项目状态总览、基础资源监控、线程池监控

系统健康监测：新增「健康巡检」强提醒按钮，各指标图和明细表标题联动显示时间点，快速定位时间

服务器监控：「服务器明细」中新增指标「TCP 排队数据量」

应用监控：新增指标「系统资源优先级监控」，删除指标「线程死锁数」、「temp 目录磁盘余量」，删除「BI 资源监测」相关内容

## 日志管理

运维日志：支持查看各节点运维日志存储位置

运维日志：为了节约用户下载耗时，支持用户自定义选择需下载的日志类型

2023-07-06 发布，详情请参见：[V1.5.9 更新日志](#)

V1.5.7

部署接入

部署应用：为了降低用户使用运维平台相关功能的难度，运维组件安装后全部启动

运维功能

日志管理：日志管理功能重新组合拆分，划分为三个模块：运维日志、性能堆栈、dump 文件

日志管理：对于多节点的集群运维项目，支持单次下载多节点运维日志

日志管理：各类日志下载时，支持选择服务类型，按需下载日志

链路追踪：非容器化部署的项目，开启链路追踪步骤简化，交由 FineOps 运维平台一键配置，用户只需重启工程

链路追踪：支持对运维项目快速停用链路追踪功能，重启项目后生效

链路追踪：链路追踪查询界面新增「重置」按钮，一键清空所有查询条件，方便重置查询。

链路追踪：链路列表展示优化，支持展开收起，方便放大记录明细区

链路追踪：调用链路详情区，支持根据耗时，筛选符合要求的链路

2023-06-14 发布，详情请参见：[V1.5.7 更新日志](#)

V1.5.6

部署接入

部署应用：安装包移除了部分冗余运维组件镜像，为部署瘦身，让用户更便捷、快速的部署运维平台

升级项目：运维平台升级成功重启后，自动一键升级运维组件，确保整体升级成功

接入项目：帆软应用只允许被一个运维平台接入。若已被接入，应用的运维密钥处显示接入状态为「已连接」

运维功能

系统健康观测：系统健康观测界面展示优化，各指标图支持互相定位当前时间、支持拖动调整时间区间、支持放大查看

健康巡检：为了满足管理员对系统的多维度检测需求，健康巡检新增多个检测项

日志下载：支持下载运维平台自身日志，方便管理员在运维平台功能出现问题时定位问题。

日志分析：删除「日志分析」功能，不再提供对出错日志、业务分析日志进行分析的相关功能

**2023-05-31 发布，详情请参见：[V1.5.6 更新日志](#)**

## V1.5.5

运维平台正式发布的初始版本

### 部署接入

部署应用：FineOps 运维平台分别提供内外网部署包，支持在线/离线部署。支持在线/离线升级

部署项目：FineOps 运维平台支持一键容器化部署并接入 FineReport/FineBI/FineDataLink

接入项目：FineOps 运维平台支持手动接入并管理已有 FineReport/FineBI/FineDataLink

升级项目：对于管理的运维项目，FineOps 支持在线/离线升级到最新/指定版本

管理项目：对于管理的运维项目，FineOps 支持启停管理各个组件、支持修改组件配置

### 运维管理

项目管理：提供多项目管理的能力，进行统一监控运维管理，解决多项目运维管理混乱的痛点

文件管理：支持前端界面化管理集群文件服务器，方便上传下载文件

备份管理：支持对项目进行一键备份、整体备份、异机备份

集群节点管理：提供前端界面管理集群节点，支持修改节点信息

镜像管理：支持对接镜像仓库，用于后续运维/项目组件更新

运维组件：支持前端界面化，对运维组件进行启停管理、配置修改

### 监控与告警

监控总览：提供总览监控大屏，用于关注各个运维项目的全局状态

服务器监控：提供服务器监控大屏，用于监控帆软应用及相关组件运行的服务器的状

态

应用监控：提供应用监控大屏，针对应用单个节点的运行状态的详细数据进行查询

运维平台监控：提供运维平台监控大屏，监控运维平台中各个运维组件的运行情况

组件监控：提供组件监控大屏，监控外接数据库、Redis、Nginx、Elasticsearch、Minio 组件

系统健康观测：对帆软应用的健康状态和运行状态进行智能化监测

告警管理：内置多种告警规则，支持通过邮箱、webhook 和企业微信，及时自动推送异常情况给管理员

## 业务工具

健康巡检：对应用进行定期健康巡检，确保应用所在的环境及应用内配置合理，以保证应用的正常运行

宕机自动处理：支持为应用配置宕机处理策略，支持自动宕机重启，自动导出 dump，快速定位宕机原因

资源清理：提供磁盘清理功能，支持手动/自动清理冗余文件，预防磁盘宕机问题

会话管理：支持界面化查看应用会话占用情况，支持对异常会话手动中止

资源优先级：一个系统中存在很多模板，但是业务的优先级有先有后，FineOps 支持对部分重要业务的资源进行优先保证

调度管理：帮助用户了解系统中的调度任务和系统压力，促进任务配置优化

业务配置更改：提供安全方式修改相关应用底层配置

## 日志管理

日志下载：用于下载应用相关日志以及应用环境信息

日志分析：帮助用户对错误堆栈、出错日志、业务分析日志进行分析并给出相关改进意见

**2023-05 发布，详情请参见：**[运维平台简介](#)

# 1.2.1 V1.15.0 更新日志

## 1. 概述

### 1.1 版本

运维平台版本	更新日期
V1.15.0	2023-12-14

## 2. 部署运维平台

### 2.1 调整运维平台组件默认端口

调整部分运维组件默认端口，防止与运维项目冲突

- node\_exporter 组件默认端口由 9100 调整为 9101
- ops\_agent 组件默认端口由 9070 调整为 9071

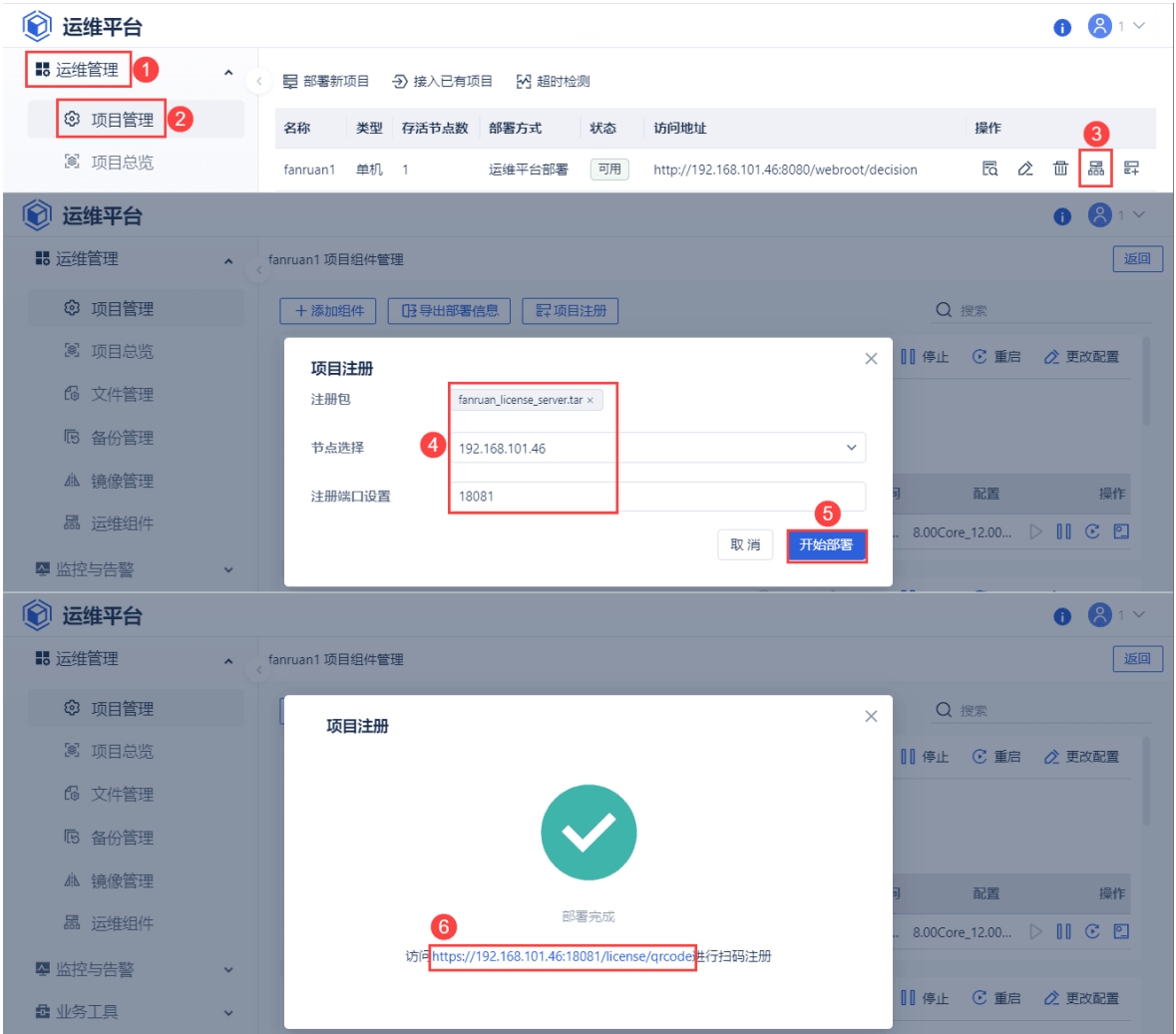
详情请参见：[部署运维平台](#)

## 3. 部署运维项目

### 3.1 支持界面注册运维项目

对运维平台部署的项目，支持界面化上传授权镜像，便捷地进行容器私有云注册。

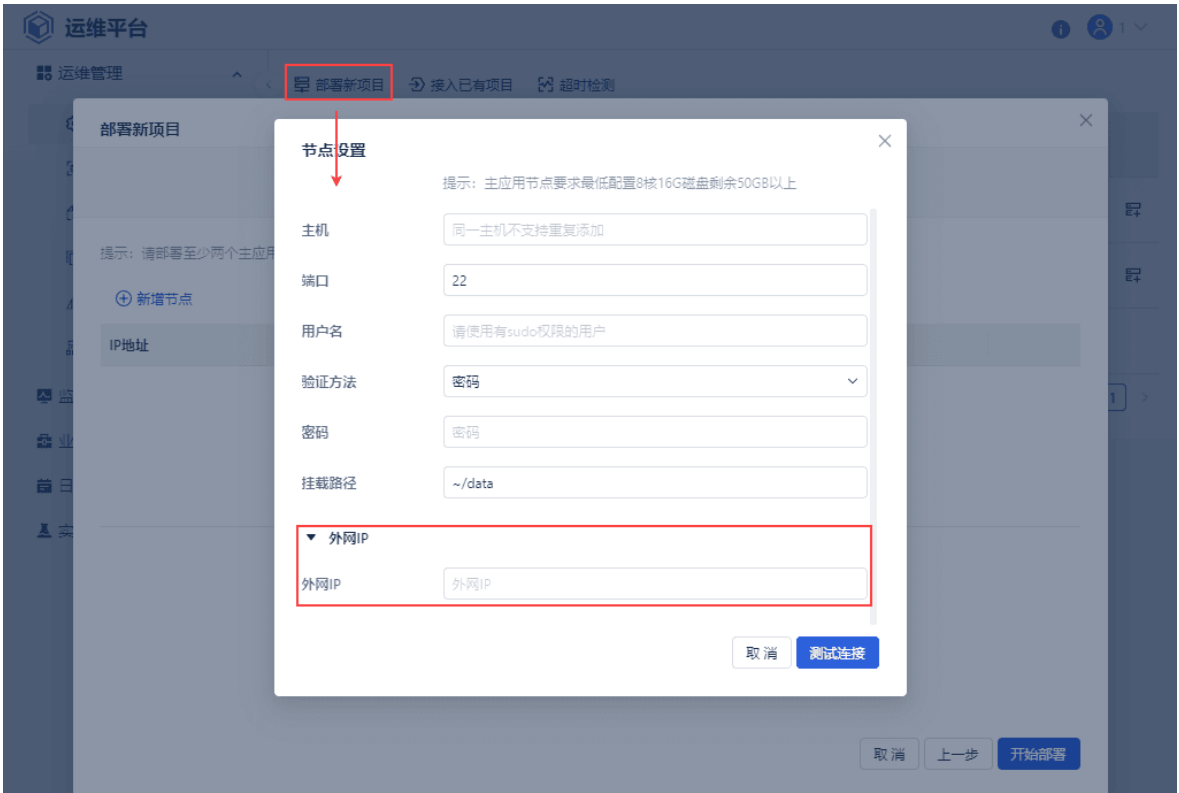
详情请参见：[注册新项目](#)



### 3.2 部署运维项目支持填写外网 IP

部署运维项目时, 支持填写外网 IP, 以适配不支持内网访问、仅可外网映射的服务器。

详情请参见：[部署新 FineBI 集群项目](#)



### 3.3 支持中断部署运维项目

部署运维项目时, 支持「中止部署」, 自动停止本次部署并回退已有操作。

详情请参见：[部署新 FineBI 集群项目](#)

新项目部署

✓ 拉取镜像

2 执行部署

正在部署，请稍后。  
需部署组件数量：8，已完成数量：3

ops_agent	✓ 部署成功	-
mysql	✓ 部署成功	-
mysqld_exporter	✓ 部署成功	-
elasticsearch	部署中...	
elasticsearch_exporter	部署中...	
node_exporter	部署中...	
bi6	部署中...	
flink	部署中...	

中止部署

隐藏至后台进行

3.4 FineBI 组件默认 Xmx 限制提升

容器化部署 FineBI 运维项目时，部署的 FineBI 组件，默认 Xmx 限制提升，降低 oomkill 风险

- 优化前：默认 Xmx=机器内存减掉 8G 后的内存大小
- 优化后：默认 Xmx=机器内存减掉 10G 后的内存大小

详情请参见：[部署新 FineBI 集群项目](#)

4. 日志管理

4.1 支持选择堆栈日志类型下载

下载性能堆栈时，支持选择日志类型，定向下载指定内容。

详情请参见：[性能堆栈](#)





## 5. 实验室功能

### 5.1 支持自定义链路追踪数据留存天数

支持自定义链路追踪数据留存天数，满足个性化分析诉求。

详情请参见：[首屏链路](#)



### 5.2 支持手动清空运维项目链路数据

支持手动清空运维项目链路数据，防止磁盘不足引起异常。

详情请参见：[首屏链路](#)

运维平台

运维管理

监控与告警

业务工具

日志管理

实验性功能

链路追踪

资源隔离管理

全屏链路

调用链路

全局设置

链路追踪配置

数据留存天数

5

增加天数需要确保磁盘有充足的空间，当磁盘空间不足时会自动停止数据采集，可按项目数\*增加的数据留存天数\*10G来评估磁盘增量

采集次数上限

-1

即3秒内同一端点的采集次数上限，上限为10000，-1表示不限制

不采集的端点

POST:/decision/v5/conf/items/updateslock

POST:/decision/skywalking/\*\*

GET/

GET:/decision/metrics/\*

GET:/websockets/

GET:/decision/system/health

POST:/decision/remote/design/channel

+

新增

最大跨度限制

500

即采集单个端点的最大跨度数量，超出的将被舍弃，上限为10000

冗余数据过滤

1

对部分异常概率较低的信息提高采集门槛，如连接前校验，有助于凸显关键信息，-1表示不限制

保存

关闭链路追踪

提示：对项目关闭链路追踪后，需手动重启项目

bi容器化

链路追踪

链路数据

清空

fr容器化

链路追踪

链路数据

清空

fanruan2

链路追踪

链路数据

清空

# 1.2.2 V1.14.0 更新日志

## 1. 概述

### 1.1 版本

运维平台版本	更新日期
V1.14.0	2023-12-01

## 2. 部署升级

### 2.1 部署 FineOps 支持快速更改挂载目录路径

部署运维平台时，经常有用户因为挂载目录所在分区磁盘空闲空间不足，不得不中断部署，手动更改 yaml，再进行部署。

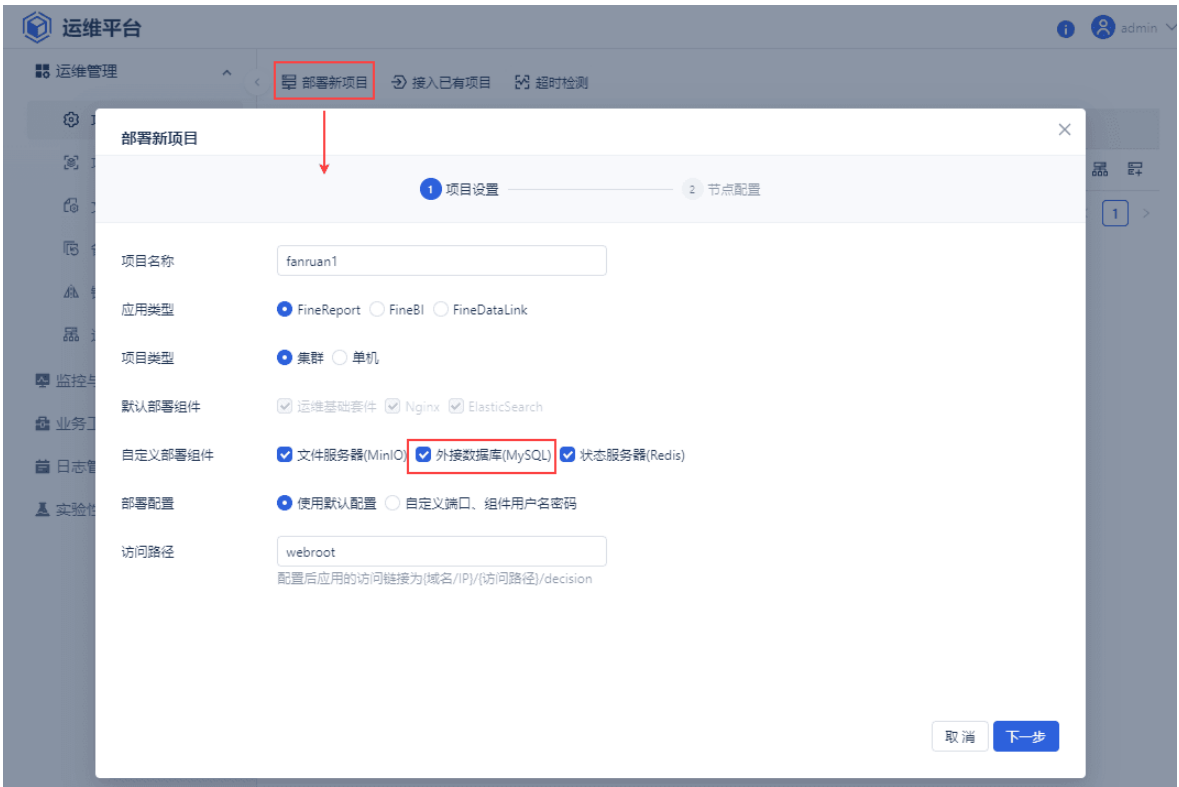
V1.14.0 及之后版本，如果部署过程中，检测到挂载目录磁盘可用空间不足 100G，用户无需退出部署环节，可直接输入满足条件的目录，继续部署。

详情请参见：[部署运维平台](#)

```
[root@k8s ops]# cd /home/ops/finekey/bin/
[root@k8s bin]# ./finekey
INFO : 2023/11/27 16:22:02 begin install use config: /home/ops/finekey/conf/finekey.yaml
INFO : 2023/11/27 16:22:02 localhost :use local exec
ERROR: 2023/11/27 16:22:04 node localhost path ~/data disk available capacity:30658212, less
then 100G
INFO : 2023/11/27 16:22:04 The current path capacity check failed, Please re-enter the path
Enter the folder path and press ENTER to confirm, or enter q then ENTER to exit:
```

### 2.2 部署运维项目配套组件优化

使用 FineOps 部署运维项目时，项目中默认安装的配置库组件，由 MySQL5 优化为 MySQL8，满足更多使用场景。



### 3. 监控与告警

#### 3.1 应用卡顿告警任务触发逻辑优化

为了防止造成理解误差，告警任务触发逻辑与宕机自动生成性能堆栈逻辑匹配。

内置告警任务「应用卡顿告警——节点无响应」触发逻辑优化。异常状态持续触发时长从 30s 优化为 60s。

详情请参见：[告警任务](#)



### 4. 日志管理

#### 4.1 项目宕机自动生成堆栈逻辑优化

为了防止造成理解误差，告警任务触发逻辑与宕机自动生成性能堆栈逻辑匹配。

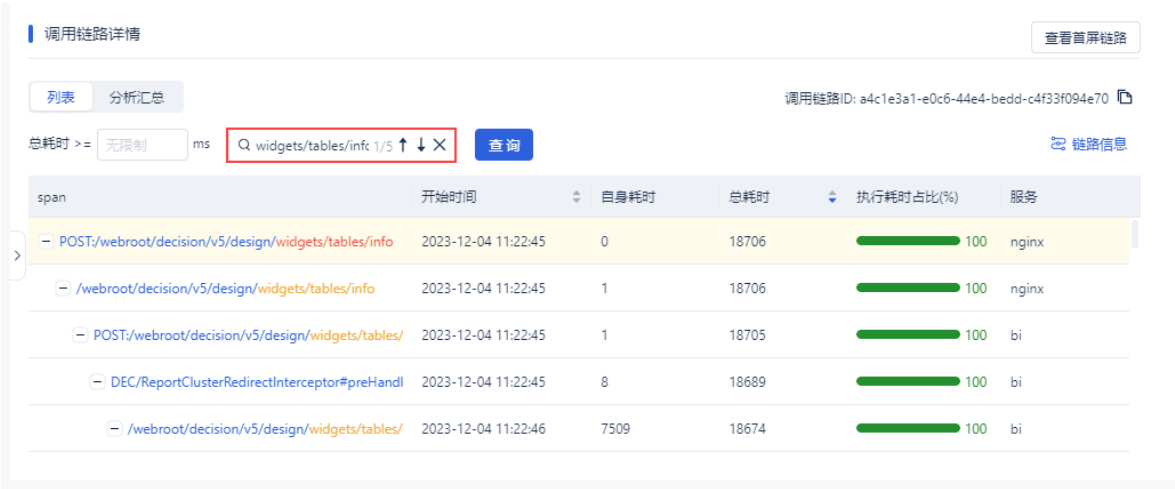
运维平台监测到项目宕机，自动触发生成堆栈日志条件优化。

- 1) 应用无响应触发时间由 3 分钟优化为 2 分钟
- 2) 新增 FullGC 触发条件：5min 内 FullGC 次数总和大于 3 次，且时间总和超过 50s

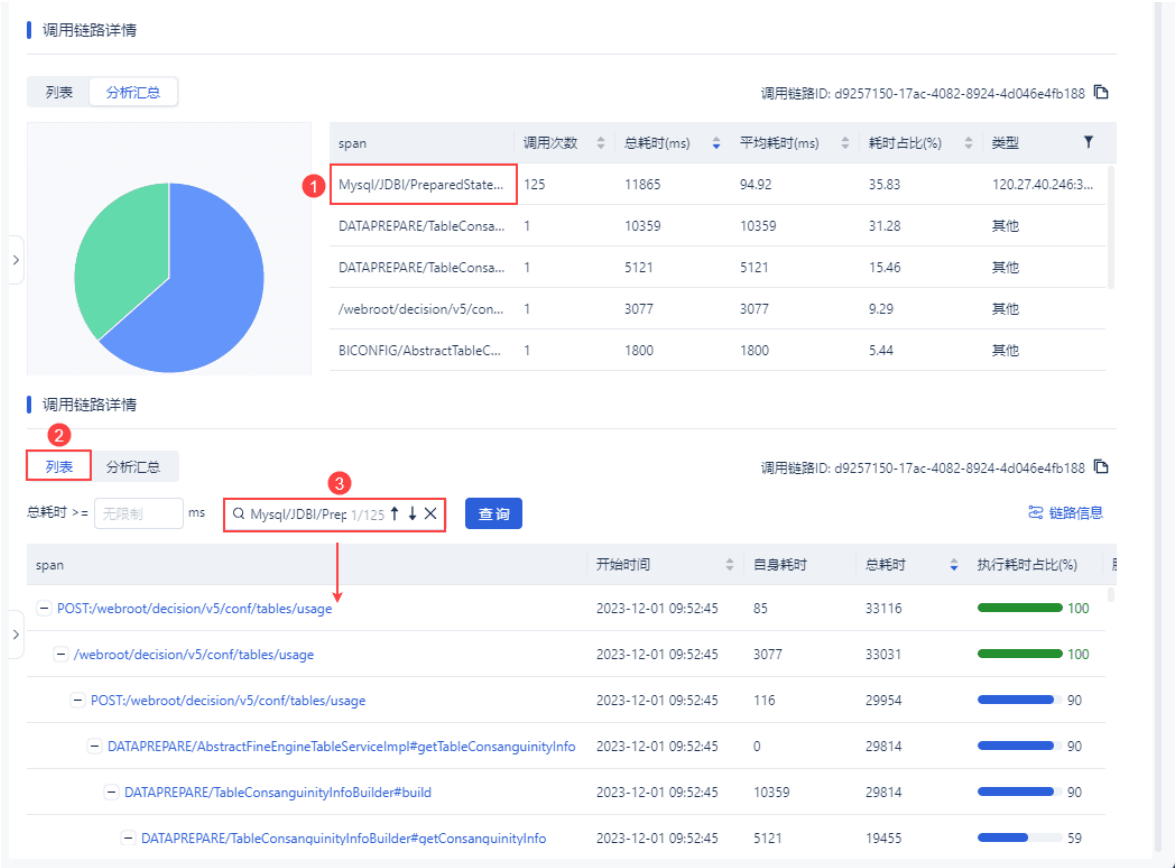
5. 实验室功能

5.1 调用链路展示详情优化

- 1) 调用链路详情「列表」视图下，链路信息中，新增「自助数据集」字段，若有即可查看。
- 2) 调用链路详情「列表」视图，新增「span 关键字查询」功能，支持搜索包含指定关键字的 span。



- 3) 调用链路详情「分析汇总」视图下，支持点击 span 名称跳转至「列表」视图下，以名称作为关键词进行搜索。



# 1.2.3 V1.13.0 更新日志

## 1. 概述

### 1.1 版本

运维平台版本	更新日期
V1.13.0	2023-11-17

## 2. 运维管理

### 2.1 支持为项目配置 SSL

运维平台部署的项目，FineOps 支持界面化为项目 Nginx 组件配置 SSL。

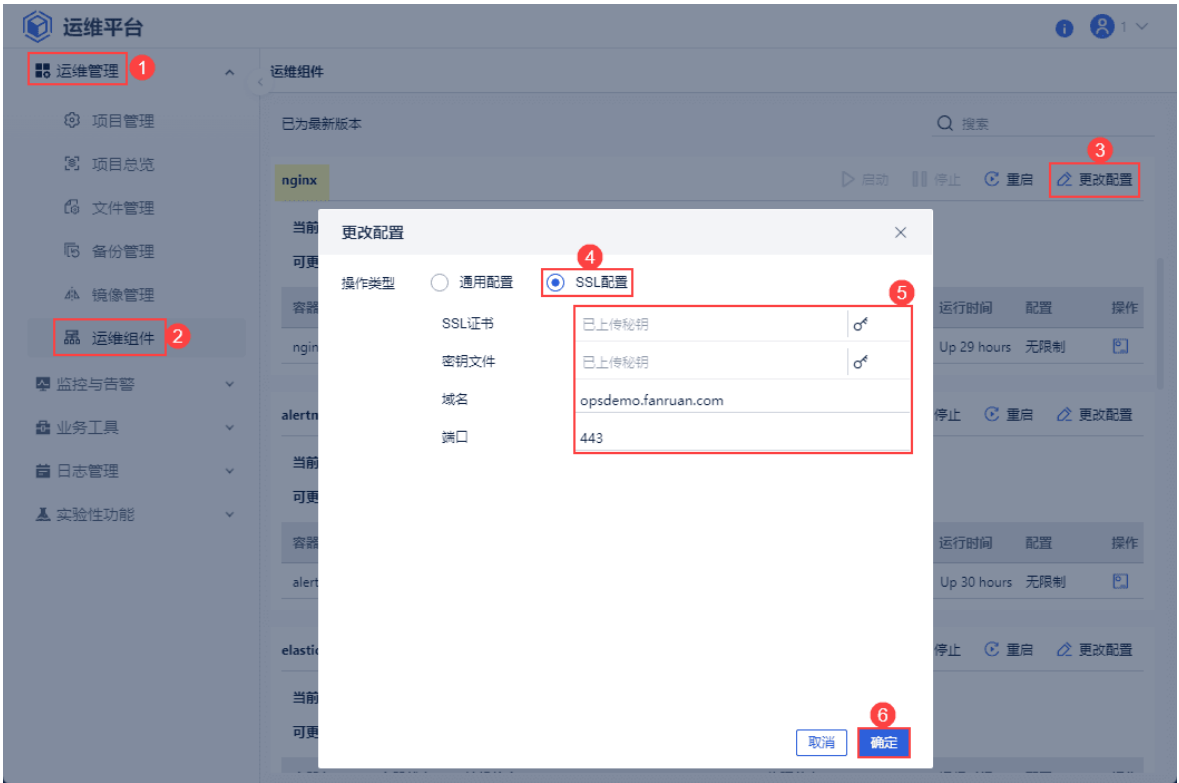
详情请参见：[为运维项目开启 SSL](#)



### 2.2 运维平台 SSL 配置入口优化

运维平台 SSL 配置入口优化，调整到运维组件-Nginx 组件中，方便查找。

详情请参见：[为运维平台开启 SSL](#)



### 3. 日志管理

#### 3.1 新增 FineBI 运维日志

项目运维日志新增覆盖多种 FineBI 日志，便于精确分析性能问题。

详情请参加：[运维日志](#)

日志	说明
	仅 FineBI 项目支持查看和下载该日志
eventlog	记录了数据集更新在 spark 引擎中计算的具体信息，用于分析 spark 执行慢问题
	仅 FineBI 项目支持查看和下载该日志
polars.log	仪表板和数据集编辑涉及的计算日志
	仅 FineBI 项目支持查看和下载该日志
operator_calculate.log	仪表板和数据集编辑涉及的计算日志
agent.log	ops-agent 组件日志

#### 3.2 新增多种运维组件日志

运维平台日志新增覆盖多种运维组件日志，方便用户获取 FineOps 自身运行情况。

详情请参加：[运维日志](#)

日志	说明
	docker 内容器的日志，包括以下容器：
dockerLogs.log	<ul style="list-style-type: none"><li>alertmanager</li></ul>

- elasticsearch
- elasticsearch\_exporter
- grafana
- nginx
- nginx\_exporter
- node\_exporter
- ops
- ops\_agent
- prometheus
- registry
- skywalking\_oap

access.log	Nginx 访问日志的访问日志文件，用于记录用户访问网站的请求信息，如访问时间、IP 地址、请求方法、访问路径、HTTP 状态码等
error.log	记录 FineOps 运维平台预期外的报错
fanruan.log	记录 FineOps 运维平台运行日志
agent.log	ops-agent 日志



## 1.2.4 V1.12.0 更新日志

### 1. 版本

运维平台版本	发布日期
V1.12.0	2023-11-01

### 2. 部署升级

#### 2.1 部署运维平台新增地址提示

部署运维平台成功后，提示工程地址，方便用户访问。

详情请参见：[部署运维平台](#)

```
INFO : 2023/10/31 14:48:24 192.168.101.198 :begin install ops_agent
INFO : 2023/10/31 14:48:24 docker image 192.168.101.198:5000/ops_agent:v1.12.0.1
exist

INFO : 2023/10/31 14:48:27 192.168.101.198 :begin install skywalking_oap
INFO : 2023/10/31 14:48:28 docker image 192.168.101.198:5000/skywalking_oap:v20.
1.4.3-9.1.0 exist
INFO : 2023/10/31 14:48:31 deployment success! The access address is: http://19
2.168.101.198:80/ops/decision . bye~
[root@k8s bin]#
[root@k8s bin]#
```

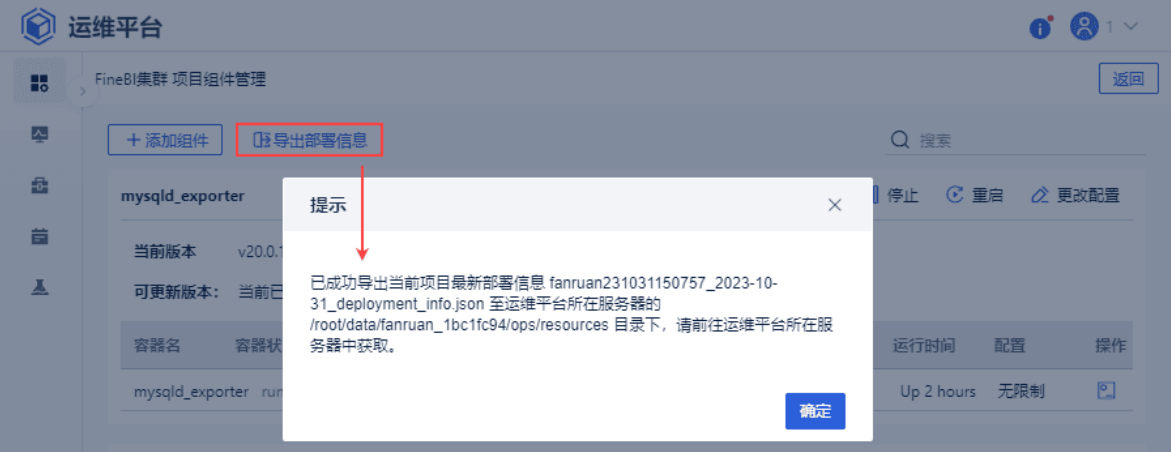
### 3. 项目管理

#### 3.1 支持导出项目部署信息

对于运维平台部署的组件/项目，在后期运维时，可能需要查阅原始部署信息。

可点击「导出部署信息」，导出 json 文件，用户可前往运维平台所在服务器中获取。

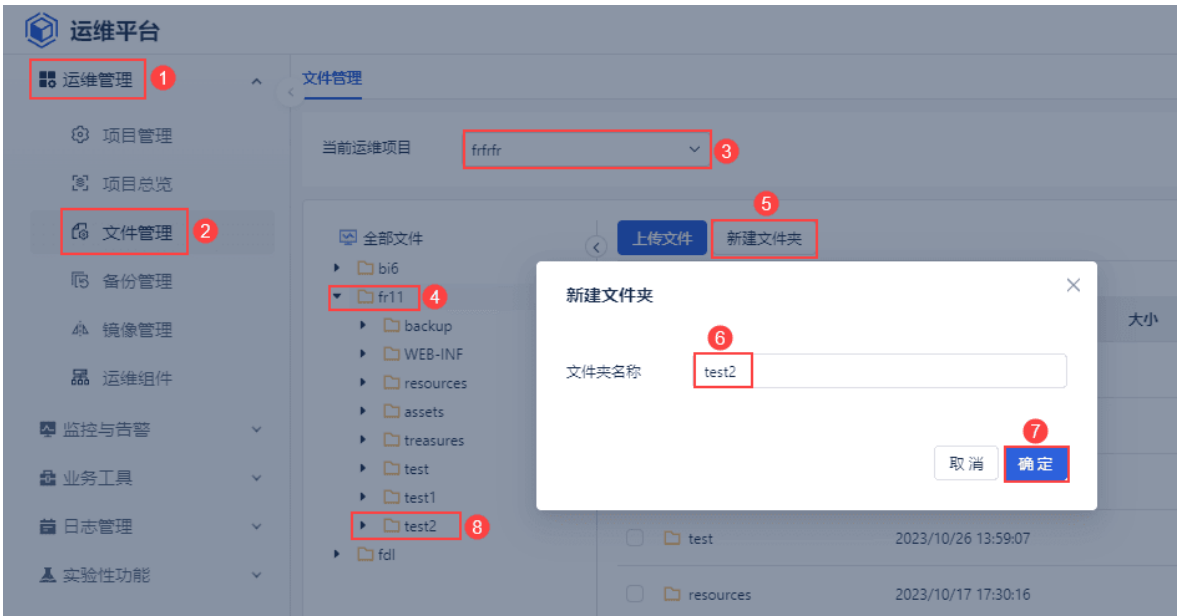
详情请参见：[组件管理](#)



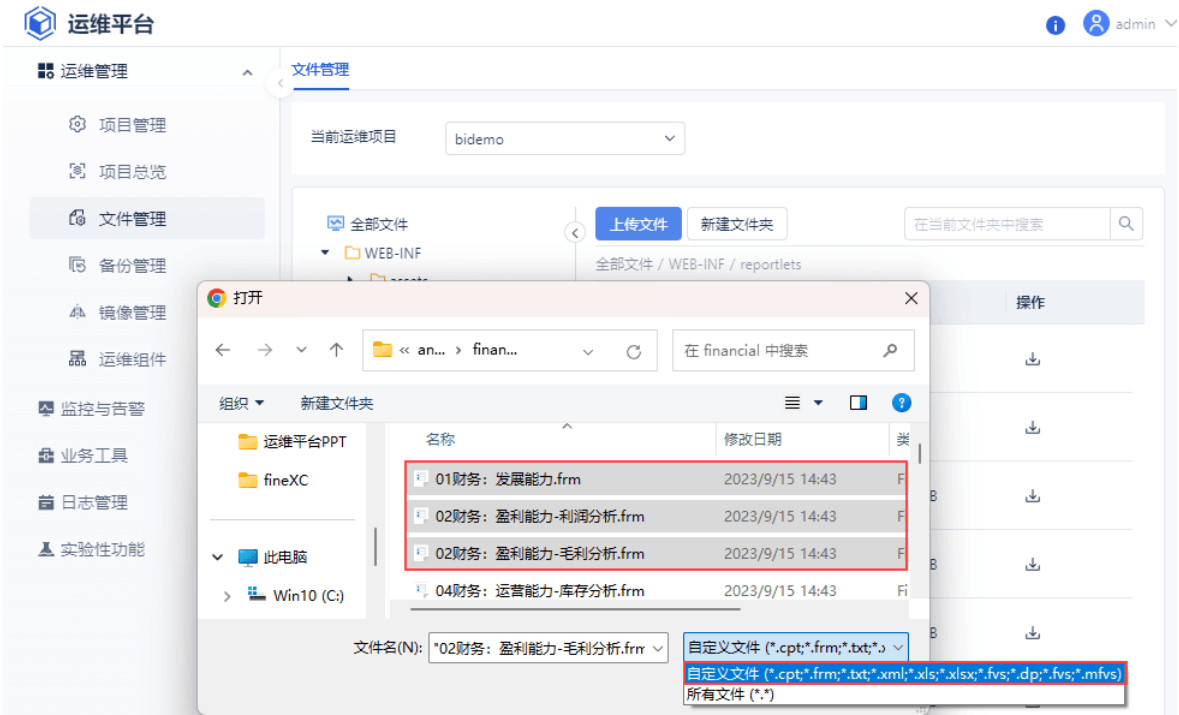
#### 3.2 文件管理优化

详情请参见：[文件管理](#)

##### 1) 支持新建文件夹



- 2) 支持批量上传文件
- 3) 支持上传.fvs 和.mfvs 格式文件



- 4) 上传同名文件时新增覆盖提示



### 3.3 备份管理逻辑优化

- 1) 备份前磁盘空间检测优化

为了防止备份导致磁盘爆满，提升备份前磁盘空间检查要求。

备份保存目录剩余可用空间必须 > （最近一次备份大小+自定义限制值）时，方可进行备份。

2) 升级/还原前备份逻辑优化

降低运维项目升级和还原前备份要求，不强制必须立刻备份，检测存在 24 小时内备份即可升级/还原

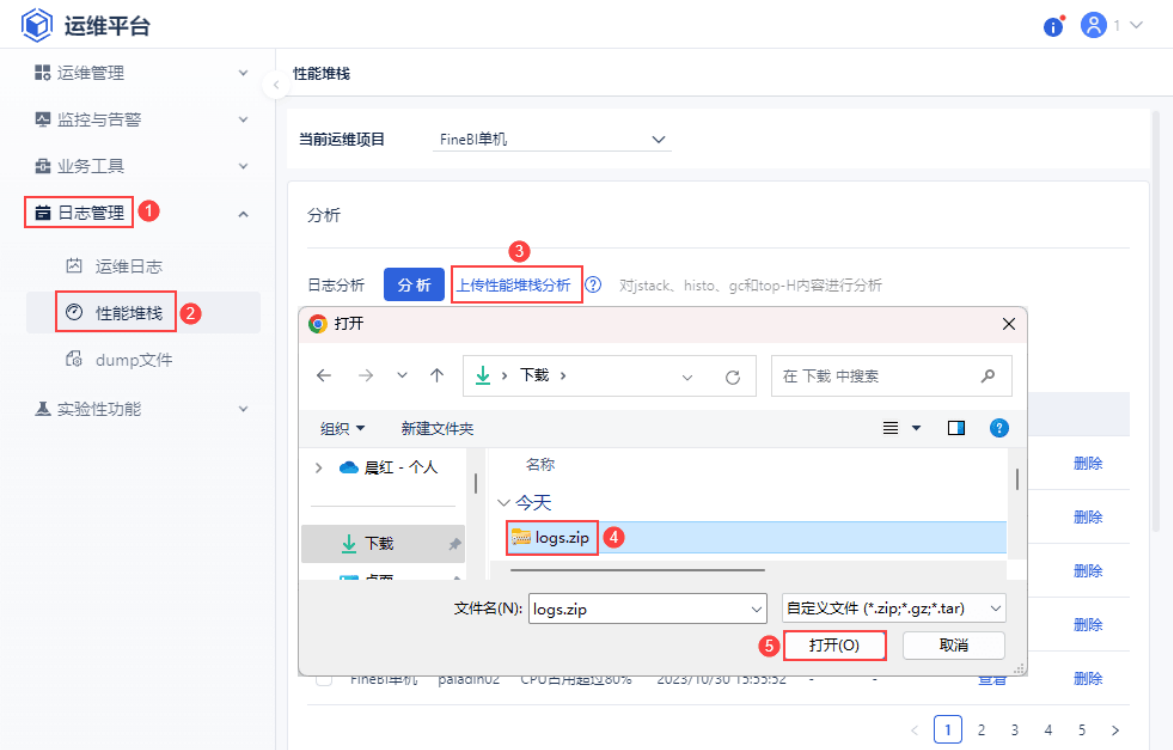
详情请参见：[备份管理](#)

4. 日志管理

4.1 支持上传本地性能堆栈进行分析

支持手动上传本地性能堆栈，至运维平台进行分析

详情请参见：[性能堆栈](#)



4.2 展示历史性能堆栈生成记录

支持展示历史性能堆栈生成记录

详情请参见：[性能堆栈](#)



4.3 性能堆栈使用顺序优化

性能堆栈使用顺序优化，变更为：分析>下载>生成

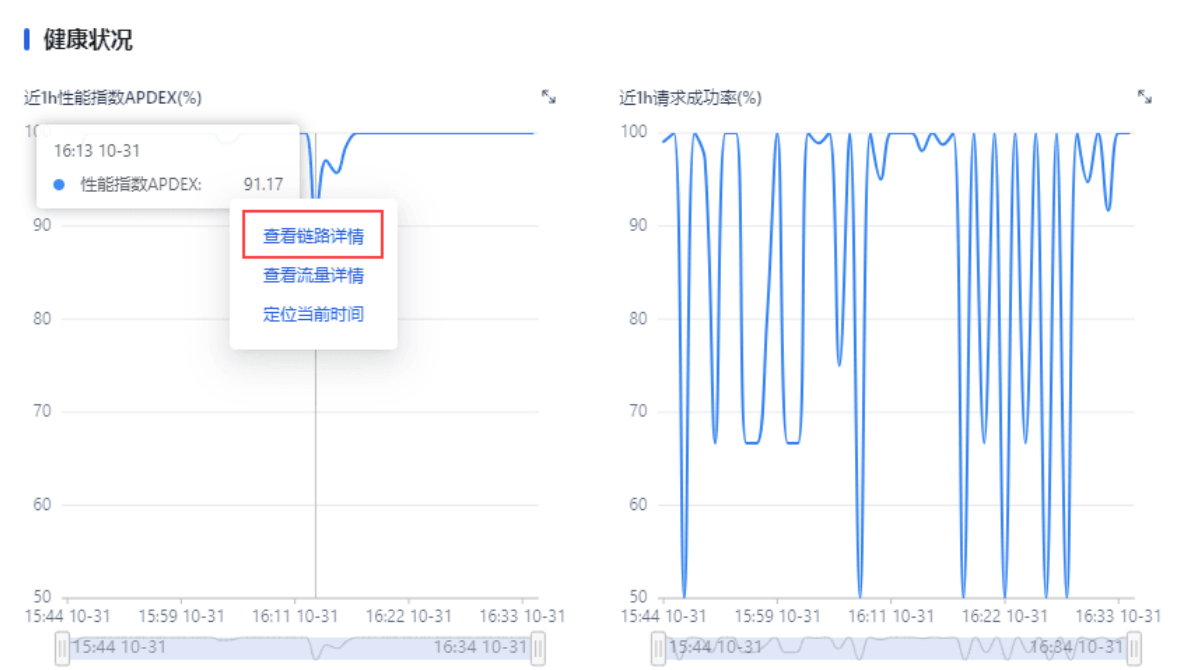
详情请参见：[性能堆栈](#)

5. 实验室功能

5.1 系统健康观测跳转链路追踪优化

为确保精确定位链路信息，系统健康观测联动跳转链路详情时，部分跳转项从「首屏链路」调整为「调用链路」

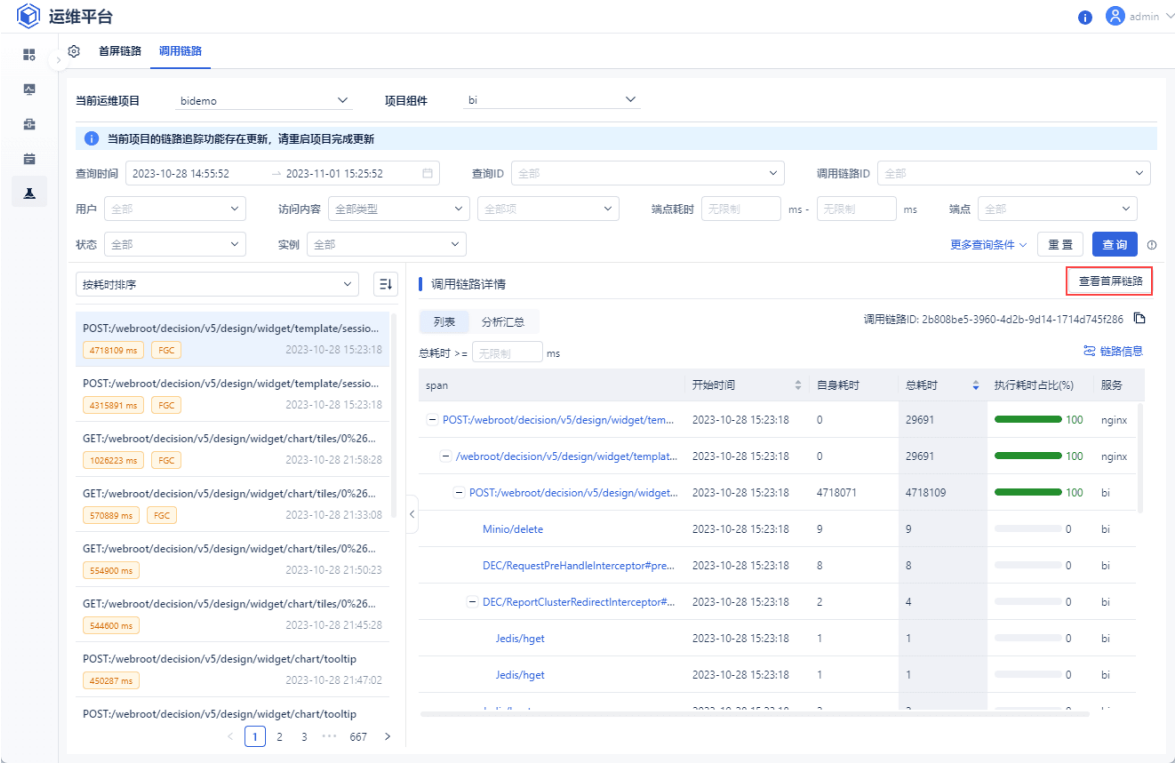
详情请参见：[系统健康观测](#)



5.2 调用链路支持调整首屏链路

若某个调用链路所在 session 有相关首屏链路数据，则支持联动跳转「查看首屏链路」

详情请参见：[调用链路](#)



## 1.2.5 V1.11.0 更新日志

### 1. 版本

运维平台版本	发布日期
V1.11.0	2023-10-17

### 2. 部署升级 FineOps

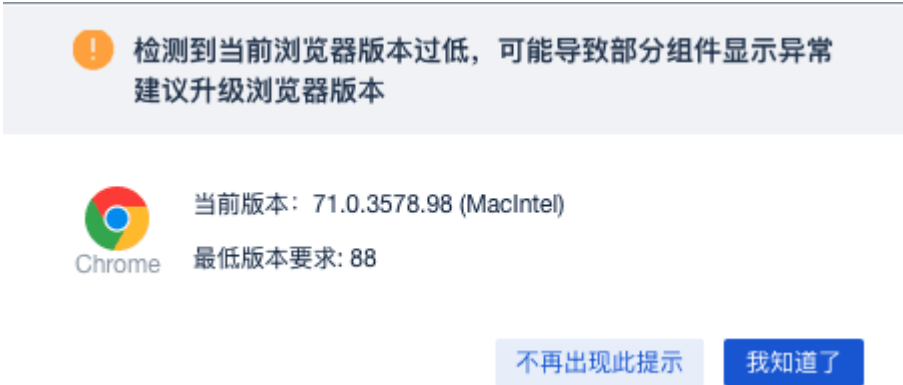
#### 2.1 新增版本更新强提醒

新增「版本更新」提醒，帮助用户及时享受更多功能和更优性能。



#### 2.2 新增浏览器版本校验

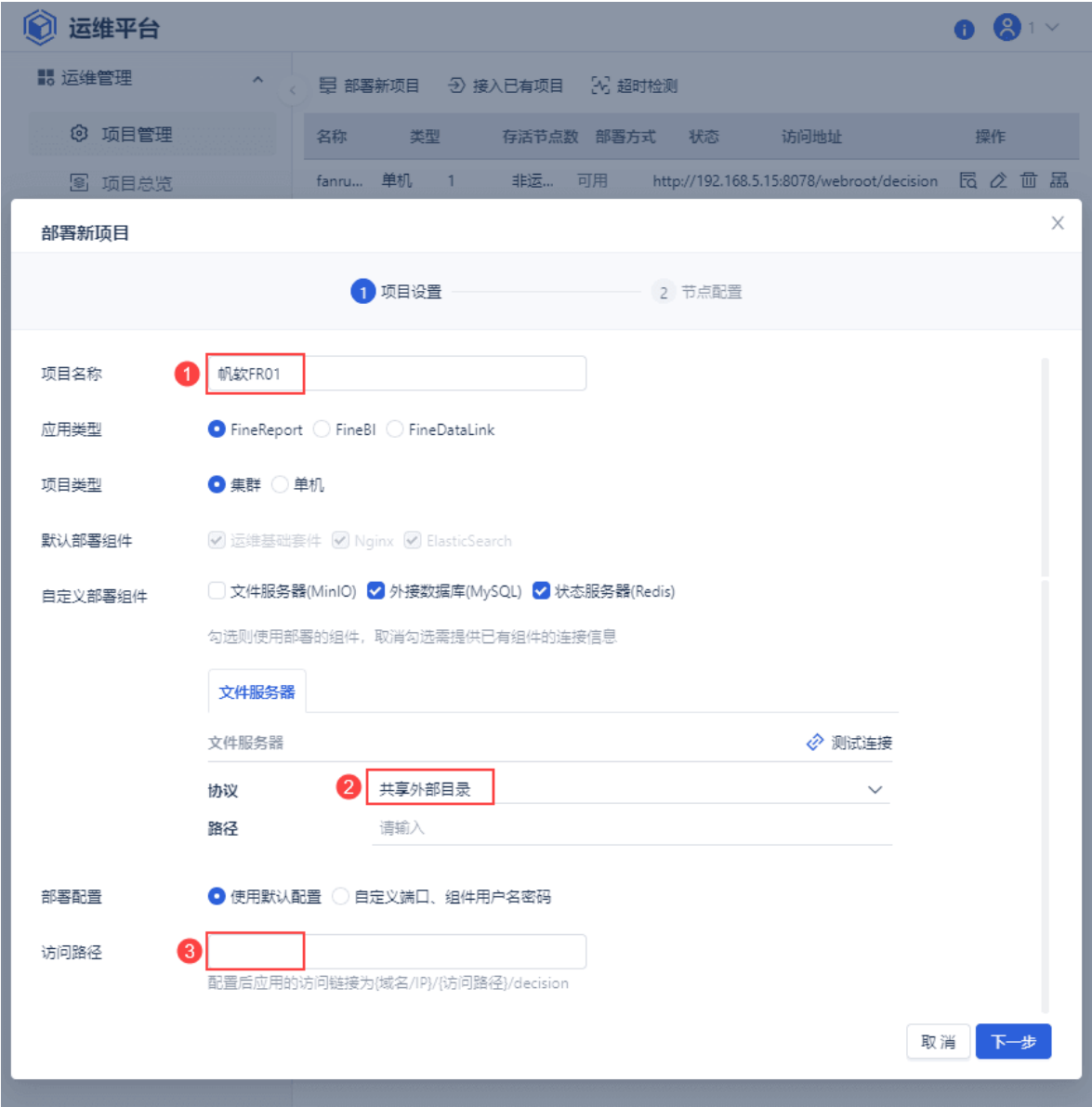
新增浏览器版本校验，避免使用低版本浏览器，导致运维平台界面异常。



### 3. 运维管理

#### 3.1 项目设置放宽

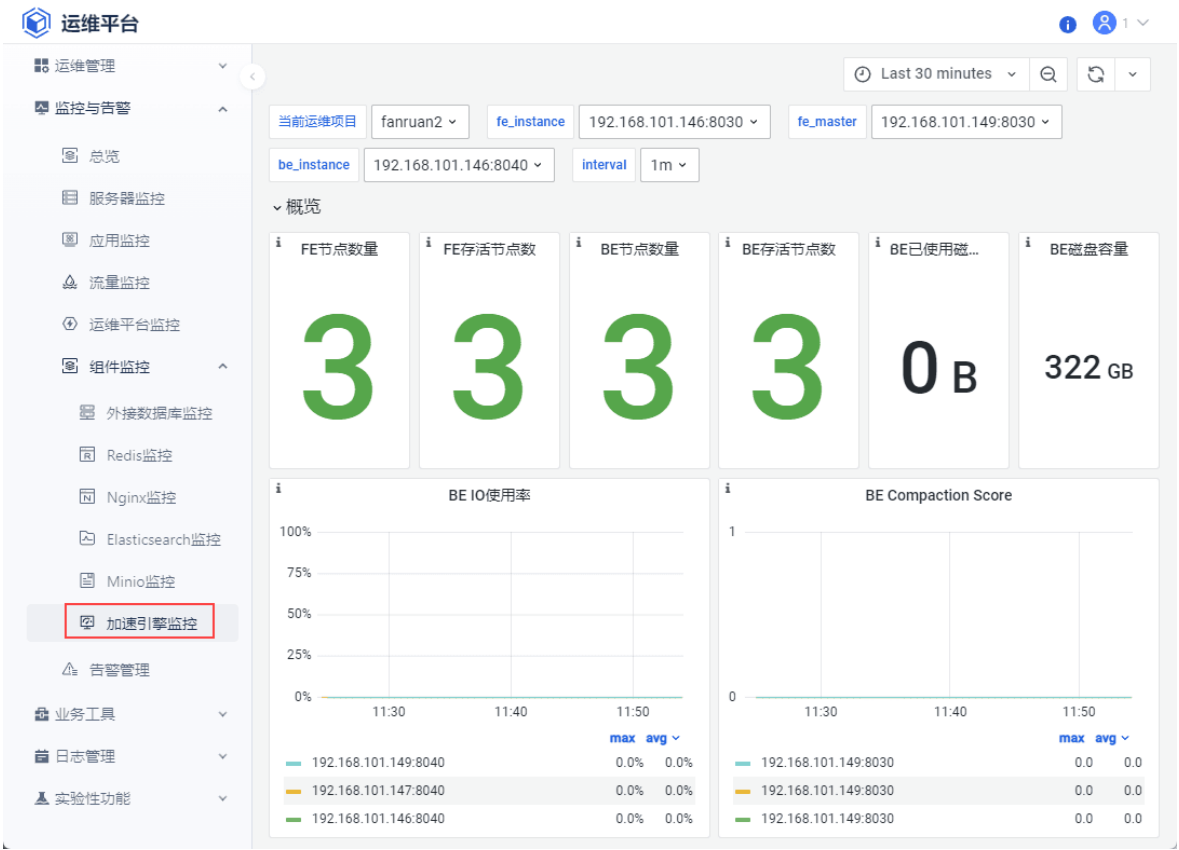
- 1) 部署/接入项目时，项目名称限制放宽，新增支持英文大写、中文。
- 2) 部署新项目时，访问路径支持为空，方便配置工程短链接。
- 3) 部署新集群项目时，支持使用外部共享目录作为文件服务器。



## 4. 监控与告警

### 4.1 新增加速引擎监控

组件监控新增「加速引擎监控」，帮助管理员监控管理 BI 大数据引擎组件。



## 4.2 新增运维平台挂载磁盘预警

新增默认告警「运维平台挂载磁盘预警」，覆盖通用运维场景。



## 5. 实验室功能

### 5.1 调用链路优化

详情请参见：[调用链路](#)

- 1) 调用链路详情新增「分析汇总」视图，帮助用户快速洞察耗时分布。

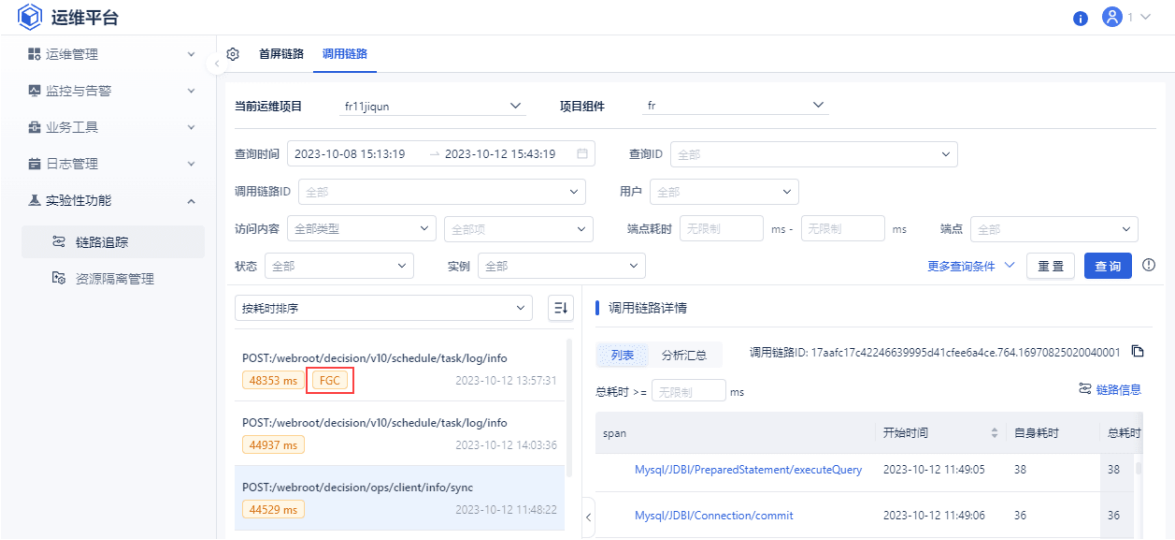




2) 调用链路详情新增「链路信息」按钮，汇总展示链路关键信息。



3) 调用链路列表新增「FGC」标识，标记链路时间范围内是否发生 FullGC。



## 1.2.6 V1.10.0 更新日志

### 1. 版本

运维平台版本	发布日期
V1.10.0	2023-09-28

### 2. 兼容说明

由于更换了云端镜像仓库，自 V1.9.2 以下版本在线升级至 V1.10.0 及以上版本时，需要升级两次。

即先从低版本升级至 V1.9.2 版本，再从 V1.9.2 版本升级至高版本。

### 3. 监控与告警

#### 3.1 新增三种告警通知方式

新增三种告警通知方式，满足用户多样化需求：企业微信机器人、钉钉机器人、飞书机器人

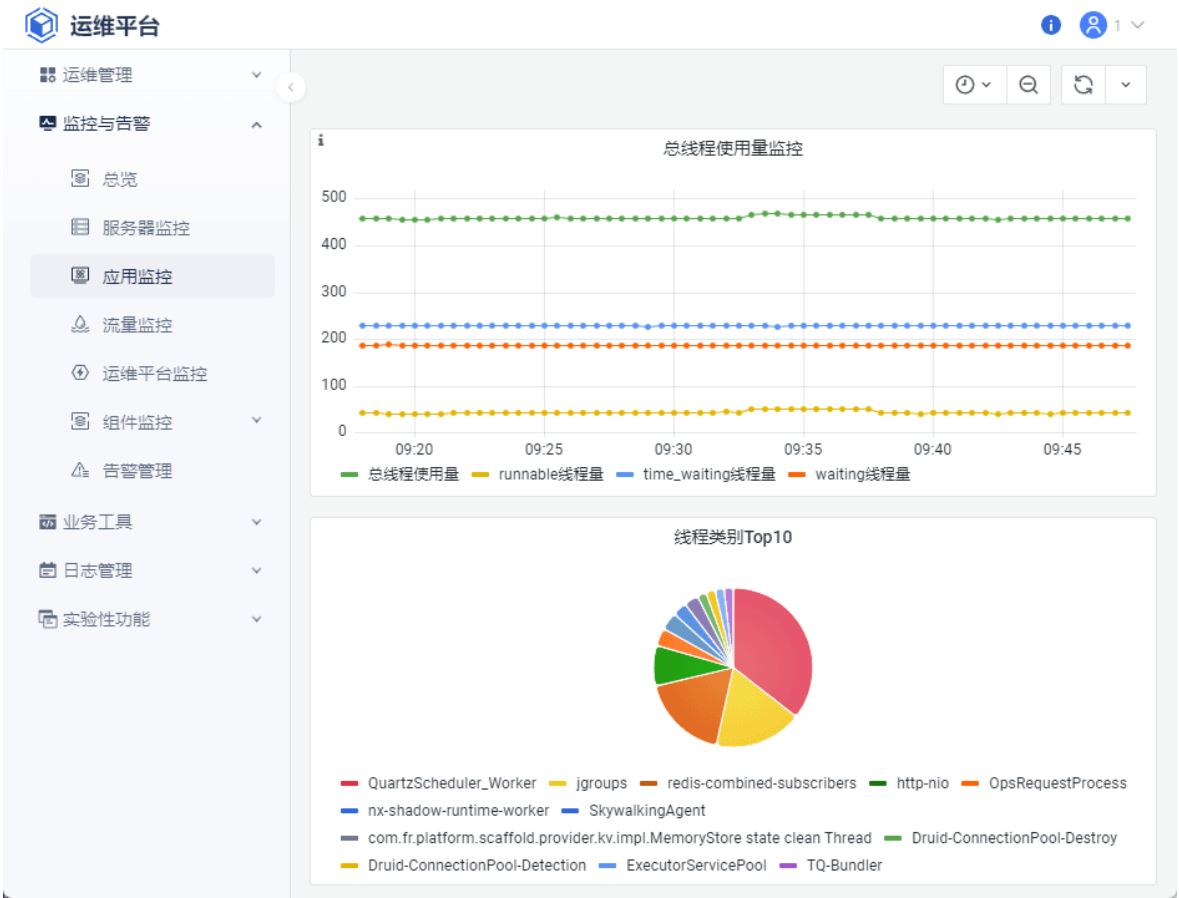
详情请参见：[告警方式](#)



#### 3.2 应用监控新增观察指标

应用监控新增观察指标：总线程使用量监控、线程类别 Top10，帮助用户排查线程泄露场景。

详情请参见：[应用监控](#)



3.3 系统健康观测交互优化

- 1) 新增全局控件：模块切换按钮、请求类型筛选按钮，交互更加合理
- 2) 系统健康观测界面分区优化，「节点情况明细表」更名为「节点概况表」

详情请参见：[系统健康观测](#)



4. 业务工具

4.1 集群支持健康巡检快速修复

集群运维项目，支持使用健康巡检快速修复功能。

运维平台

运维管理

监控与告警

业务工具

健康巡检

宕机自动处理

资源清理

会话管理

资源优先级管理

调度管理

业务配置更改

健康巡检

当前运维项目: bidemo

共发现异常项目 12 项 [如何修复异常?](#) [快速修复](#)

系统检查

共 3 项异常

序号	检查项	节点名称	当前配置	修改建议
0110	交换分区	de5a07df9132	48.0GB	建议关闭容器所在服务...
0110	交换分区	9856452d5e9f	48.0GB	建议关闭容器所在服务...
0110	交换分区	8d8b16d94a36	48.0GB	建议关闭容器所在服务...

系统运行状态: 无异常

# 1.2.7 V1.9.0 更新日志

## 1. 版本

运维平台版本	发布日期
V1.9.0	2023-09-07

## 2. 部署升级

### 2.1 升级运维平台新增检测 docker 引擎版本

由于 docker 与 FineOps 运维平台有匹配关系，因此升级 FineOps 时需要检测并升级 docker 引擎版本。

- [在线升级](#)：若检测到 docker 引擎版本不符合升级需求，则无法进行在线升级，请转用手动升级方案。
- [手动升级](#)：如需手动升级运维平台，必须确保运维平台所在服务器、运维平台对接的项目所在服务器，均支持升级 docker 引擎。



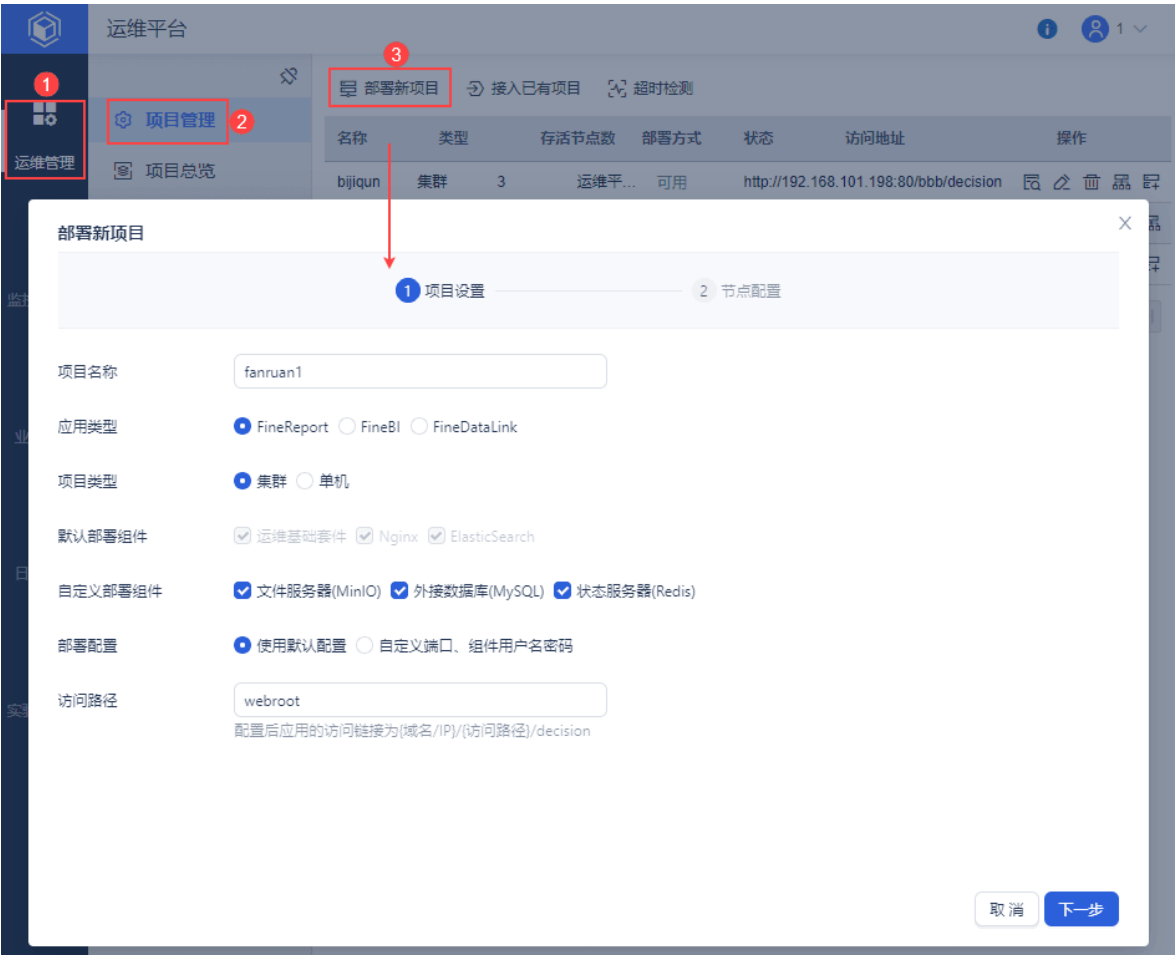
## 3. 运维管理

### 3.1 项目管理交互界面优化

为保持 FineOps 平台风格统一，部署新项目和接入已有项目交互界面优化，拆分默

认部署组件和自定义部署组件。

详情请参见：[部署新项目](#)

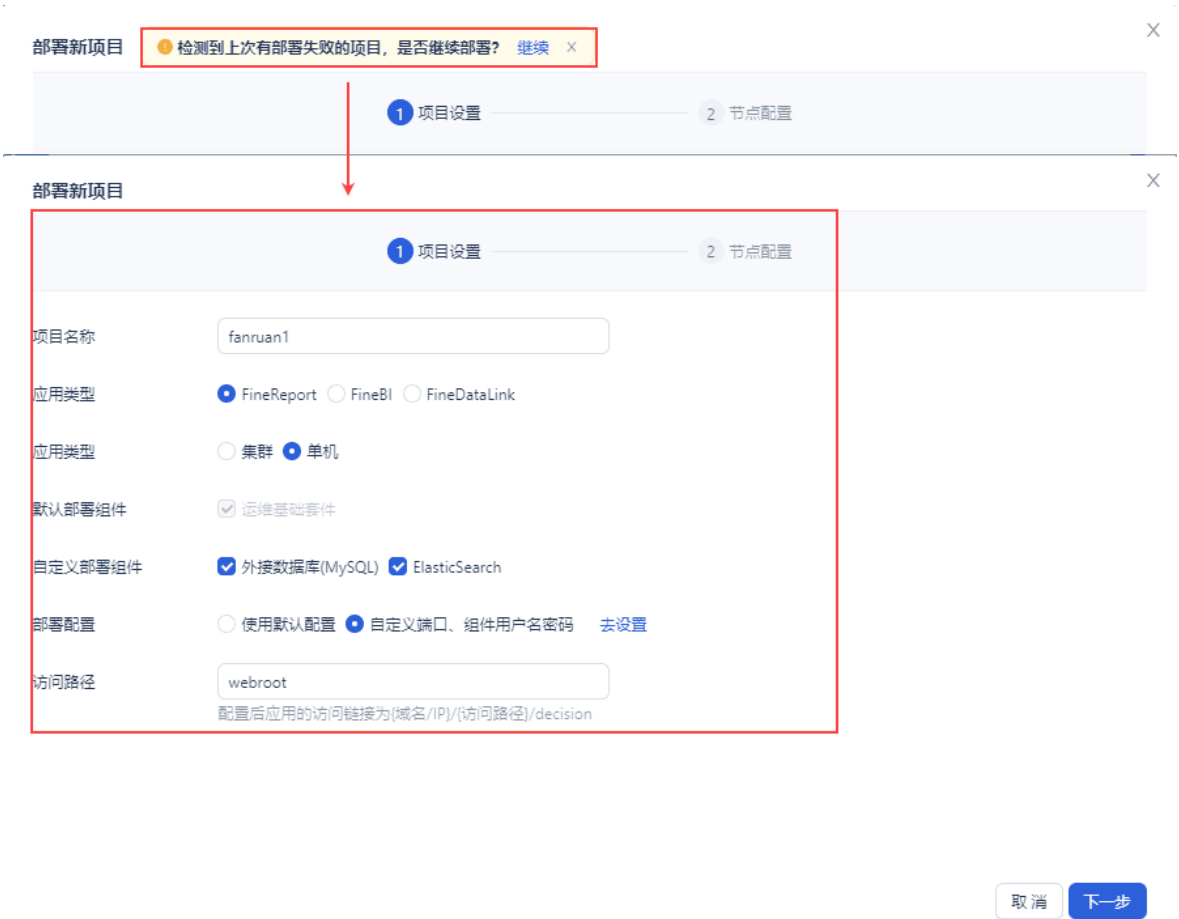


### 3.2 部署失败再次部署自动填充信息

若运维平台上一次部署项目失败，再次部署新项目时，会出现提示「检测到上次有部署失败的项目，是否继续部署？」

若点击「继续」按钮，则自动填充上次部署的项目设置和节点配置，节约用户时间。

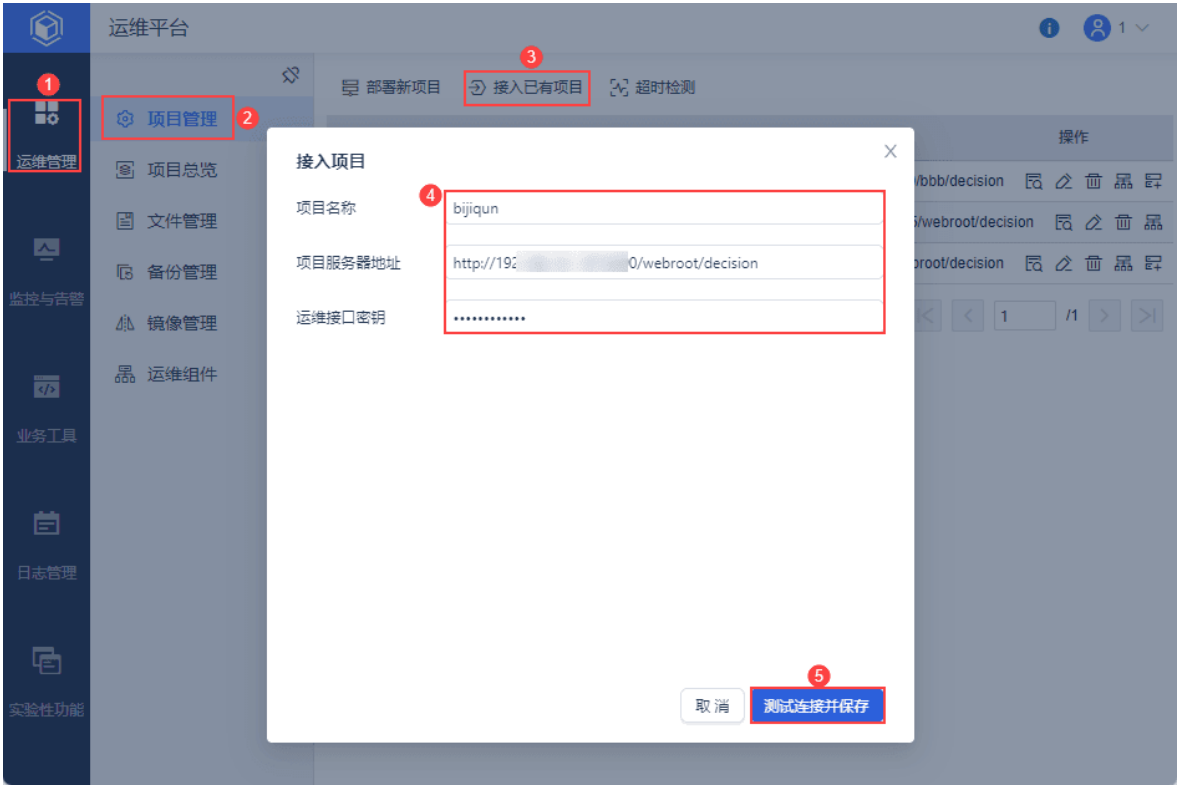
详情请参见：[运维平台项目异常问题排查](#)



### 3.3 接入已有容器化项目步骤精简

接入已有容器化项目步骤精简, 删除冗余操作步骤: 组件信息设置、服务器信息设置。

详情请参见: [接入已有容器化项目](#)



### 3.4 添加应用节点交互优化

支持从组件管理处添加应用节点, 集群项目扩容操作步骤更流畅。

详情请参见：[集群运维项目扩容](#)



## 4. 监控与告警

### 4.1 告警名称优化

告警管理中，内置的默认告警「负载过高告警——节点无响应」更名为「应用卡顿告警——节点无响应」，降低用户理解成本。

详情请参见：[告警任务](#)





## 5. 业务工具

### 5.1 健康巡检部分检查项支持快速修复

为降低用户优化工程的难度，健康巡检部分检查项支持快速修复：

#### 1) 环境检查部分检查项支持快速修复

- 0705 gblic 内存泄露相关配置

#### 2) Java 虚拟机检查部分检查项支持快速修复

- 0807 headless 模式
- 0809 BytecodeRecompilationCutoff 配置
- 0810 PerMethodRecompilationCutoff 配置
- 0813 codecache 配置
- 0816 zip 相关配置-Dsun.zip.disableMemoryMapping

#### 3) Java 虚拟机检查新增检测项：

- 0823 元数据空间配置

详情请参见：[健康巡检内容](#)

# 1.2.8 V1.8.0 更新日志

## 1. 版本

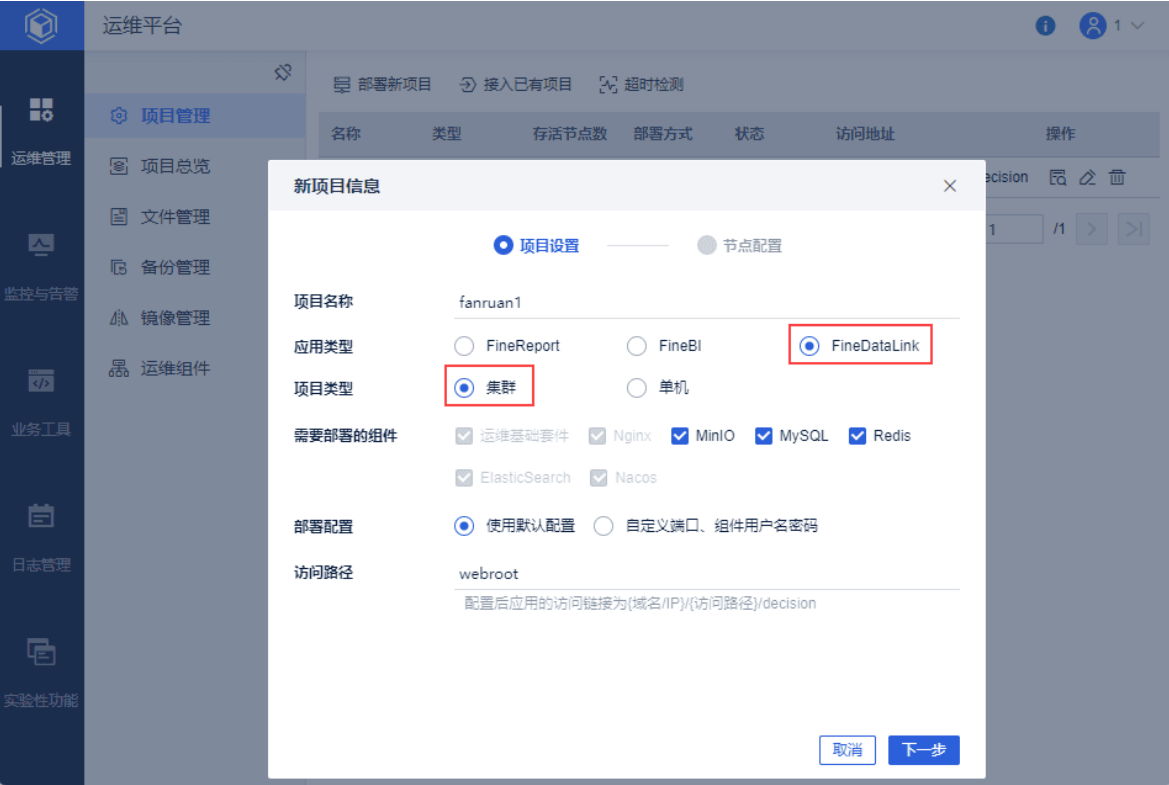
运维平台版本	发布日期
V1.8.0	2023-08-23

## 2. 运维管理

### 2.1 支持部署新 FineDataLink 集群项目

FineDataLink 集群具有高可用性、高性能、易于管理、可伸缩性和安全保障等优点  
运维平台支持容器化部署全新的 FineDataLink 集群项目。

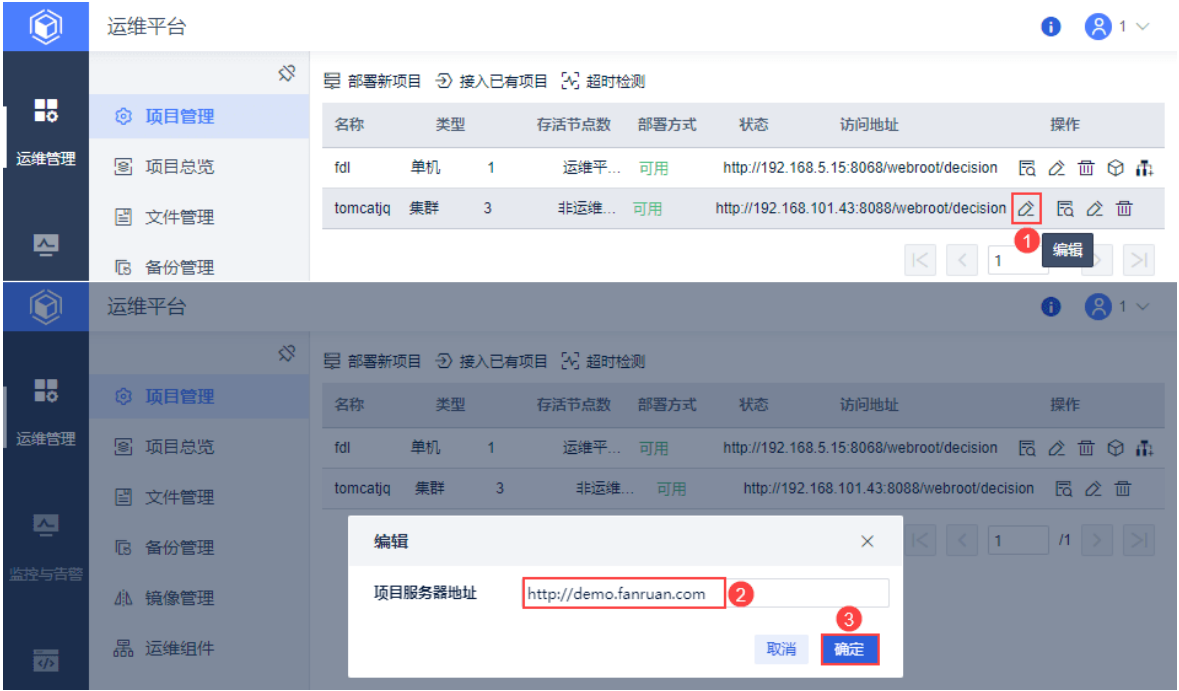
详情请参见：[部署新 FineDataLink 集群项目](#)



### 2.2 支持编辑项目访问地址

运维平台显示的默认访问地址是根据接入时的 IP 生成的。而管理员可能为工程配置了域名，配置了内外网 IP，会导致自动生成的地址无效，无法访问。  
此时管理员可将自动生成的访问地址修改为实际项目访问地址。

详情请参见：[项目管理](#)



### 2.3 项目接入入口优化

部署新项目、接入已有项目入口位置调整，更醒目。

详情请参见：[项目管理](#)



### 2.4 新增项目对比功能

大部分公司，在实际应用过程，往往会有多套工程环境：开发环境、测试环境、生产环境。

如果为了确保开发和测试环境制作的模板，可以成功上线生产环境，必须确保这些工程的 JAR 包和插件版本完全一致。

运维平台新增项目对比功能，支持展示两个工程中 JAR 包和插件的不一致清单。

详情请参见：[项目总览](#)



对比结果

当前项目 tomcatjq 对比项目 fdl

JAR包对比

当前项目	对比项目
<ul style="list-style-type: none"><li>core-11.0.17-Build#release-2023.06.27.17.29.47.490</li><li>datasource-11.0.17-Build#release-2023.06.27.17.30.18.48</li><li>decision-11.0.17-Build#release-2023.06.27.17.34.17.184</li><li>report-11.0.17-Build#release-2023.06.27.17.55.07.367</li></ul>	<ul style="list-style-type: none"><li>core-11.0.17-Build#release-2023.06.27.17.29.47.490</li><li>datasource-11.0.17-Build#release-2023.06.27.17.30.18.48</li><li>decision-11.0.17-Build#release-2023.06.27.17.34.17.184</li><li>report-11.0.17-Build#release-2023.06.27.17.55.07.367</li></ul>

插件对比

「当前项目」比「对比项目」多出的插件

- fine\_conf\_entity可视化配置插件-1.9.18-true
- 扩展图表-4.5.7-true
- HTML5移动端展现-11.0.68-true

「当前项目」比「对比项目」缺少的插件

- minio资源仓库-1.1.3-true
- logdb的ElasticSearch实现-1.0.0-true
- 消息队列rocketmq-1.1-true

「当前项目」与「对比项目」版本/启停状态不同的插件

- 共享外部目录-1.4-false<=>共享外部目录-1.4-true
- 模板助手-1.2.18-true<=>模板助手-1.2.17-true
- 资源调度熔断插件-1.3.6-true<=>资源调度熔断插件-1.3.2-true

取消 确定

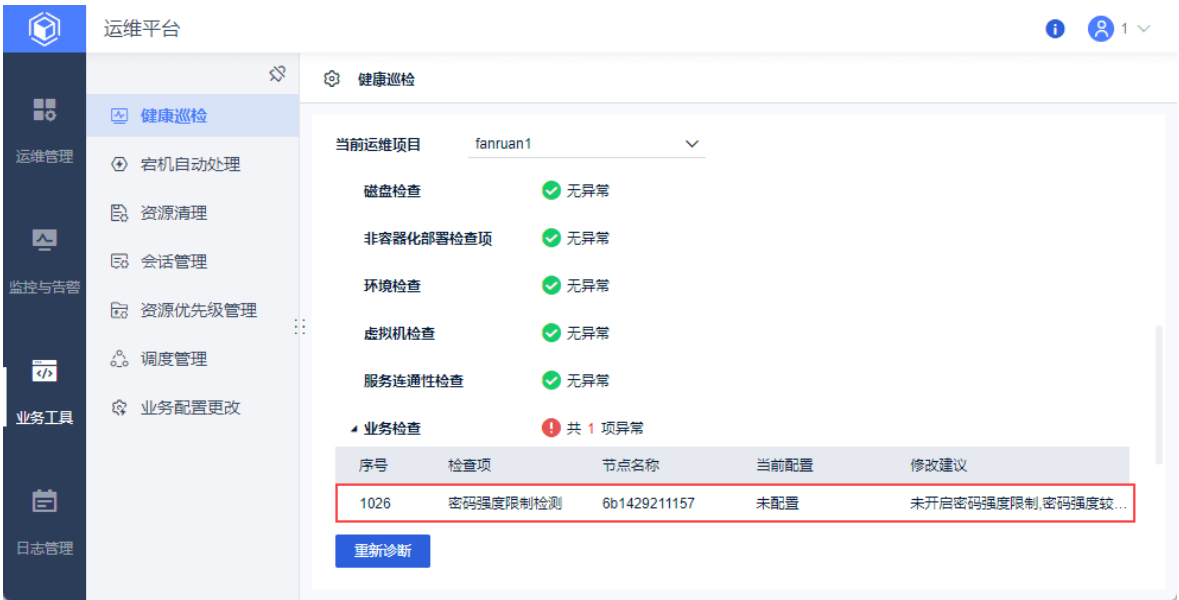
3. 业务工具

3.1 健康巡检新增检测项

健康巡检「业务检查」模块新增检测项 1026 密码强度限制检测

未开启密码强度限制，密码强度较低可能容易被攻击，导致安全问题，建议在[管理系统]-[系统管理]-[登录]中打开密码强度限制

详情请参见：[健康巡检内容](#)



4. 日志管理

4.1 性能堆栈日志分析优化

- 1) 完善自动触发堆栈日志的场景
- 2) 分析记录列表新增展示「触发类型」

详情请参见：[性能堆栈](#)



5. 实验室功能

5.1 调用链路新增查询筛选项

查询界面新增筛选项：端点、状态、实例

详情请参见：[调用链路](#)

运维平台

运维管理

监控与告警

业务工具

日志管理

实验性功能

链路追踪

资源隔离管理

首屏链路

调用链路

当前运维项目

fanruan1

项目组件

fr

查询时间

2023-08-23 19:09:27 → 2023-08-23 19:39:27

查询ID

全部

调用链路ID

全部

用户

全部

访问内容

全部类型

全部项

端点耗时

无限制 ms - 无限制 ms

端点

GET/w...

状态

成功

实例

192.168.101.46:8080

更多查询条件

重置

查询

按耗时排序

GET/webroot/decision/ops/client/cluster/url

121 ms

2023-08-23 19:19:36

GET/webroot/decision/ops/client/cluster/url

44 ms

2023-08-23 19:11:36

GET/webroot/decision/ops/client/cluster/url

1

2

调用链路详情

总耗时 >= 无限制 ms

traceID: 8de6964d1ca24527afc7b54730ed00d8.7691.16927895763130057

span

开始时间

GET/webroot/decision/ops/client/cluster/url

2023-08-23

MySQL/JDBI/Connection/commit

2023-08-23

## 1.2.9 V1.7.0 更新日志

### 1. 版本

运维平台版本	发布日期
V1.7.0	2023-08-07

### 2. 监控与告警

#### 2.1 支持自定义告警

当运维项目遇到高负载、宕机风险、磁盘空间不足等问题时，需要及时通知管理员。

FineOps 新增支持自定义异常触发条件和达成告警的持续时间。

详情请参见：[告警任务](#)



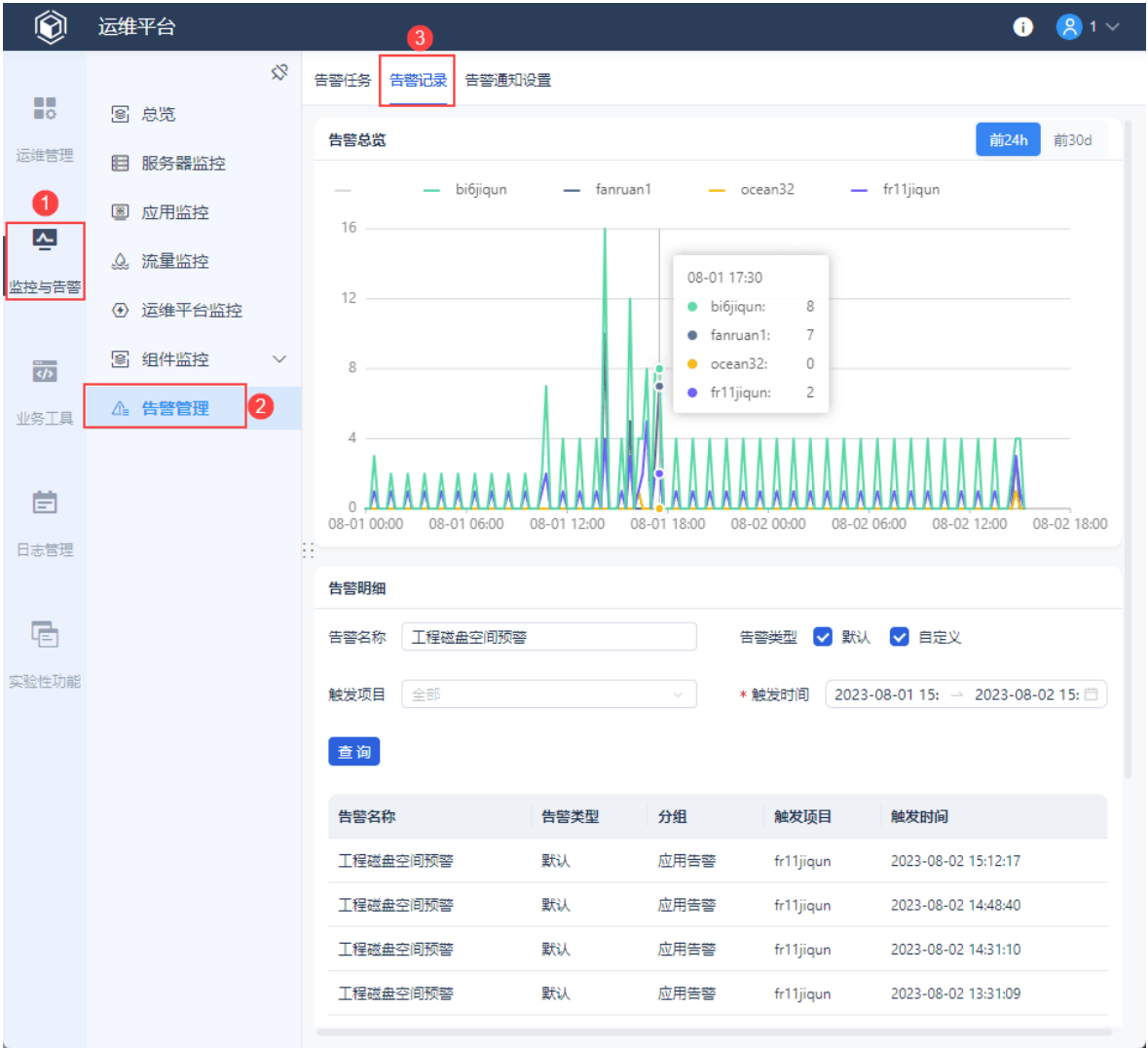
#### 2.2 新增告警记录查看

告警记录是指在告警管理中，针对运维项目出现异常情况时所触发的告警任务的记录。

通过查看告警记录，管理员可以获取关键指标和详细信息，如告警发生时间、告警类型、触发项目等。

管理员可以快速识别常见问题、确定趋势和模式，并采取相应的措施来解决或预防潜在的故障或风险。

详情请参见：[告警记录](#)



### 2.3 告警管理界面重构

告警管理界面和交互展示完全重构，拆分为：告警任务、告警记录、告警通知设置三个模块。



## 3. 实验室功能

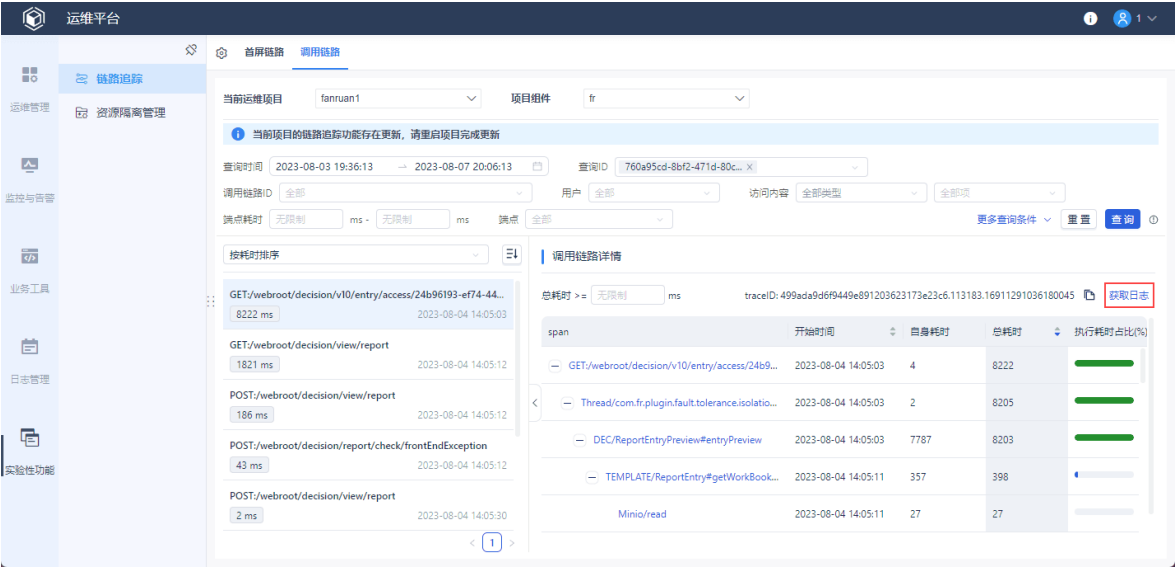
### 3.1 支持获取链路日志到本地

对于使用 elasticsearch 套件获取和存储日志的运维项目，在查看链路详情时，支持



获取日志到本地。

详情请参见：[调用链路](#)



# 1.2.10 V1.6.0 更新日志

## 1. 版本

运维平台版本	发布日期
V1.6.0	2023-07-21

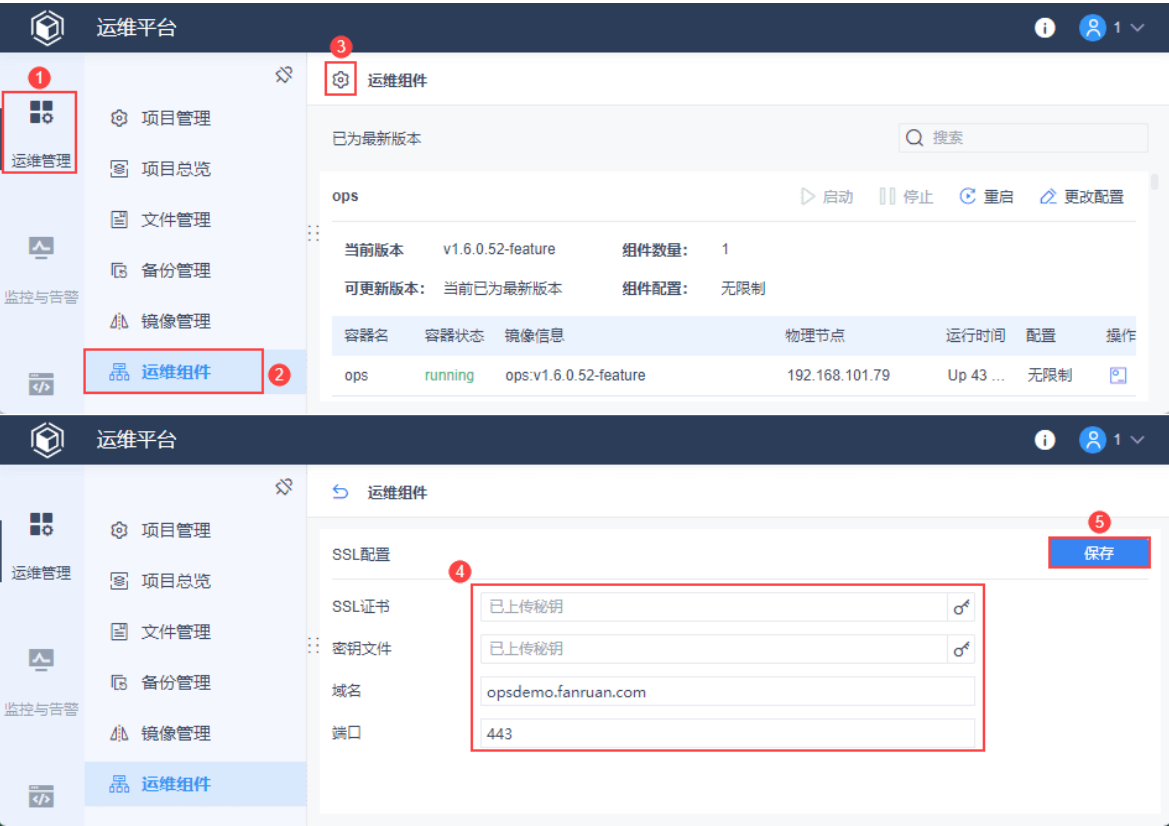
## 2. 运维管理

### 2.1 支持界面化为运维平台开启 SSL

要启用 HTTPS 协议，我们需要使用有效的 SSL 证书。SSL 证书中包含身份验证信息，帮助保障用户的通信加密和安全。

用户可通过简单的前端界面操作，为运维平台的 Nginx 配置 SSL 证书，以满足使用 HTTPS 访问的需求。

详情请参见：[为运维平台开启 SSL](#)

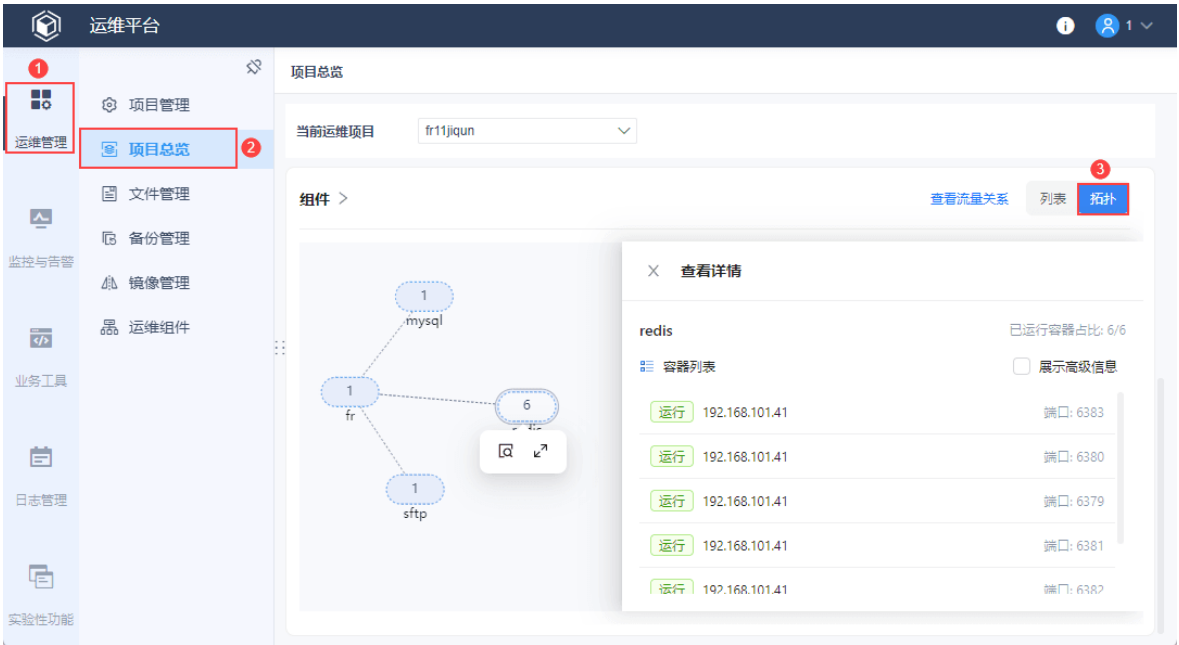


### 2.2 新增组件拓扑视图

以拓扑形式展示项目内各服务、组件间的网络结构及联通性，主要应用于集群项目：

- 帮助管理员查看每个节点和集群组件是否连接联通。
- 帮助管理员检查工程节点数是否与计划数量一致。
- 帮助管理员查看每个组件占用的服务器和端口是否与计划一致。

详情请参见：[项目总览](#)

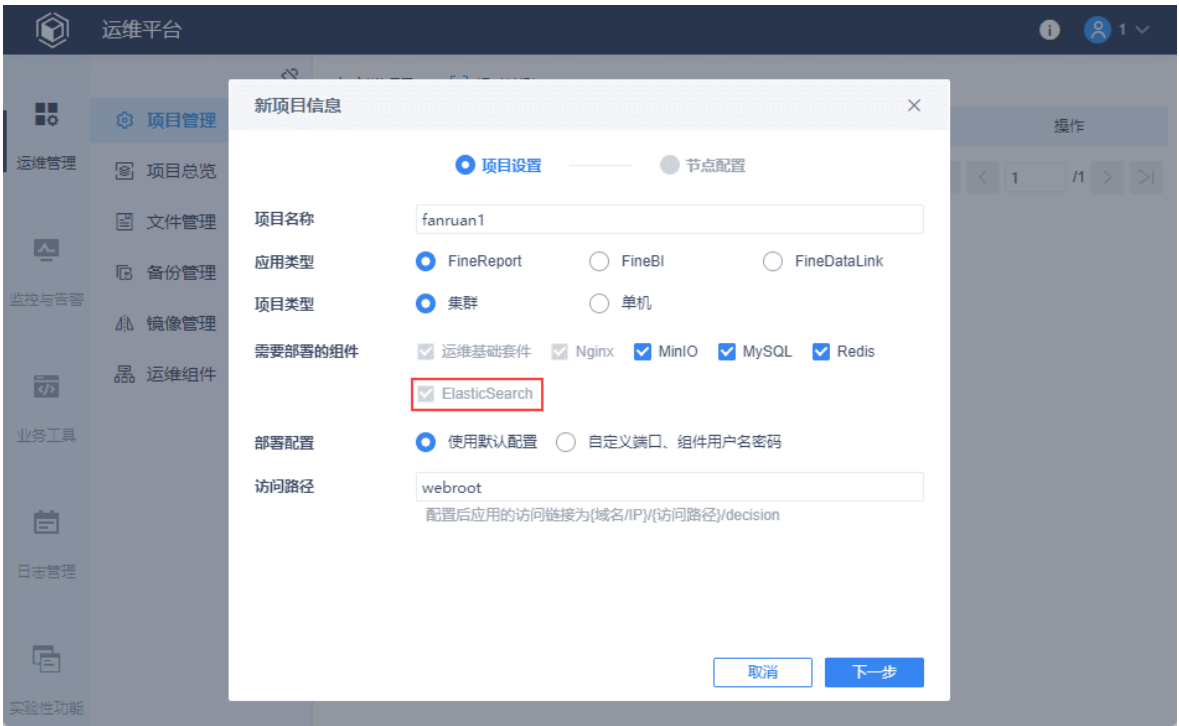


### 2.3 部署集群项目时自动安装日志组件

为了确保集群日志存储的稳定性：

- 1) 容器化部署集群项目时，elasticsearch 套件不再可选安装，而是必须安装，elasticsearch 套件将替换 FineReport/FineBI 中原有的 swift，作为日志存储，对项目中 FineReport、外接数据库、业务库和其他外置组件的请求进行采集。
- 2) 若选择安装 elasticsearch，同步在每一个工程节点安装 filebeat 组件，该组件主要用于日志收集。

详情参见：[部署新项目-FineReport](#)



### 2.4 项目管理展示信息优化

项目管理列表中，部署方式措辞优化，展示为：非运维平台部署/运维平台部署

详情请参见：[项目管理](#)



### 3. 监控与告警

#### 3.1 新增流量监控

对于集群工程，管理员希望了解各个工程节点和组件之间的请求流量情况，以便排查性能问题、联通性问题、节点平衡等。

FineOps 运维平台提供「流量监控」功能，帮助管理员查看不同时间段内各个节点间的请求流量情况。

详情请参见：[流量监控](#)



### 4. 业务工具

#### 4.1 新增堆栈触发类型记录

内存堆栈导出记录，新增「触发类型」，记录自动导出堆栈的触发原因，方便后续分析。

详情请参见：[宕机自动处理](#)

运维平台

运维管理

监控与告警

业务工具

健康巡检

宕机自动处理

资源清理

会话管理

资源优先级管理

调度管理

业务配置更改

宕机自动处理

宕机处理策略

内存堆栈导出记录

项目	导出内容	导出开始时间	导出时长	触发类型	是否成功	失败原因
tomcatjq	jstack	2023-07-19 17:37:43	0s	CPU占用超过80%	是	null
tomcatjq	jstack	2023-07-19 17:37:40	0s	CPU占用超过80%	是	null
tomcatjq	jstack	2023-07-19 17:37:36	0s	CPU占用超过80%	是	null
tomcatjq	jstack	2023-07-19 09:56:24	0s	CPU占用超过80%	是	null
tomcatjq	jstack	2023-07-19 09:56:21	0s	CPU占用超过80%	是	null
tomcatjq	jstack	2023-07-19 09:56:17	0s	CPU占用超过80%	是	null
tomcatjq	jstack	2023-07-18 19:56:38	0s	CPU占用超过80%	是	null

## 1.2.11 V1.5.9 更新日志

### 1. 版本

运维平台版本	更新日期
V1.5.9	2023-07-06

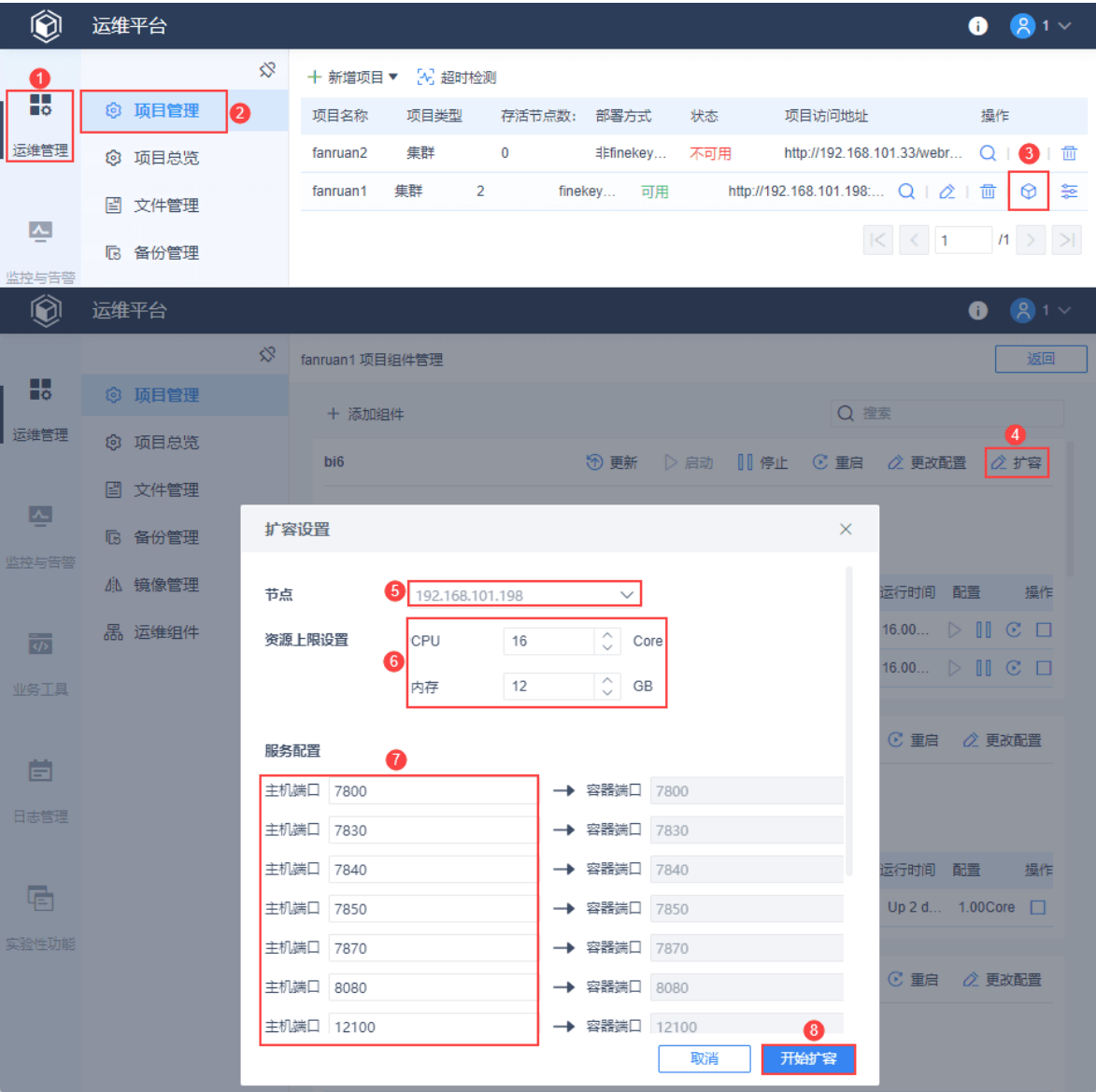
### 2. 项目管理

#### 2.1 提供集群扩充节点方案

随着业务数据量和用户访问量的增加，原本部署的集群节点无法满足需求，需要通过添加集群节点，提供更高的性能和可扩展性。

运维平台提供「添加节点」和「工程扩容」功能，支持为容器化集群运维项目，前端界面化扩充工程节点。

详情请参见：[集群运维项目扩容](#)



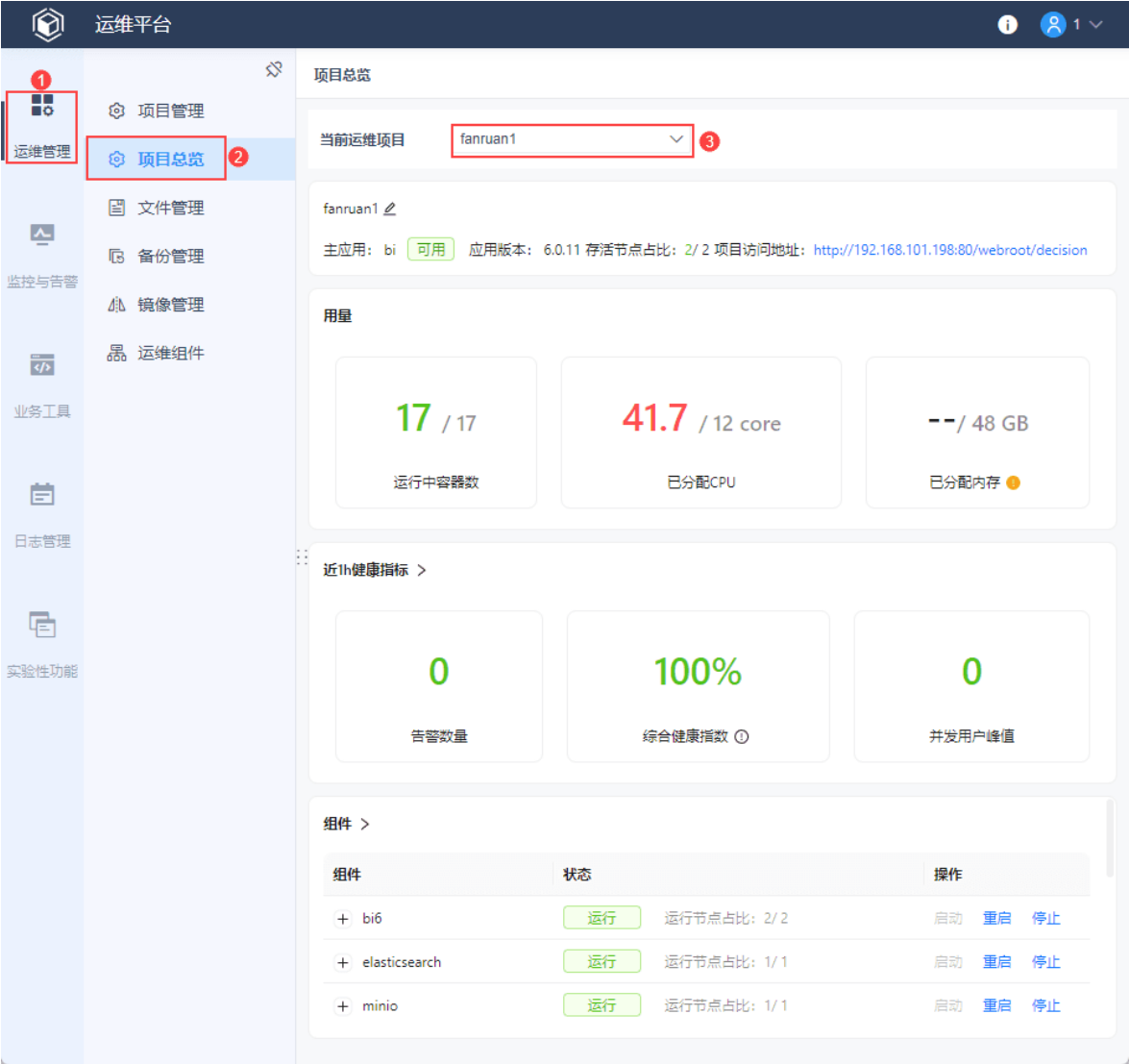
### 3. 运维管理

#### 3.1 新增项目总览界面

对于接入的运维项目，FineOps 运维平台为管理员提供多角度监控和各类运维操作。

运维平台提炼出用户最关心的信息和最常用的操作，组合成了「项目总览」界面。

详情请参见：[项目总览](#)



#### 4. 业务工具

##### 4.1 健康巡检检测内容优化

1) 磁盘检查新增检测项：

- 0509 顺序读取(非缓冲池)
- 0510 顺序写入(非缓冲池)

2) 磁盘检查删除检测项：

- 0503 随机读

3) 磁盘读写速度若低于 100，则不显示具体数值，仅提示「速率小于 100MB/S」

详情请参见：[健康巡检内容](#)

5. 监控与告警

5.1 系统资源监控优化

1) 功能更名

「系统运行状态」更名为「系统资源监控」

「当前系统状态」更名为「基础资源监控」

2) 展示内容优化

精选新增高频关注指标，剔除部分冗余指标。

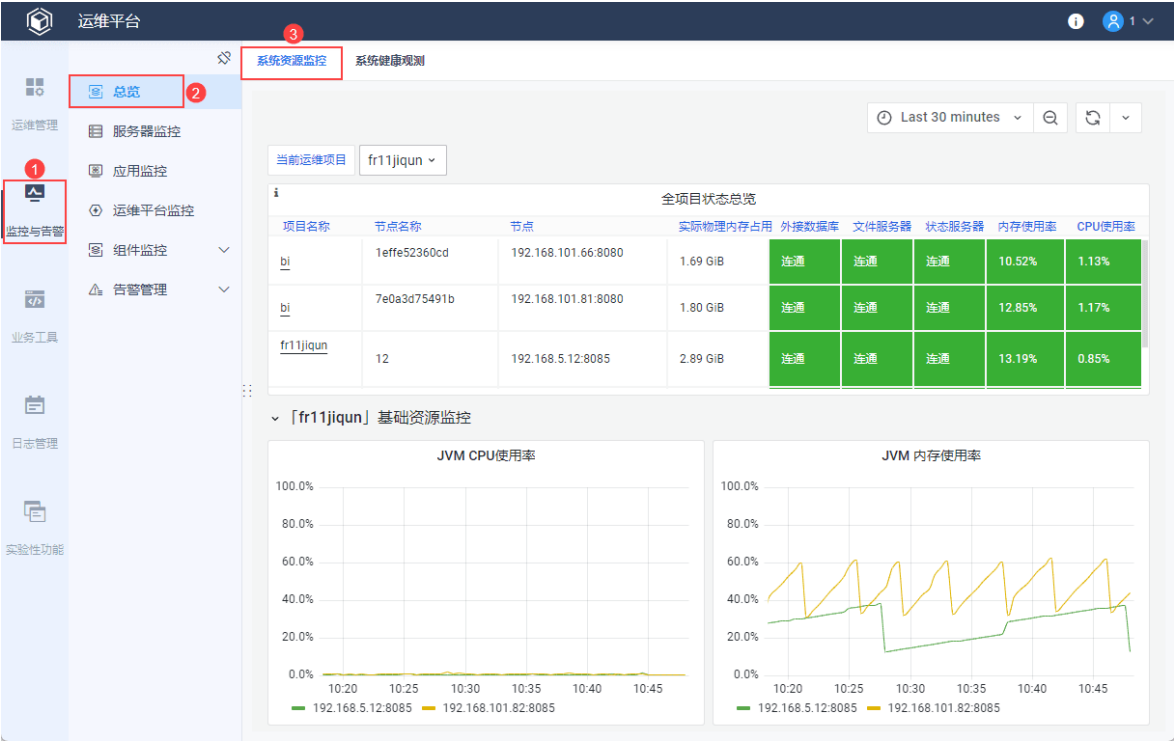
仪表盘重新划分为：全项目状态总览、基础资源监控、线程池监控。

3) 联动交互优化

顶部运维项目筛选按钮联动，基础资源监控和线程池监控中，默认仅展示所选项目的信息。

部分组件支持联动跳转「日志管理>性能监控>性能堆栈」，进行必要的堆栈分析。

详情请参见：[系统资源监控](#)



5.2 系统健康观测优化

1) 新增「健康巡检」强提醒按钮

2) 各指标图和明细表标题联动显示时间点，快速定位时间

详情请参见：[系统健康观测](#)





5.3 服务器监控优化

「服务器明细」中新增指标「TCP 排队数据量」

详情请参见：[服务器监控](#)



5.4 应用监控优化

- 1) 节点状态总览中：删除指标「线程死锁数」、「temp 目录磁盘余量」
- 2) 应用信息中：新增指标「系统资源优先级监控」
- 3) 删除「BI 资源监测」相关内容

详情请参见：[应用监控](#)

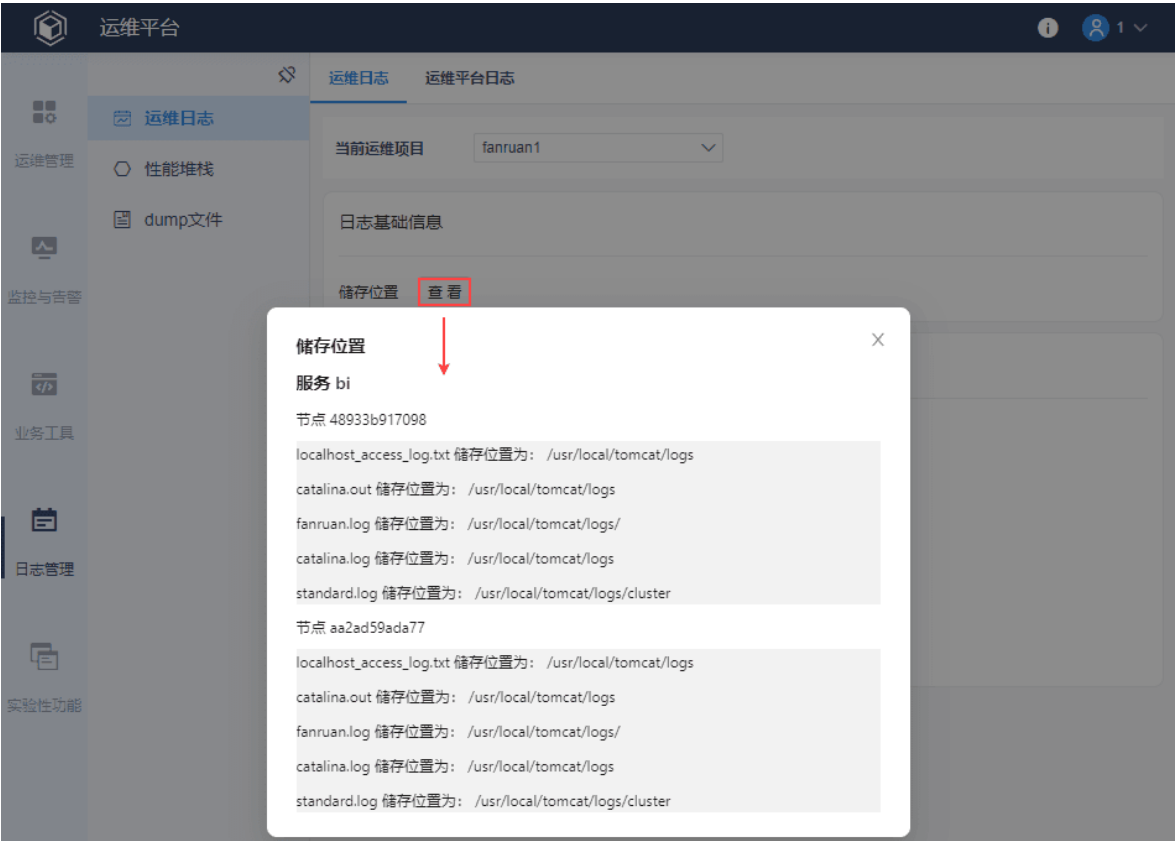
6. 日志管理

6.1 支持查看各节点运维日志位置

当用户遇到需要使用日志的场景时，部分公司对机房管控较严格，申请进机房流程繁琐，日志获取费时费力。

FineOps 运维平台，支持查看运维日志存储位置

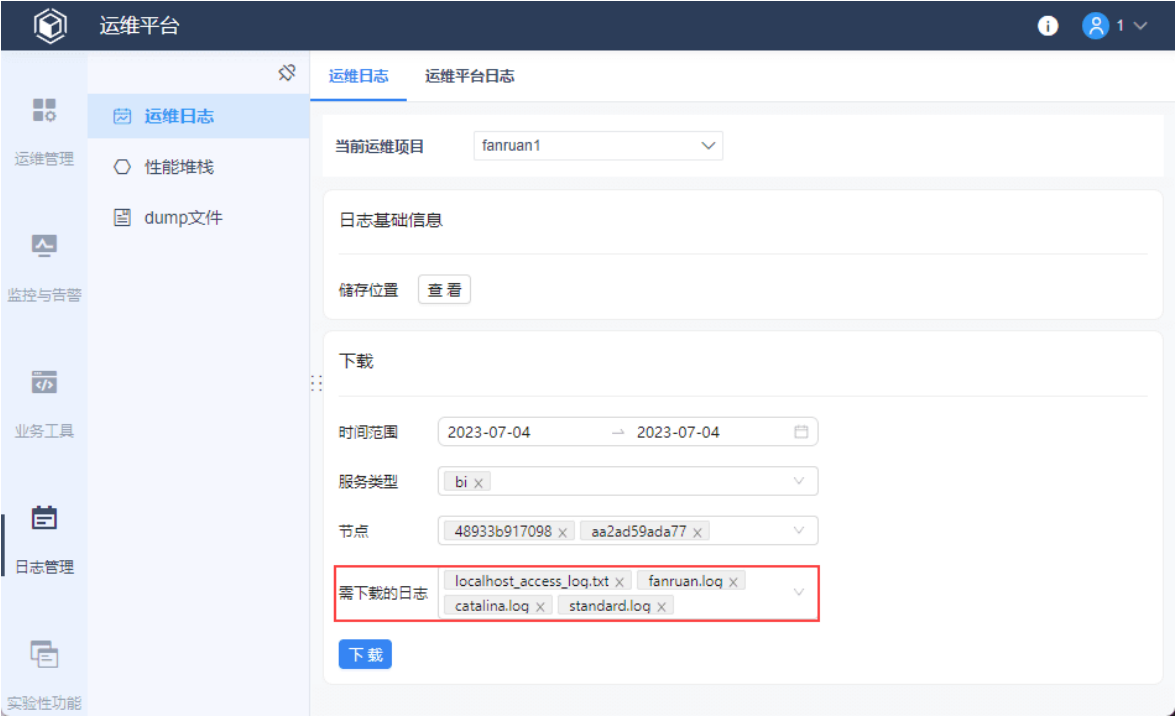
详情请参见：[运维日志](#)



## 6.2 支持选择需下载的运维日志

运维日志多种多样，有时用户只需要下载指定类型的日志，即可用于问题定位。

为了节约用户下载耗时，支持用户自定义选择需下载的日志类型



# 1.2.12 V1.5.7 更新日志

## 1. 版本

运维平台版本	更新日期
V1.5.7	2023-06-14

## 2. 部署升级

### 2.1 运维组件默认启用

为了降低用户使用运维平台相关功能的难度，运维组件安装后全部启动。

全新安装 V1.5.7 及之后版本的 FineOps 运维平台，elasticsearch、skywalking\_oap 组件默认启动。

V1.5.6 及之前版本，若运维平台默认未启动/手动关闭了这些运维组件，升级到 V1.5.7 版本后，组件仍然默认不启动，用户可自行启动。

详情请参见：[部署运维平台](#)



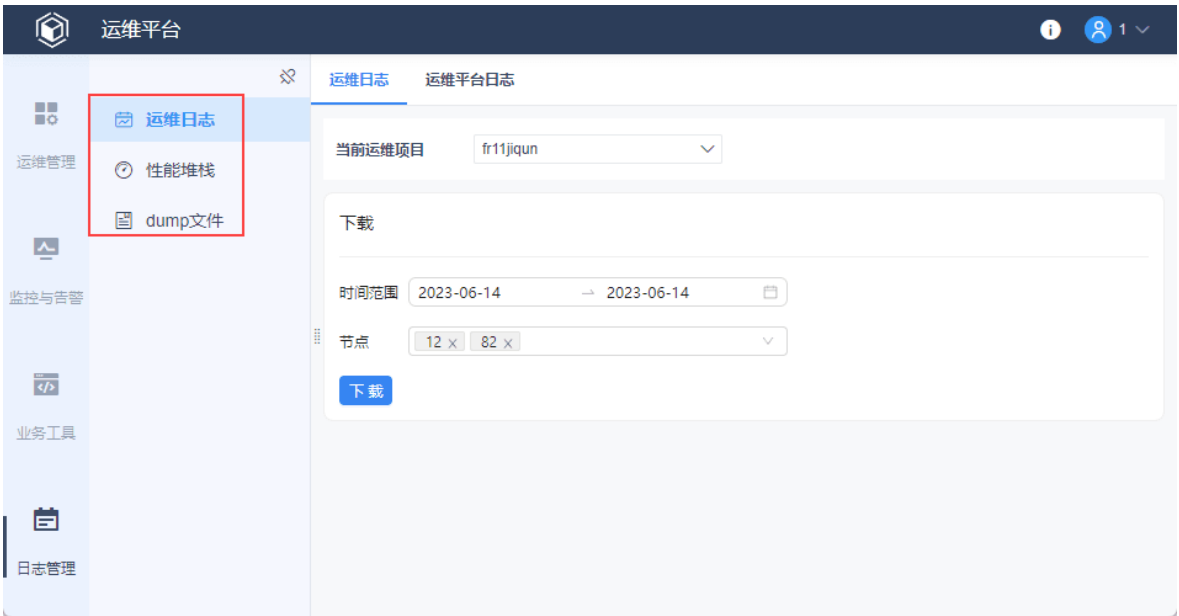
## 3. 运维功能

### 3.1 日志管理功能拆分优化

日志管理功能重新组合拆分，划分为三个模块：运维日志、性能堆栈、dump 文件。

不再以日志处理方式（生成、下载、分析）划分功能，而是以日志类型（运维日志、堆栈日志、dump 日志）划分，更贴合用户使用习惯。

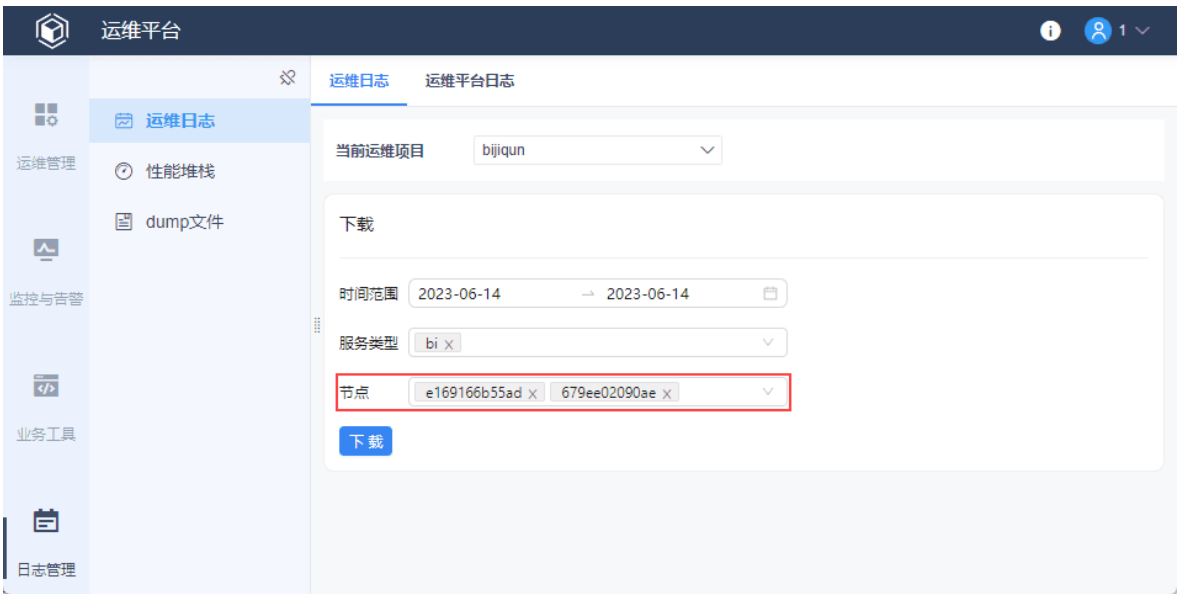
详情请参见：[日志管理](#)



### 3.2 支持单次下载多节点运维日志

对于多节点的集群运维项目，下载运维日志时，支持自定义选择多个节点，单次下载多节点日志文件。

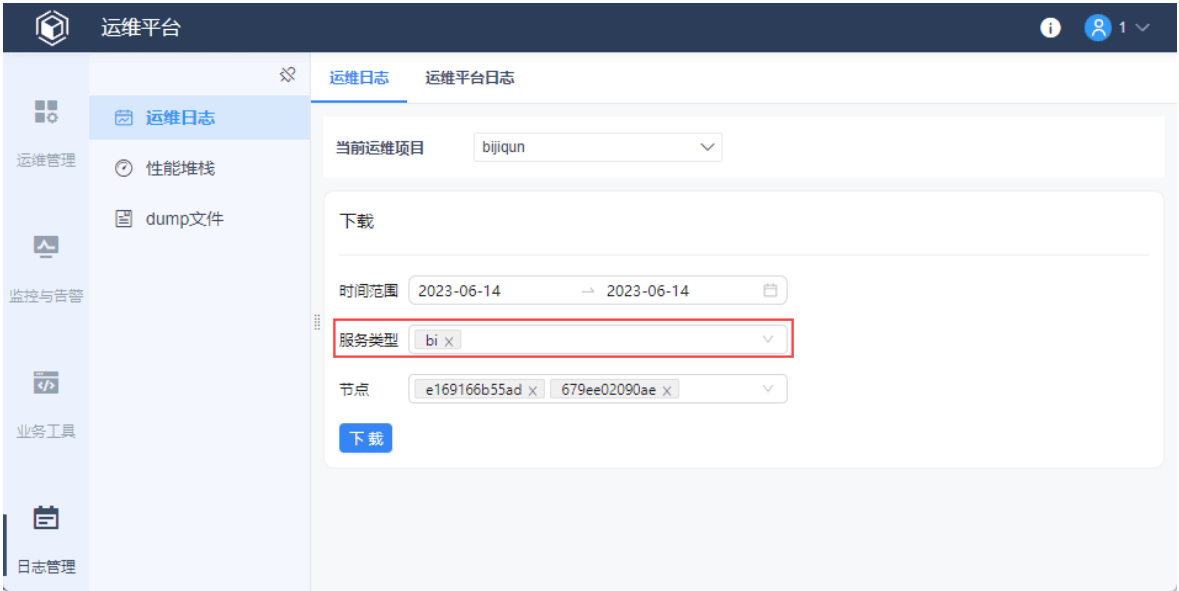
详情请参见：[运维日志](#)



### 3.3 支持自定义选择指定服务日志

各类日志下载时，支持选择服务类型，按需下载日志。

详情请参见：[运维日志](#)

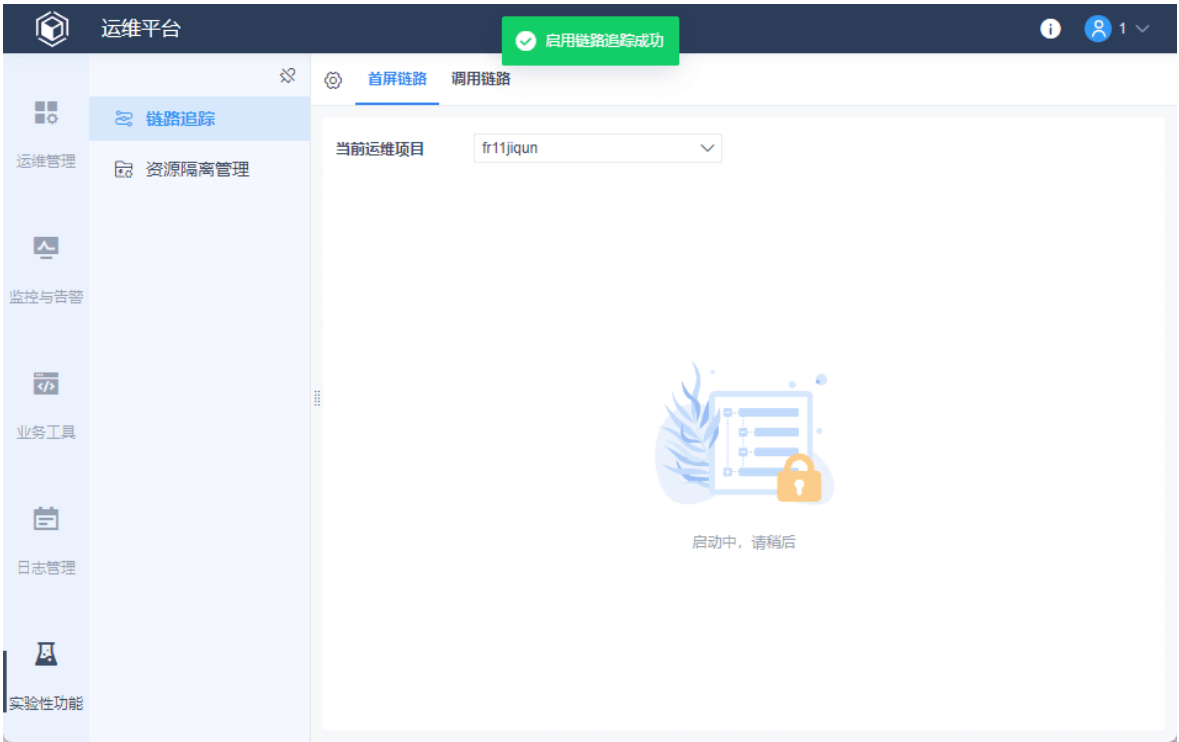


### 3.4 非容器化项目启用链路追踪步骤简化

为了降低用户使用成本，对于非容器化部署的项目，开启链路追踪功能的步骤全部简化

无需自行获取项目名称、手动拷贝文件、手动修改启动参数，全部交由 FineOps 运维平台一键配置，用户只需重启工程即可。

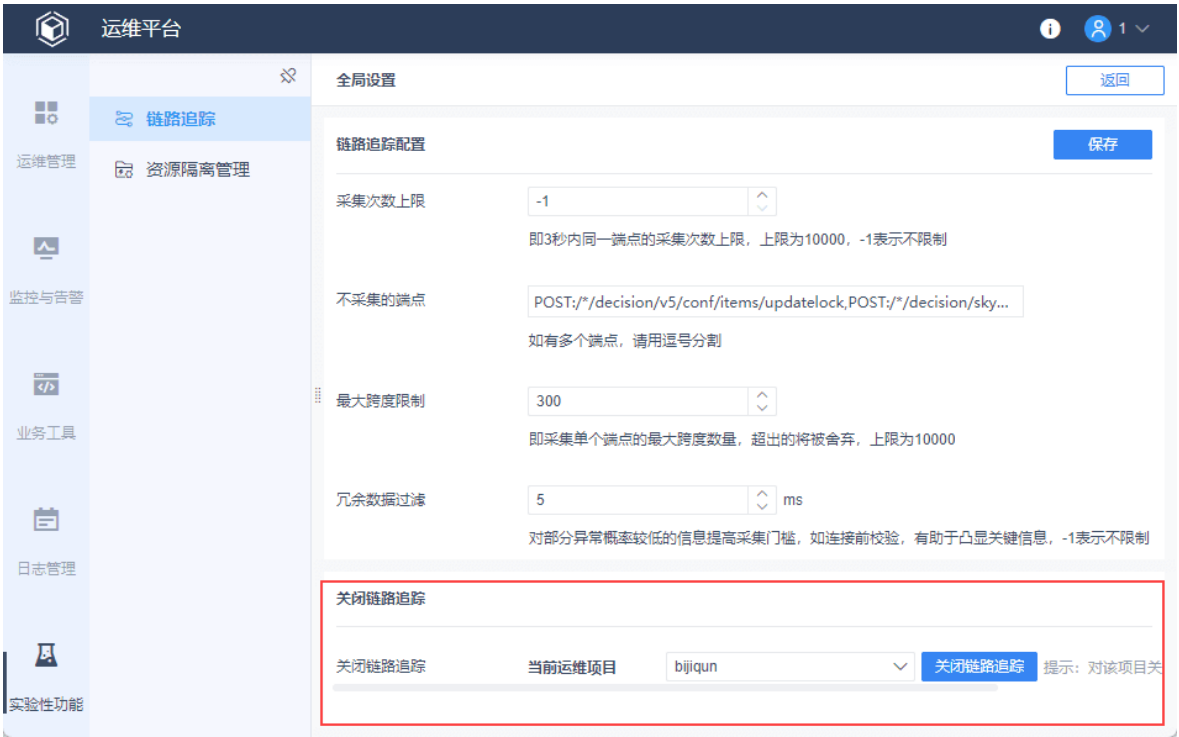
详情请参见：[非 finekey 部署项目接入链路追踪](#)



### 3.5 支持关闭链路追踪功能

FineOps 运维平台支持对运维项目快速停用链路追踪功能，重启项目后生效。

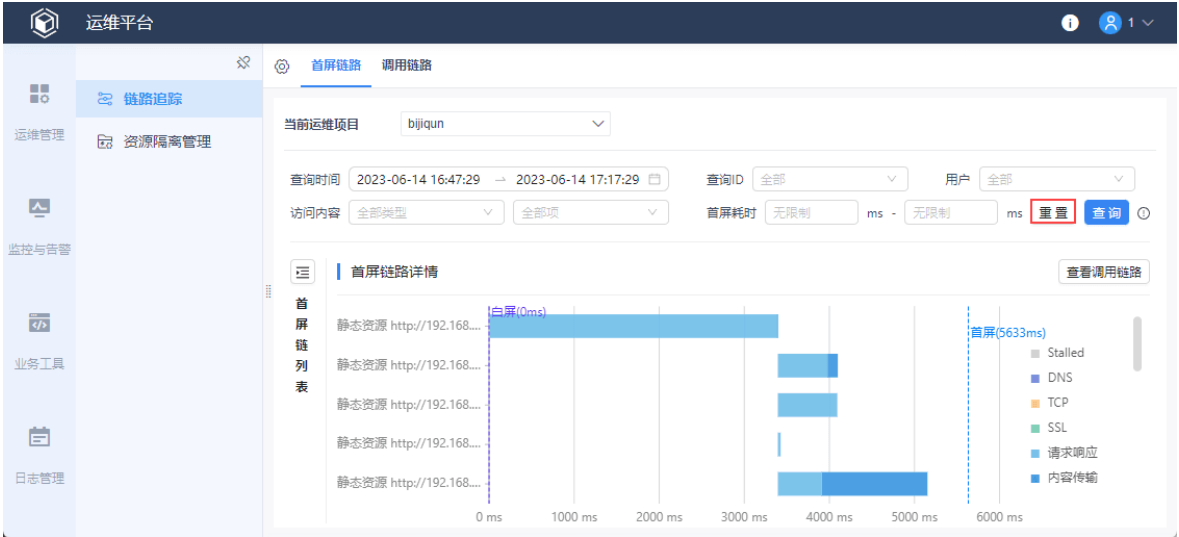
详情请参见：[非 finekey 部署项目接入链路追踪](#)



### 3.6 链路追踪新增重置按钮

链路追踪查询界面新增「重置」按钮，一键清空所有查询条件，方便重置查询。

详情请参见：[首屏链路](#)



### 3.7 链路追踪界面支持折叠链路列表

链路列表展示优化，支持展开收起，方便放大记录明细区。

详情请参见：[首屏链路](#)



# 1.2.13 V1.5.6 更新日志

## 1. 版本

运维平台版本	更新日期
V1.5.6	2023-05-31

## 2. 部署升级

### 2.1 部署包瘦身

为了让用户更便捷、快速的部署运维平台，运维平台安装包移除了部分冗余运维组件镜像，为部署瘦身。

1) 删除运维平台组件：loki、promtail、sherlock、nacos-server、skywalking-ui  
详情请参见：[部署运维平台](#)

2) 删除帆软应用组件：promtail、rocketmq\_namesrv、rocketmq\_broker  
详情请参见：[部署新项目-FineReport](#)

### 2.2 更新后自动升级运维组件

FineOps 运维平台升级完成后，会自动获取该版本依赖的最新运维组件镜像。运维组件也全部成功更新完成后，才能算作运维平台整体升级成功。

为了避免用户忘记更新运维组件，运维平台升级成功重启后，运维平台自动一键升级运维组件。

详情请参见：[外网升级运维平台](#)



## 3. 项目管理



### 3.1 运维平台对接应用优化

帆软应用只允许被一个运维平台接入。若已被接入，应用的运维密钥处显示接入状态为「已连接」。

详情请参见：[接入已有容器化项目](#)



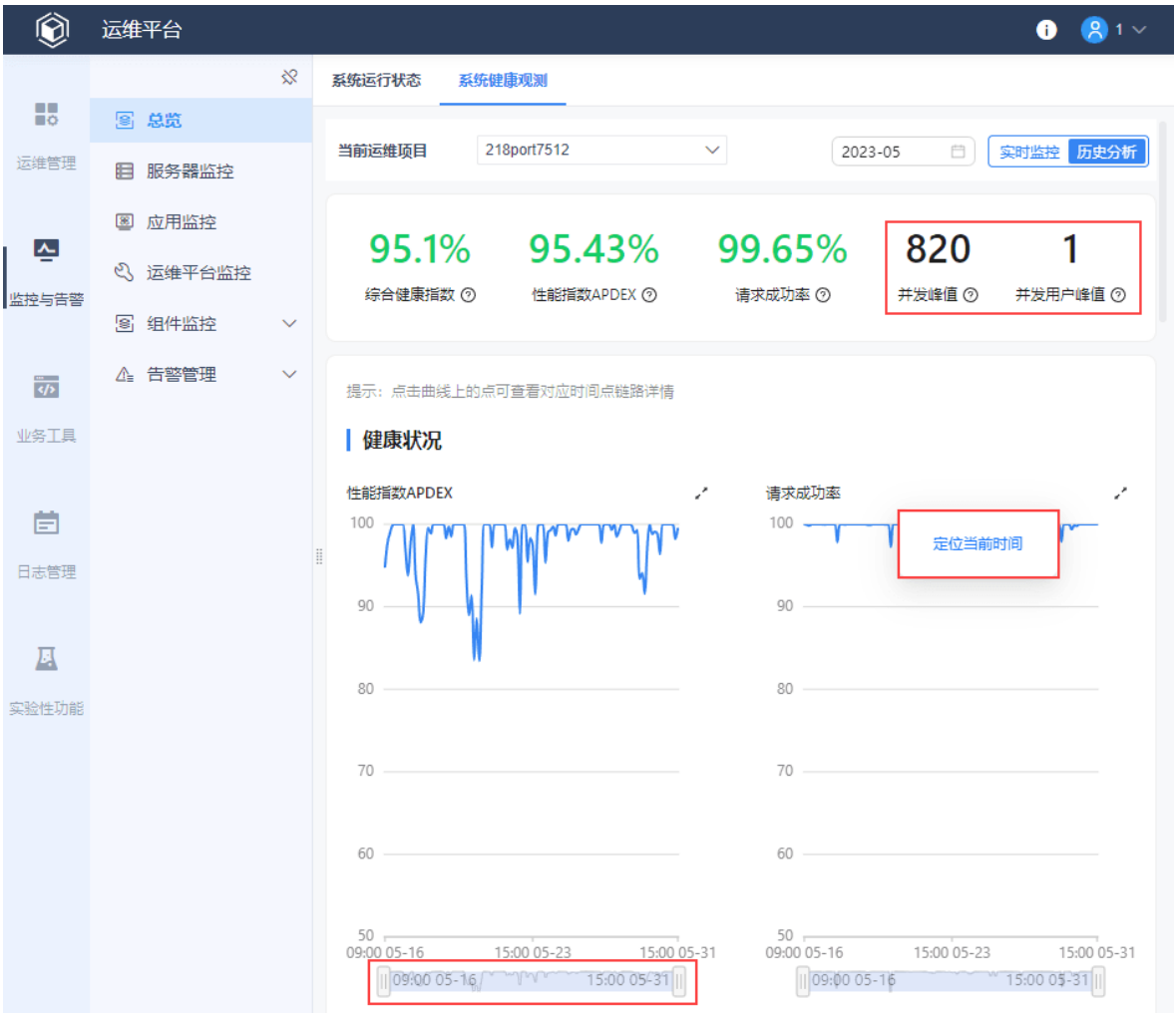
## 4. 运维功能

### 4.1 系统健康观测交互优化

系统健康观测界面展示优化

- 1) 请求并发量和在线用户数，优化为并发峰值和并发用户峰值
- 2) 各指标图支持互相「定位当前时间」
- 3) 各指标图支持拖动调整时间区间
- 4) 各指标图支持放大查看

详情请参见：[系统健康观测](#)

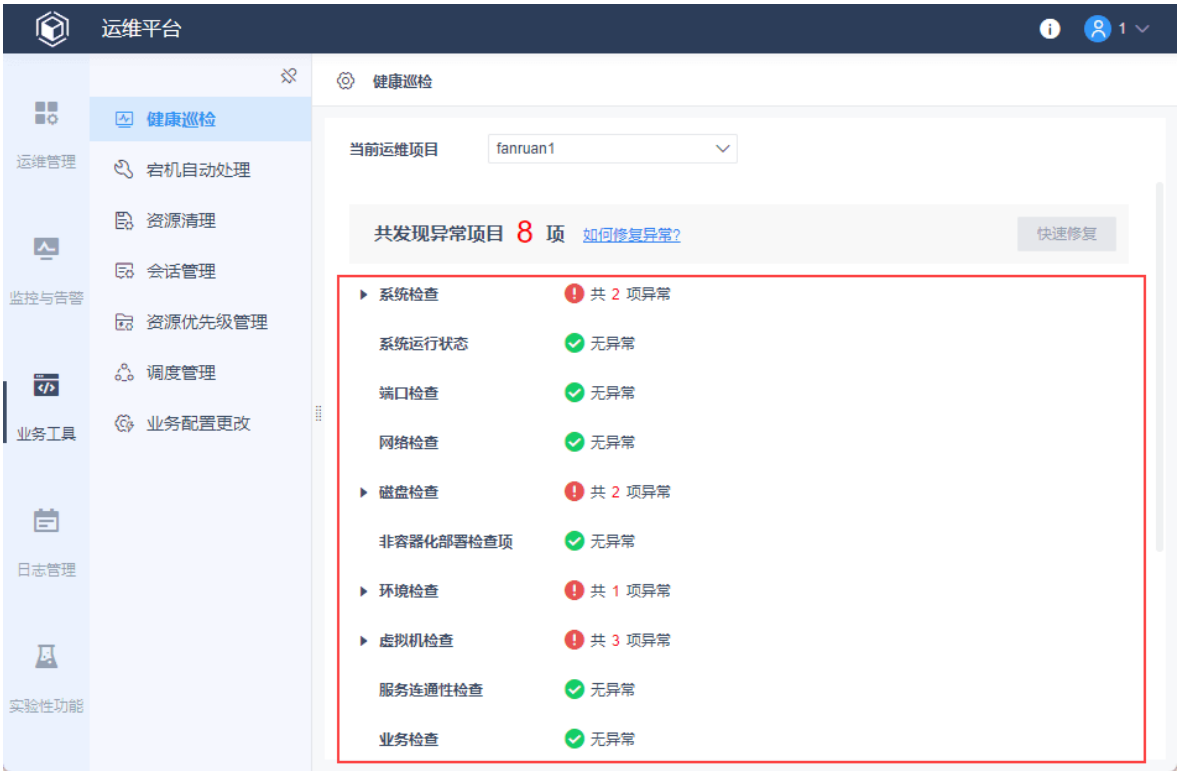


4.2 健康巡检新增检测项

为了满足管理员对系统的多维度检测需求，健康巡检新增多个检测项。

详情请参见：[健康巡检内容](#)

检测分类	新增检测项
	0113 CPU 信息
系统检查	0114 服务器字体编码
	0115 服务器字体包
业务检查	1025 是否开启云端运维
非容器化部署信息	0603 多进程检测
Java 虚拟机检查	0822 jdk 位置

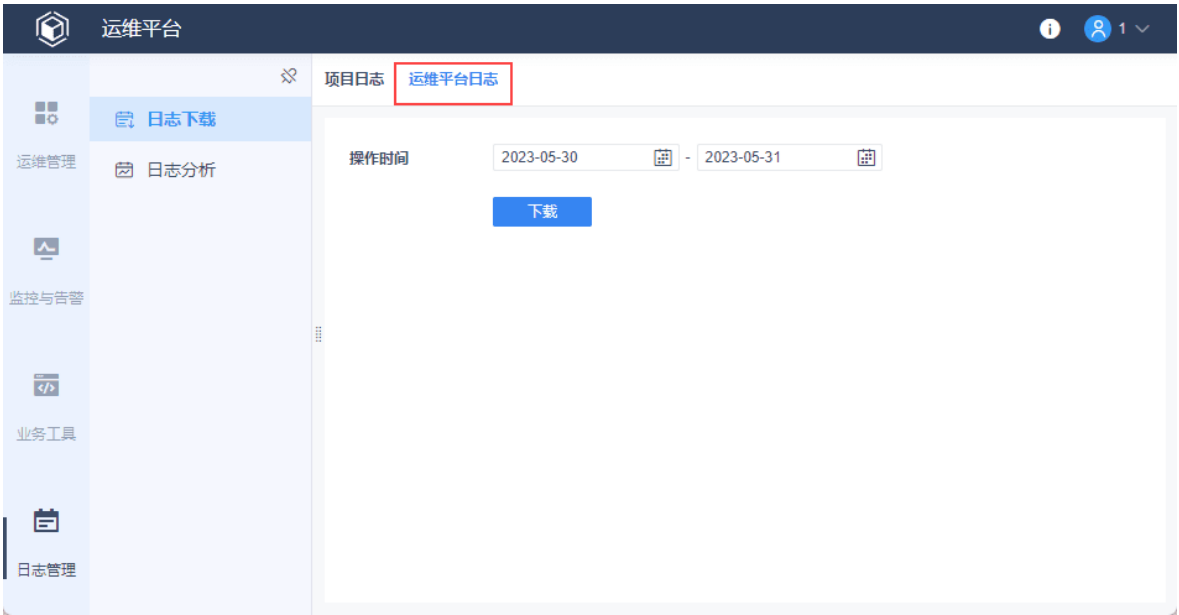


### 4.3 支持下载运维平台日志

运维平台功能出现问题时，需要通过运维平台自身的 fanruan.log 日志定位问题。

FineOps 运维平台支持下载自身日志。

详情请参见：[日志下载](#)



### 4.4 删除日志分析部分功能

删除「日志分析」部分功能，不再提供对出错日志、业务分析日志进行分析的相关功能。

详情请参见：[日志分析](#)

## 1.3 使用咨询&联系我们

### 1. 使用咨询

FineOps 运维平台，作为帆软应用（FineReport/FineBI）稳定运行和全方位运维管理的解决方案，不独立售卖。

- 对于帆软应用免费试用版：FineOps 运维平台可同步免费试用。
- 对于帆软应用商业版：购买帆软应用，即可免费使用 FineOps 运维平台。

### 2. 功能注册

FineOps 运维平台本身无需进行注册。

对于运维平台部署的帆软应用（FineReport/FineBI），推荐使用「[容器私有云认证](#)」方式进行注册。

### 3. 服务支持

若您是帆软的合作客户，当您遇到问题需要处理时，您可以直接通过服务平台进行提问，您也可以直接致电客服人员，寻求相应的帮助。

- 服务热线：400-811-8890 转 2
- [服务平台](#)：是帆软提供给合作/准合作客户的一套自助服务受理平台。请前往「服务平台」，选择「在线支持」。

客户使用此平台可以实现自助提交工单、实时查看工单进展及解决方案，为喜爱的需求投票等一系列功能。

具体使用说明可参考：[支持与服务平台使用说明](#)。

### 4. 问答社区

问答社区是帆软为所有用户提供的交流平台。该平台汇聚了不同产品的热门问题以及针对不同情况的解决方案。

您可以登录问答社区进行提问，会有专家或热心用户对您提出的问题，进行相应解答、探讨和指导。

平台入口：[问答](#)

## 5. 投诉与建议

服务不满意，请拨总裁办服务热线（24 小时在线，15 分钟回复）：17312781526，我们将竭诚为您服务！

# 1.3.1 帆软服务平台简介

## 1. 概述

### 1.1 应用场景

帆软服务平台 <https://service.fanruan.com/>（以下简称「服务平台」）是帆软提供给合作/准合作客户的一套自助服务受理平台。

客户使用此平台可以实现获取技术支持服务、自助提交工单、实时查看工单进展及解决方案，为喜爱的需求投票等一系列功能。

### 1.2 名词简介

**服务平台：**帆软服务平台 <https://service.fanruan.com/>（以下简称「服务平台」）是帆软提供给合作客户的一套在线自助服务受理平台。

**客户：**与帆软签订合作协议的公司或与帆软有合作意向的公司。

**帆软账号：**在帆软社区 <https://bbs.fanruan.com/> 注册的个人用户账号，登录后在账号中心 <https://fanruanclub.com/> 可以看到社区 ID，是帆软用户的唯一身份标识。

**企业管理员：**每个公司可以在服务平台认证为一个企业，企业管理员可以邀请公司其他成员加入本企业，只有加入后才可以访问和使用服务平台的诸多功能。企业管理员认证说明详见：[合作认证操作说明](#)

**企业成员：**被管理员邀请加入组织即可成为企业成员。加入后组织内所有成员提交的需求和工单进展共享。企业成员的操作说明详见：[企业成员操作方式](#)

## 2. 服务平台功能列表

编号	功能	网址	说明
1	企业合	<a href="https://fanruanclub.com/#/enterprise">https://fanruanclub.com/#/enterprise</a>	可查看客户与帆软的合作起始日期、销
	作信息		售顾问、售前顾问等。
2	查看	<a href="https://service.fanruan.com/">https://service.fanruan.com/</a>	管理员可以在此进行企业成员管理。
	在线支		可以在此获取实时人工支持
3	需求中	<a href="https://service.fanruan.com/demand">https://service.fanruan.com/demand</a>	【我提交的需求】可以自助提交 FineBI
	心		需求，提交后可以在此页签查看到进

			度。
			【组织需求进度查询】可以实时查看组织用户提交过的需求及进展。
			【新建工单】可以新建一条工单，新建后会有技术支持工程师与用户取得联系并解决工单问题。
			【查看工单进展】历史提交过的工单，都会实时显示当前进展，如果工单已关闭，将自动显示解决方案。
			【工单评价】可以对已提交的工单查看解决方案并评价。
			【主动提醒】提交的工单和需求如果状态发生变化，将以邮件方式推送到用户邮箱。
4	工单中心	<a href="https://service.fanruan.com/ticket">https://service.fanruan.com/ticket</a>	
5	服务月报	<a href="https://service.fanruan.com/evaluation">https://service.fanruan.com/evaluation</a>	可以查看过去一年/任意一个月的服务报告，包含接入人次、问题总数、需求总数和状态等。可以对服务进行评价。
6	云端运维	<a href="https://service.fanruan.com/OM">https://service.fanruan.com/OM</a>	可以查看客户的帆软应用的云端运维报告，提前排除宕机隐患，详情点击： <a href="#">云端运维报告</a> ， <a href="#">云端运维使用步骤</a>
7	资源导航	<a href="https://edu.fanruan.com/">https://edu.fanruan.com/</a>	可以获取与 FineReport/FineBI 产品相关的安装包、模板、白皮书、Demo 等资源
8	成员管理（仅管理员可见）	<a href="https://fanruanclub.com/#/enterprise">https://fanruanclub.com/#/enterprise</a>	可以企业成员进行添加、删除等操作
9	产品正式注册（仅管理员可见）	<a href="https://service.fanruan.com/licenses">https://service.fanruan.com/licenses</a>	可以在此自行进行产品注册、查看注册记录以及获取授权信息

3. 网络白名单

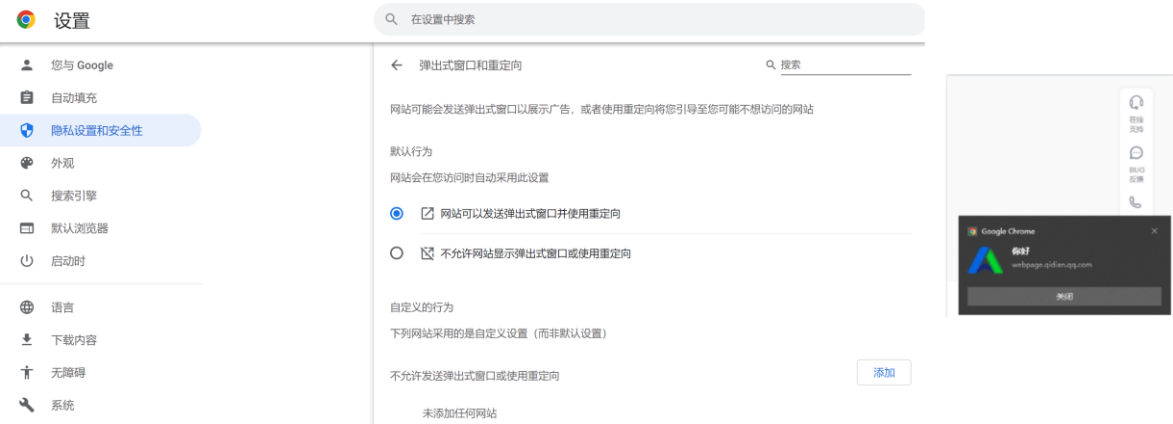
如需在内网环境下正常使用「服务平台」获取相关服务，需要联系公司网络运维相关同事开放网络白名单。需要开放的域名如下：

域名	端口
service.fanruan.com	443

at.alicdn.com	443
bqq.gtimg.com	443
fanruanclub.com	443
cid.fanruan.com	443
bbs.fanruan.com	443
shequ-oss.obs.cn-east-2.myhuaweicloud.com	443
cdn.fanruanclub.com	443
qq.com	-
qidian.qq.com	-
c2cpicdw.qpic.cn	-

4. 其他说明

- 1) 如客户帆软项目相关用户发生变动，企业管理员应及时维护相关人员信息，避免企业服务信息泄露、关键数据泄露等损失。成员管理方法见：[管理员管理方式](#)
- 2) 社区用户应遵守法律、行政法规相关规定，否则，帆软有权对违规用户账号采取包括但不限于限制、停止使用账号等相关措施。
- 3) 推荐您设置浏览器为“允许弹出通知”，以防无法收到消息提醒。



如果按照上述设置还是无法收到消息通知，可能是最新版的 Chrome 浏览器禁用了本机消息通知，可采用如下方案解决：

- 1) 在 Chrome 浏览器中访问地址：`chrome://flags`
- 2) 在搜索栏中搜索「notifications」，找到「Enable system notifications」选项，将其选项值改为「Disabled」，重启浏览器即可。



🔍 notifications

✕

Reset all

Experiments

112.0.5615.138

Available

Unavailable

Enable system notifications.

Enable support for using the system notification toasts and notification center on platforms where these are available. – Mac, Windows, Linux  
[#enable-system-notifications](#)

Default ▾

Default  
Enabled  
Disabled

Disruptive notification permission revocation

Enables revoking the notification permission on sites that send disruptive notifications unless the permission was granted through a prompt that informed the user about this possibility. – Mac, Windows, Linux, ChromeOS, Android, Fuchsia, Lacros  
[#disruptive-notification-permission-revocation](#)

Default ▾

Permission Module for notifications in Safety Check

When enabled, adds the notification permission module to Safety Check on desktop. The module will be shown depending on the browser state. – Mac, Windows, Linux, ChromeOS, Fuchsia, Lacros  
[#safety-check-notification-permissions](#)

Default ▾

## 1.3.2 合作认证操作说明

### 1. 概述

帆软旗下支持与服务平台 <https://service.fanruan.com/>（以下简称「服务平台」）是帆软提供给合作/准合作客户的一套自助服务受理平台。

客户使用此平台可以实现自助提交工单、实时查看工单进展及解决方案，为喜爱的需求投票等一系列功能。

本文介绍企业管理员如何在「服务平台」进行合作认证操作。

### 2. 前置准备工作

#### 2.1 确定管理员

客户需指定一人为服务平台的组织管理员，保管双方相关合作信息，管理双方项目相关用户信息，确保提供给帆软的所有资料和信息真实性、合法性、准确性和有效性。

我们推荐帆软系统的主要开发人员或者负责人作为管理员。

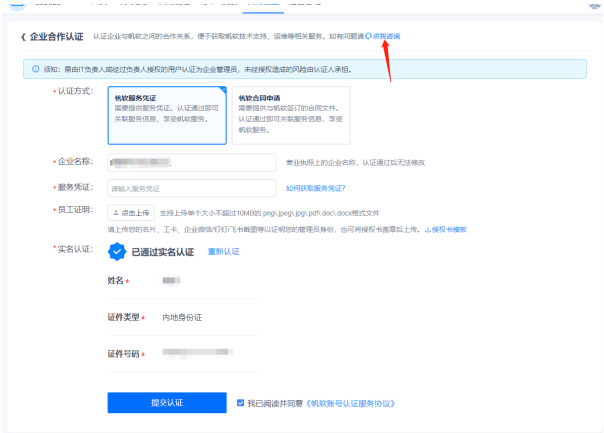
#### 2.2 准备管理员身份证明

请准备您的名片、工卡、企业微信/钉钉/飞书截图等以证明您的管理员身份，也可将[授权书](#)盖章后上传。

#### 2.3 准备合作关系证明

合作关系的证明有两种，具体用法见下表。

材料	获取方式	说明
服务凭证	详情请参见文档： <a href="#">服务凭证</a>	如果按照文档无法获取，请点击 <a href="#">认证页面</a> 右上角小助手咨询：
合同	合作双方自行留存，请联系贵公司采购部等部门获取	请扫描或者拍照上传，拍照需包涵金额页和盖章页



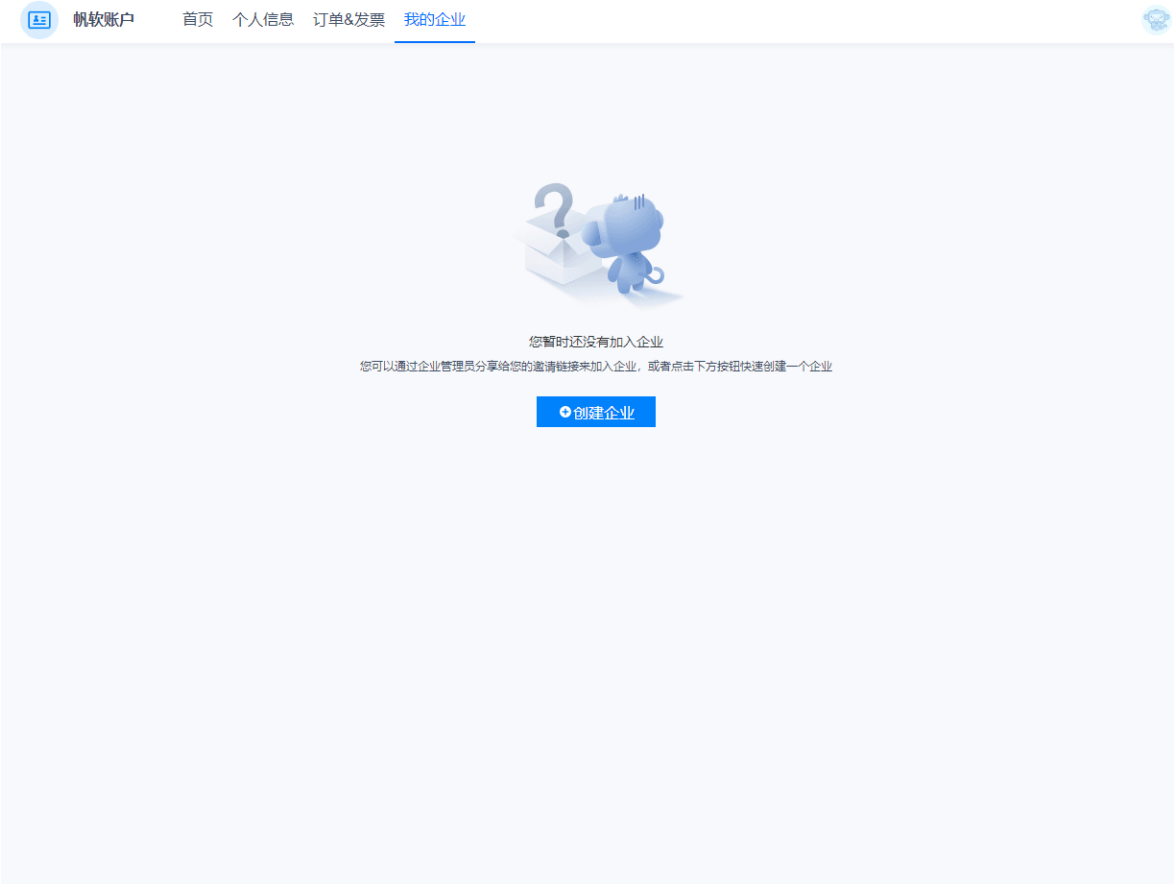
### 3. 注册帆软账号

访问帆软社区：<https://bbs.fanruan.com/>，点击右上角「注册」按钮进行注册。如下图所示：

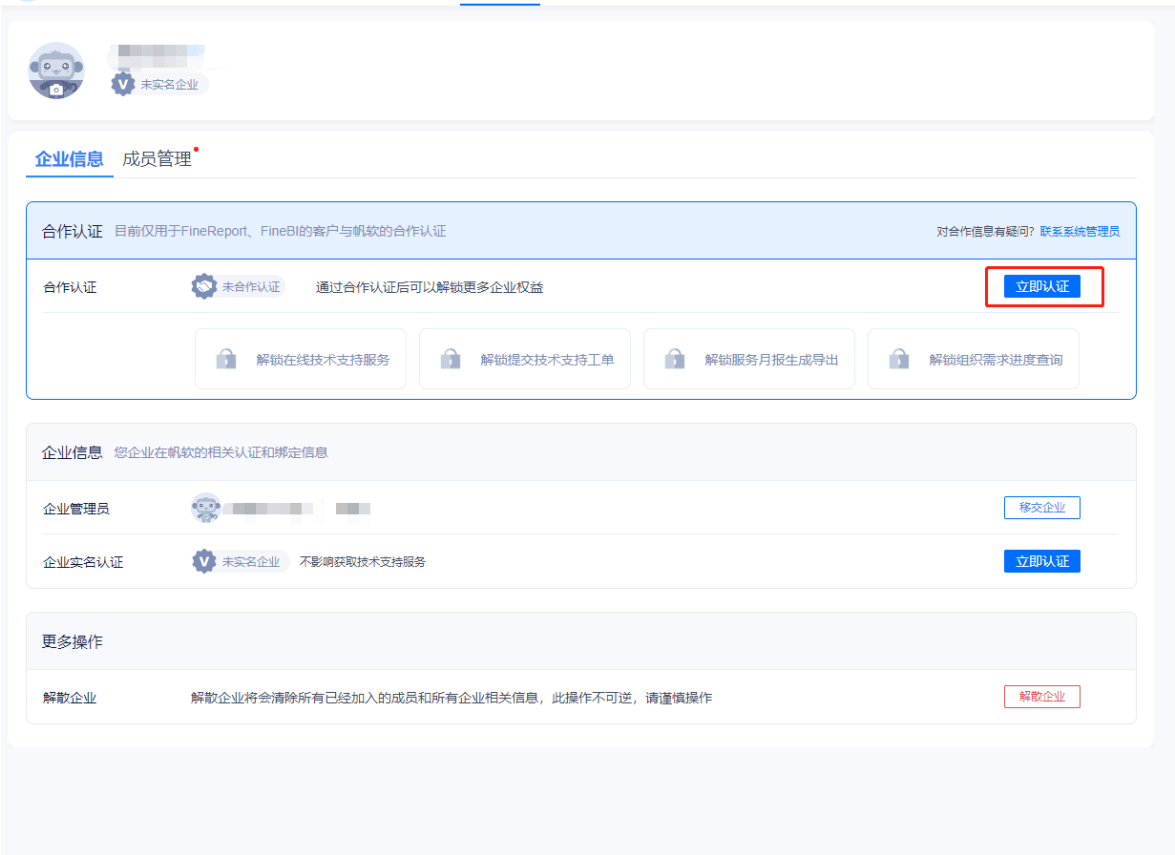


### 4. 进行合作认证

访问 [账号中心-我的企业](#)，点击「创建企业」。如下图所示：



创建成功后，点击合作认证中的「立刻认证」按钮，进行认证。如下图所示：



选择服务凭证或者合同方式，填写好的相关资料，提交即可。

< 企业合作认证

认证企业与帆软之间的合作关系，便于获取帆软技术支持、运维等相关服务。如有问题请[联系系统管理员](#)

须知：需由IT负责人或经过负责人授权的用户认证为企业管理员，未经授权造成的风险由认证人承担。

\* 认证方式：

帆软服务凭证

需要提供服务凭证。认证通过即可关联服务信息，享受帆软服务。

帆软合同申请

需要提供与帆软签订的合同文件。认证通过即可关联服务信息，享受帆软服务。

\* 企业名称：

营业执照上的企业名称，认证通过后无法修改

\* 服务凭证：

请输入服务凭证

[如何获取服务凭证？](#)

\* 员工证明：

点击下载

支持上传单个大小不超过10MB的.png\jpeg\jpg\pdf\doc\docx格式文件

请上传您的名片、工卡、企业微信/钉钉/飞书截图等以证明您的管理员身份，也可将授权书盖章后上传。[↓ 授权书模板](#)

\* 实名认证：

已通过实名认证

重新认证

姓名 \*

证件类型 \*

内地身份证

证件号码 \*

提交认证

☒ 我已阅读并同意《帆软账号认证服务协议》

认证成功后请等待管理员审核，通常 1 个工作日内审核完毕，如有疑问请点击 [我的企业](#) 右上角小助手咨询：

104



## 5. 变更管理员

认证成功后，当企业管理员发生变化，需要变更管理员时。

原管理员可以访问[账号中心-我的企业-企业信息](#)，点击移交企业按钮，选中某位成员成为新管理员。移交成功后立即生效。

注：新管理员需要先加入组织才能被选中。



## 6. 其他说明

- 1) 如客户帆软项目相关用户发生变动，企业管理员应及时维护相关人员信息，避免企业服务信息泄露、关键数据泄露等损失。成员管理方法见：[管理员管理方式](#)
- 2) 社区用户应遵守法律、行政法规相关规定，否则，帆软有权对违规用户账号采取

包括但不限于限制、停止使用账号等相关措施。

### 1.3.3 管理员管理方式

#### 1. 概述

帆软旗下支持与服务平台 <https://service.fanruan.com/>（以下简称「服务平台」）是帆软提供给合作/准合作客户的一套自助服务受理平台。

客户使用此平台可以实现自助提交工单、实时查看工单进展及解决方案，为喜爱的需求投票等一系列功能。

本文介绍企业管理员如何在「服务平台」邀请成员加入企业或删除离职员工。

#### 2. 添加成员方法

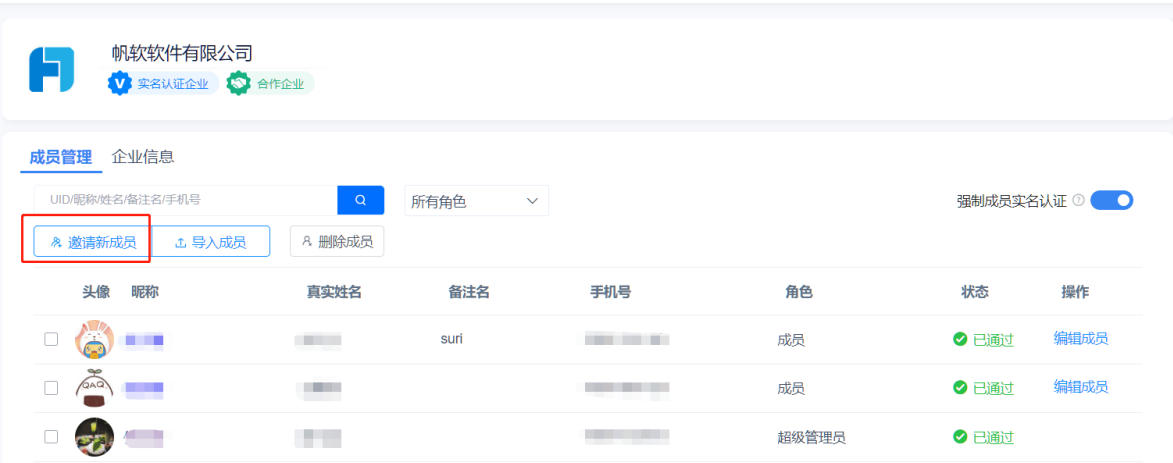
添加成员有两种方法，一种是生成分享链接发送给用户，用户点击后即可加入企业；一种是通过 Excel 导入预先设置好的成员名单。管理员可任选其一进行添加。

下面分两部分进行介绍。

##### 2.1 通过邀请链接添加用户

管理员认证成功后，访问 [账号中心-我的企业-成员管理](#)，点击“邀请新成员”按钮，可以生成邀请链接，其他用户点击该链接即可加入企业。

注：邀请按钮支持邀请普通管理员或者成员，普通管理员可以协助超级管理员进行成员管理。

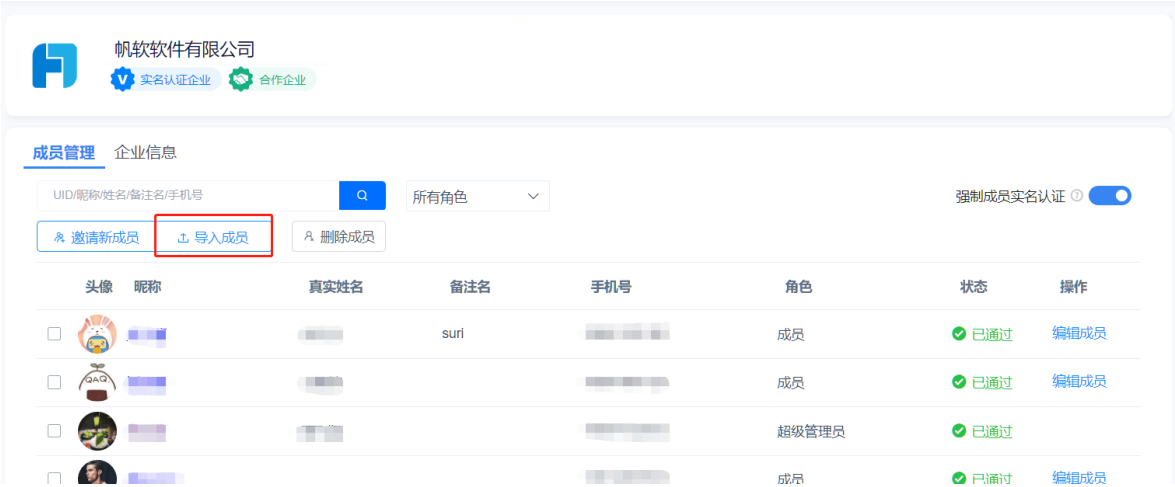


##### 2.2 通过 Excel 导入用户

管理员认证成功后，访问[账号中心-我的企业-成员管理](#)，点击“导入成员”按钮，下载成员导入模板，将成员信息填写好后，上传 excel 即可。

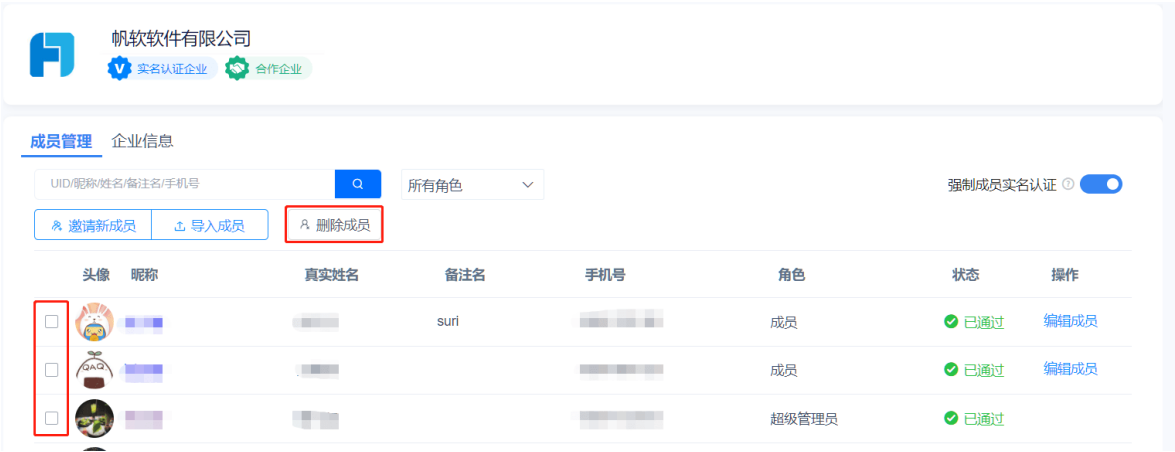
注 1：成员导入是增量导入，每次导入都会新增用户，如需删除，请参考下方删除成员操作。

注 2：需要填写成员之前注册帆软时使用的手机号，否则会匹配不上。



3. 删除已离职成员

勾选要删除的成员， 点击删除按钮即可删除。



4. 其他说明

- 1) 如客户帆软项目相关用户发生变动，企业管理员应及时维护相关人员信息，避免企业服务信息泄露、关键数据泄露等损失。
- 2) 社区用户应遵守法律、行政法规相关规定，否则，帆软有权对违规用户账号采取包括但不限于限制、停止使用账号等相关措施。



## 1.3.4 企业成员操作方式

### 1. 概述

帆软旗下支持与服务平台 <https://service.fanruan.com/>（以下简称「服务平台」）是帆软提供给合作/准合作客户的一套自助服务受理平台。

客户使用此平台可以实现自助提交工单、实时查看工单进展及解决方案，为喜爱的需求投票等一系列功能。

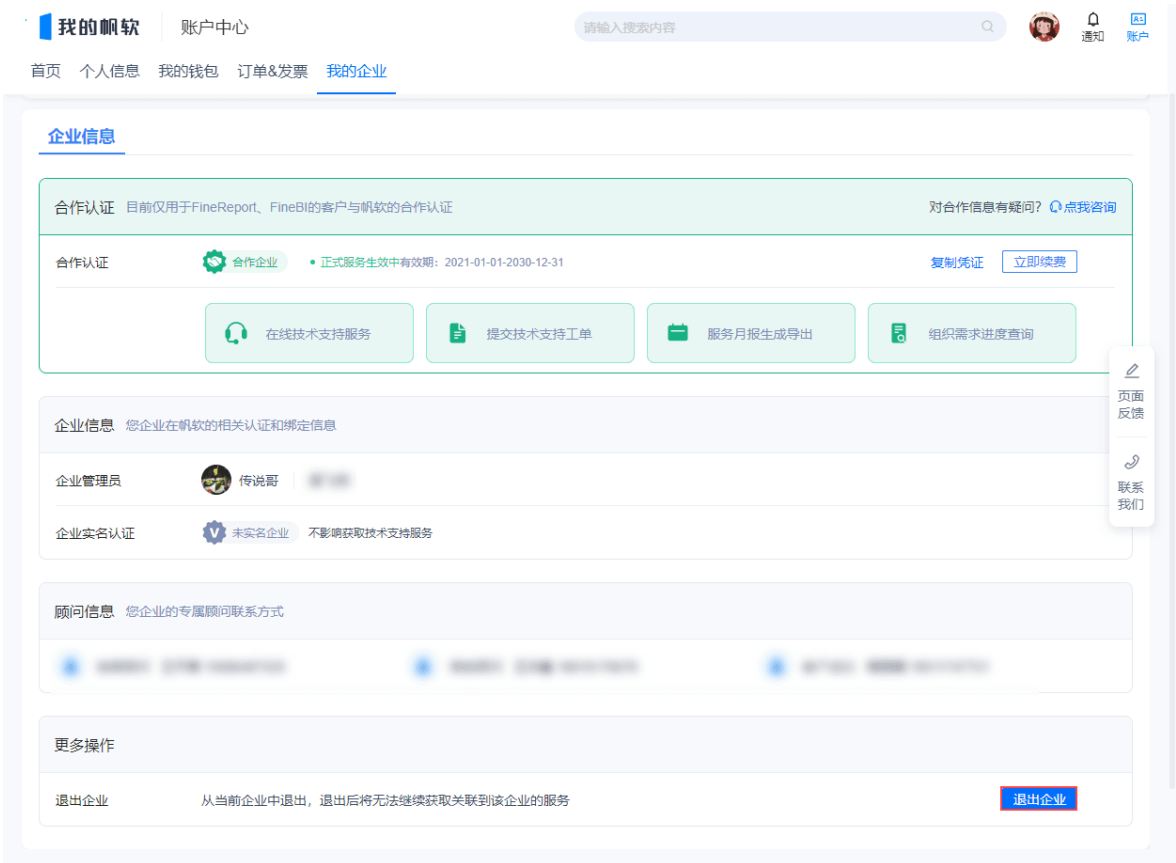
本文介绍企业员工如何在「服务平台」自主加入企业或者退出企业。

### 2. 加入企业

可以向管理员索要加入企业的连接，点击连接即可加入。

### 3. 退出企业

如果已加入的企业想要退出，访问[我的企业](#)，点击右下角退出该企业即可。



### 4. 其他说明

- 1) 如您的所在企业发生变动，请及时更新您的所属企业信息，避免企业服务信息泄露、关键数据泄露等损失。
- 2) 社区用户应遵守法律、行政法规相关规定，否则，帆软有权对违规用户账号采取包括但不限于限制、停止使用账号等相关措施。



## 2. 运维平台场景案例

场景主题

场景痛点解决思路[应用宕机处理与分析](#)

### 1) 没有收集宕机信息

发生了内存宕机类问题，由于生产环境，必须要第一时间恢复环境，没有来得及登录到后台输入命令收集宕机信息，导致问题无法定位。

### 2) 无法 24h 值班监守

夜间发生了宕机，由于没有人 24h 监控，或者由于没有办法触碰到内网环境，导致第二天才能够恢复，导致发生了生产事故。

运维平台自动识别宕机情况，并且自动采集宕机信息，采集完信息后，自动重启工程，保证工程持久运行。

### 1) 宕机前的预防

为了防止宕机时无人值守，需要开启宕机处理策略，确保运维平台可以自动重启工程。

### 2) 宕机时的采集

宕机时，自动导出内存堆栈日志信息

### 3) 采集后的恢复

宕机信息采集完成后，运维平台帮助管理员自动重启工程、恢复进程。

### 4) 恢复后的分析

宕机时，运维平台会自动导出内存堆栈，可以给出针对宕机的原因分析和解决方案，可以自动对宕机堆栈进行分析。

[应用配置修改与启停](#)

### 1) 组件配置修改麻烦

组件配置在运维过程中经常需要修改，每种组件修改位置和方法不完全相同，容易修改错误，因此修改成本高

对于 FR/BI 项目组件。管理员常常需要对 Nginx/Redis/配置库进行配置修改

对于 FR/BI 工程，管理员常常需要修改工程配置，确保工程平稳运行

### 2) 修改后重启组件麻烦

组件的配置修改，一般都需要配合组件的重启

重启组件一般都需要登录到后台操作，操作成本高

配置修改万一出错，需要反复重启，成本加倍

针对 Nginx/Redis/配置库、FR/BI 工程，

运维平台分别提供界面化组件修改和组件启停功能

[应用监控与告警](#)

### 1) 工程节点多，管理麻烦

一个 FR/BI 集群项目，在多台服务器上分布着各个节点，管理员需要监控每一个节点正常运行

### 2) 集群组件多，管理麻烦

一个 FR/BI 项目，不仅只有 FR/BI 组件，还有外接数据库、负载转发、状态服务器、文件服务器等组件，管理员需要监控每一个组件正常运行

### 3) 监测组件配置异常

除了确保以上服务器、工程、组件的正常运行，管理员需要监控他们的配置是否正常，是否有异常现象

当出现异常之后，类似于负载异常或组件连接异常，管理员需要能及时收到相关通知信息，而不是事后补救

### 4) 日志获取困难，分析门槛高

为了分析这些异常，管理员需要登录到这些服务器后台去拿对应的日志，获取很困难  
通过一个运维平台可以对接多个节点/应用，同时能够监控多个环境，任意一个节点/应用出现问题都可以统一预警，也能够通过运维平台方便的获取到信息。

#### 1) 监控项目正常运行

项目是否可用、节点是否正常、组件是否启动

#### 2) 界面化查看项目配置

服务器监控、应用监控、组件监控

#### 3) 项目出现异常时预警

企业微信预警、邮件预警、webhook 预警

#### 4) 获取日志分析异常

支持在线查看日志分析结果，支持下载日志到本地自行分析

[应用资源分配](#)

**1) 保障核心业务性能和稳定性**

公司中有多个部门，可能存在高并发访问，需要优先确保一些重要业务/用户优先使用资源（例如领导使用，对外大屏，会议使用等）。

**2) 业务级别故障隔离，避免不同部门业务相互影响**

系统中存在几张模板性能不好（大数据量导出/填报/预览），业务部门不希望对模板进行限制，但又不想影响到其他业务。

系统原本的模板正常运行，新增了一些功能业务后，希望先在单独环境中使用一段时间，不要影响到原有业务。

**3) 开发发布环境统一但互不影响**

公司没有测试系统，测试和生产环境在同一个应用中，需要将他们隔离到不同节点，不互相影响。

运维平台提供资源分配相关功能。

**1) 资源优先级管理**

管理员可对系统业务（目录、模板、仪表板、公共链接）和用户（支持按照部门、角色、用户筛选）进行优先级排序，确保高优先级业务优先调用系统资源

**2) 资源隔离**

集群多节点应用，可以根据节点拆分为多个业务组，将不同类型的业务（按照测试/生产、原有/新增、预览/填报）放置到不同业务组中，确保互相不影响。

**3) 调度管理**

使用调度管理功能，可检查系统中不同时段的调度压力，合理设置业务定时任务的触发时间。

**4) 组件管理**

一个 FR/BI 应用中，有很多组件，运维平台可以限制每个组件对系统 CPU 和内存的占用，防止某个组件大量占用服务器资源，导致其他组件性能差

[应用健康观测](#)

**1) 用户体验差：**

用户向管理员反馈，感觉系统使用卡慢

用户使用某种仪表板/模板时，感觉卡慢

## 2) 管理员排查难：

无法衡量和定位问题

需要被动的等待用户反馈问题

需要自行拿日志、打堆栈，排查问题，门槛高

FineOps 运维平台提供健康观测与链路追踪功能，帮助管理员定位用户查看、分析仪表板/报表的性能卡慢问题。

### 1) 提前预警

管理员可以自行查看系统健康观测，在用户反馈前主动发现系统出现了卡慢，无需被动等待

### 2) 数据还原

将用户实际体验，以数据形式准确展示出来

### 3) 快速定位

还原用户真实体验，并将定位时间从缩短至 10 分钟内

[项目备份还原](#)

FineReport/FineBI 提供了「备份还原」功能，但是存在一些不足。

#### 1) 无法异地备份

工程部署在 A 服务器上，备份的内容也在 A 服务器上，结果小明误删除的时候，把工程和备份都删除了。

#### 2) 备份速度缓慢

随着工程中的模板、数据越来越多，应用备份的速度越来越慢了。

#### 3) 还原逻辑复杂

JAR 包、模板、配置都是分开备份的，但是大多时候需要将这些内容按照时间点统一备份还原。

#### 4) 备份内容不全

一些重要内容不在备份还原范围内，会影响工程的正常运行，例如 classes、resources 等。

针对备份还原的相关诉求，运维平台提供「备份还原」功能，帮助用户对工程进行**异地、整体、快速**的备份还原。[应用更新升级](#)

为满足客户的新需求以及完善之前版本某些功能的不足之处，FineReport/FineBI 在不断地更新迭代。

无论何种场景，在对 FineReport/FineBI 进行升级之前，都必须对工程进行备份，尽量对除了状态服务器和负载均衡服务以外的工程进行完整备份。

因为可能存在一些意外情况，导致更新对配置库进行了更改，或部分资源文件出现问题，必须保证升级失败后可以快速回退。

简单介绍，对于内网/外网、FineReport/FineBI、单机/集群、容器化部署/非容器化部署等工程，不同情况下的最佳升级方案。

## 2.1 场景案例-应用宕机处理与分析

### 1. 概述

#### 1.1 用户痛点

- 发生了内存宕机类问题，由于生产环境，必须要第一时间恢复环境，没有来得及登录到后台输入命令收集宕机信息，导致问题无法定位。
- 夜间发生了宕机，由于没有人 24h 监控，或者由于没有办法触碰到内网环境，导致第二天才能够恢复，导致发生了生产事故。

#### 1.2 解决思路

运维平台自动识别宕机情况，并且自动采集宕机信息，采集完信息后，自动重启工程，保证工程持久运行。

- 宕机前的预防：为了防止宕机时无人值守，需要开启宕机处理策略，确保运维平台可以自动重启工程。
- 宕机时的采集：宕机时，自动导出内存堆栈日志信息
- 采集后的恢复：宕机信息采集完成后，运维平台帮助管理员自动重启工程、恢复进程。
- 恢复后的分析：宕机时，运维平台会自动导出内存堆栈，可以给出针对宕机的原因分析和解决方案，可以自动对宕机堆栈进行分析。

### 2. 案例详情

#### 2.1 宕机前的预防

在「业务工具>宕机自动处理>宕机处理策略」，开启宕机自动处理相关功能，确保运维平台发生宕机后运维平台可以进行处理措施。

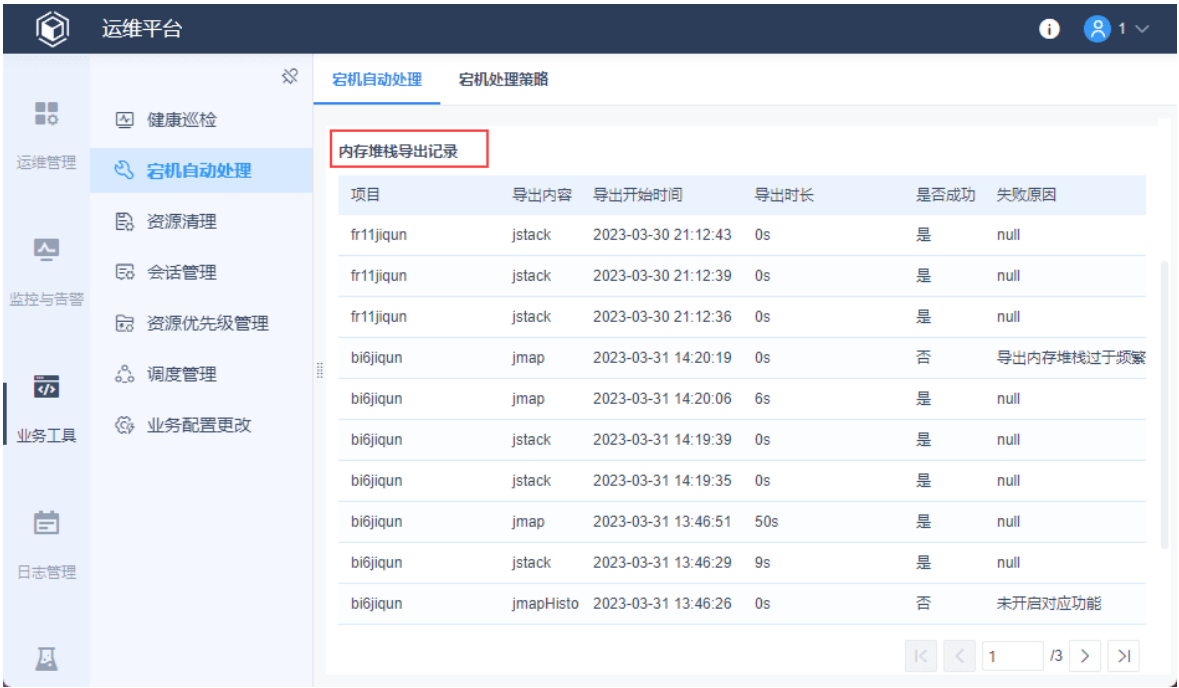


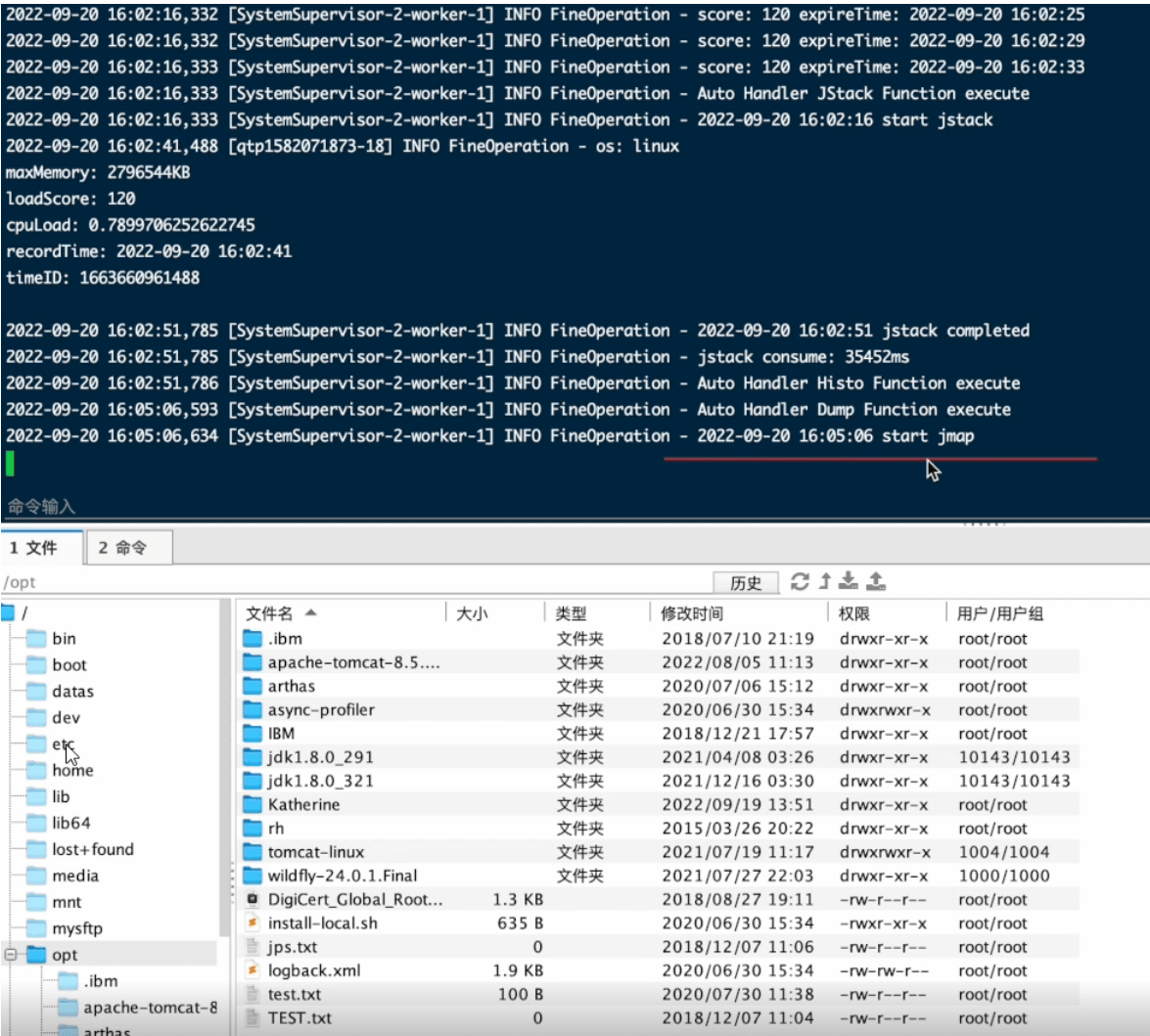


## 2.2 宕机时的采集

运维平台自动采集当前宕机状态的 FRBI 信息。

- 运维平台在负载极高判定为宕机的情况下自动触发信息采集。
- 自动采集堆栈/histo/heapdump （可在上一步根据自身需求自定义采集哪些内容）



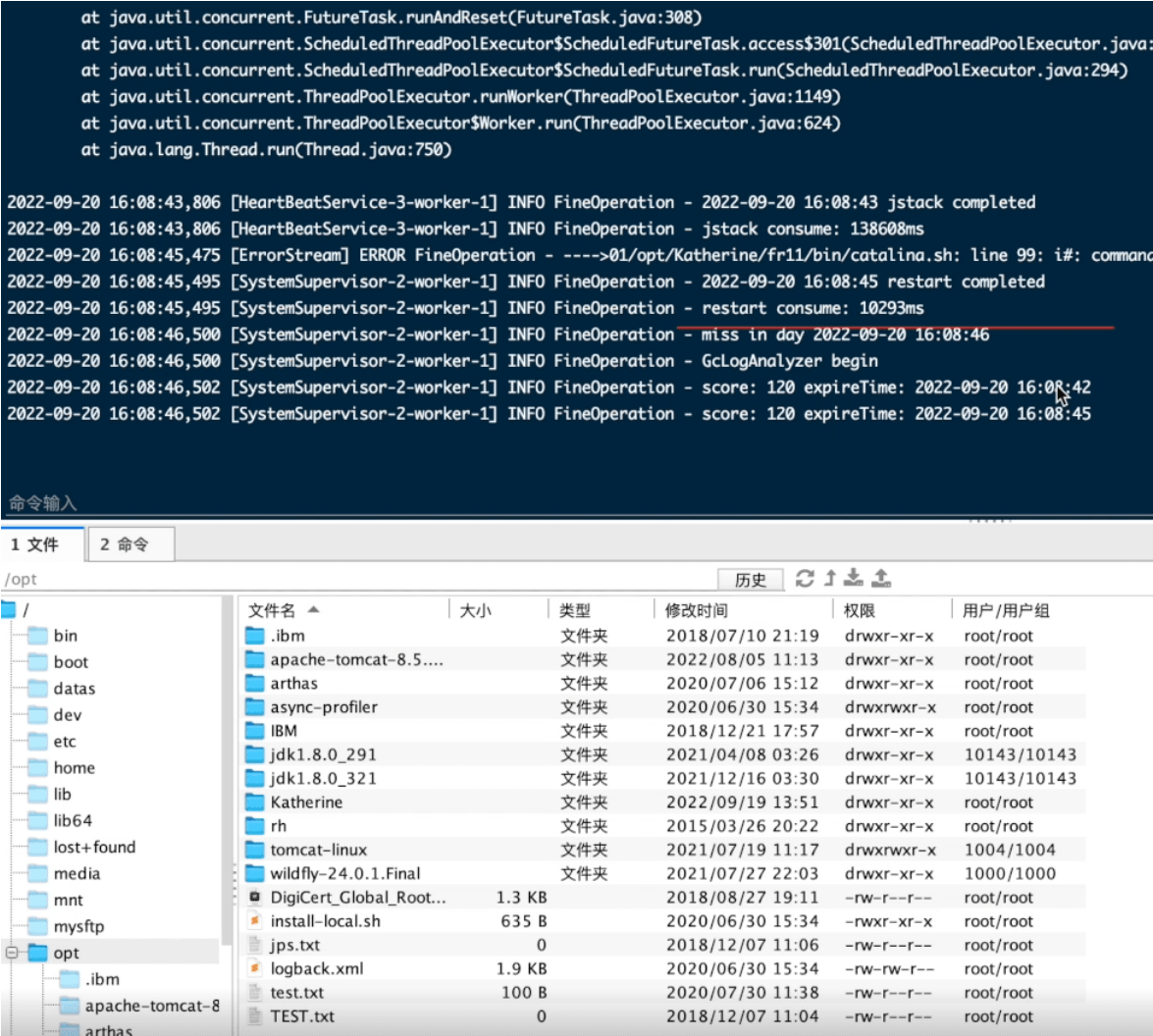


2.3 采集后的恢复

采集完宕机状态信息后自动重启 FRBI 应用，恢复系统使用。

- 采集宕机信息后自动重启
- 夜间检测到进程消失自动启动





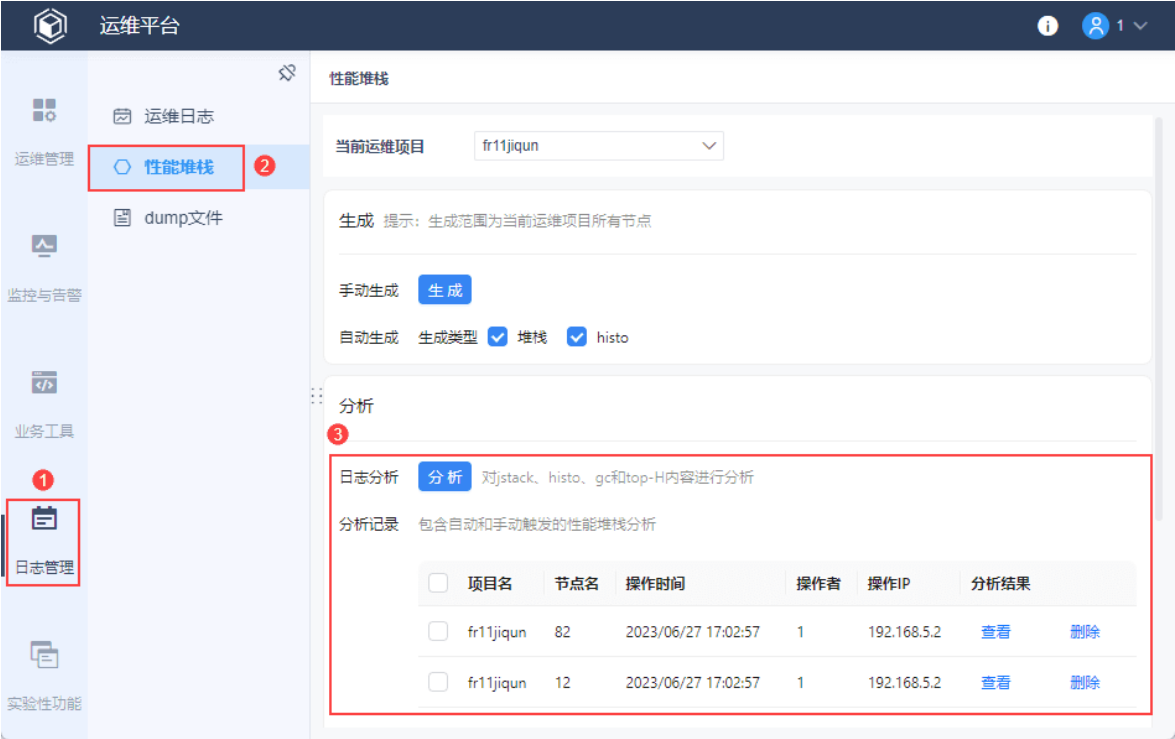
2.4 恢复后的分析

1) 运维平台宕机自动处理页面，查询宕机信息和分析结果

- 进入运维平台宕机自动处理页面可以查看到宕机事故记录/信息  
采集时间/重启时间
- 查看宕机原因和推荐处理方案

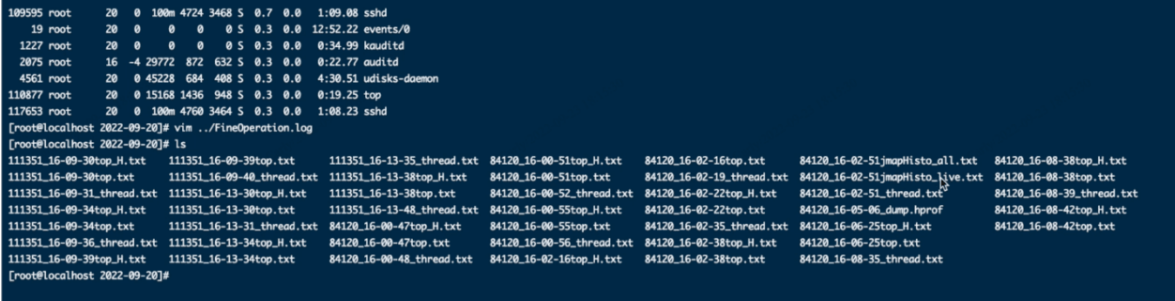


2) 运维平台「日志管理>性能堆栈」页面，查看对应宕机日志的分析报告。



3) 查看宕机信息

FRBI 应用对应目录下，可以查看到运维平台自动采集的信息。



## 2.2 场景案例-应用配置修改与启停

### 1. 概述

#### 1.1 用户痛点

1) 组件配置在运维过程中经常需要修改，每种组件修改位置和方法不完全相同，容易修改错误，因此修改成本高

- 对于 FR/BI 项目组件。管理员常常需要对 Nginx/Redis/配置库进行配置修改
- 对于 FR/BI 工程，管理员常常需要修改工程配置，确保工程平稳运行

2) 组件的配置修改，一般都需要配合组件的重启

- 重启组件一般都需要登录到后台操作，操作成本高
- 配置修改万一出错，需要反复重启，成本加倍

#### 1.2 解决思路

针对 Nginx/Redis/配置库、FR/BI 工程，运维平台分别提供界面化组件修改和组件启停功能。

### 2. 案例一：修改 FR/BI 项目组件配置

#### 1) 发现问题

使用 MySQL 数据库作为外接配置库时，为了防止单条记录过长导致写入失败，必须调整 MySQL 数据库的「max\_allowed\_packet」值。

此时如果查看 mysql 的 max\_allowed\_packet 值，发现并未配置。

```
mysql> select @@max_allowed_packet;
+-----+
| @@max_allowed_packet |
+-----+
|          137363456 |
+-----+
1 row in set (0.00 sec)

mysql> show variables like 'max_allowed_packet';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| max_allowed_packet | 137363456 |
+-----+-----+
1 row in set (0.01 sec)
```

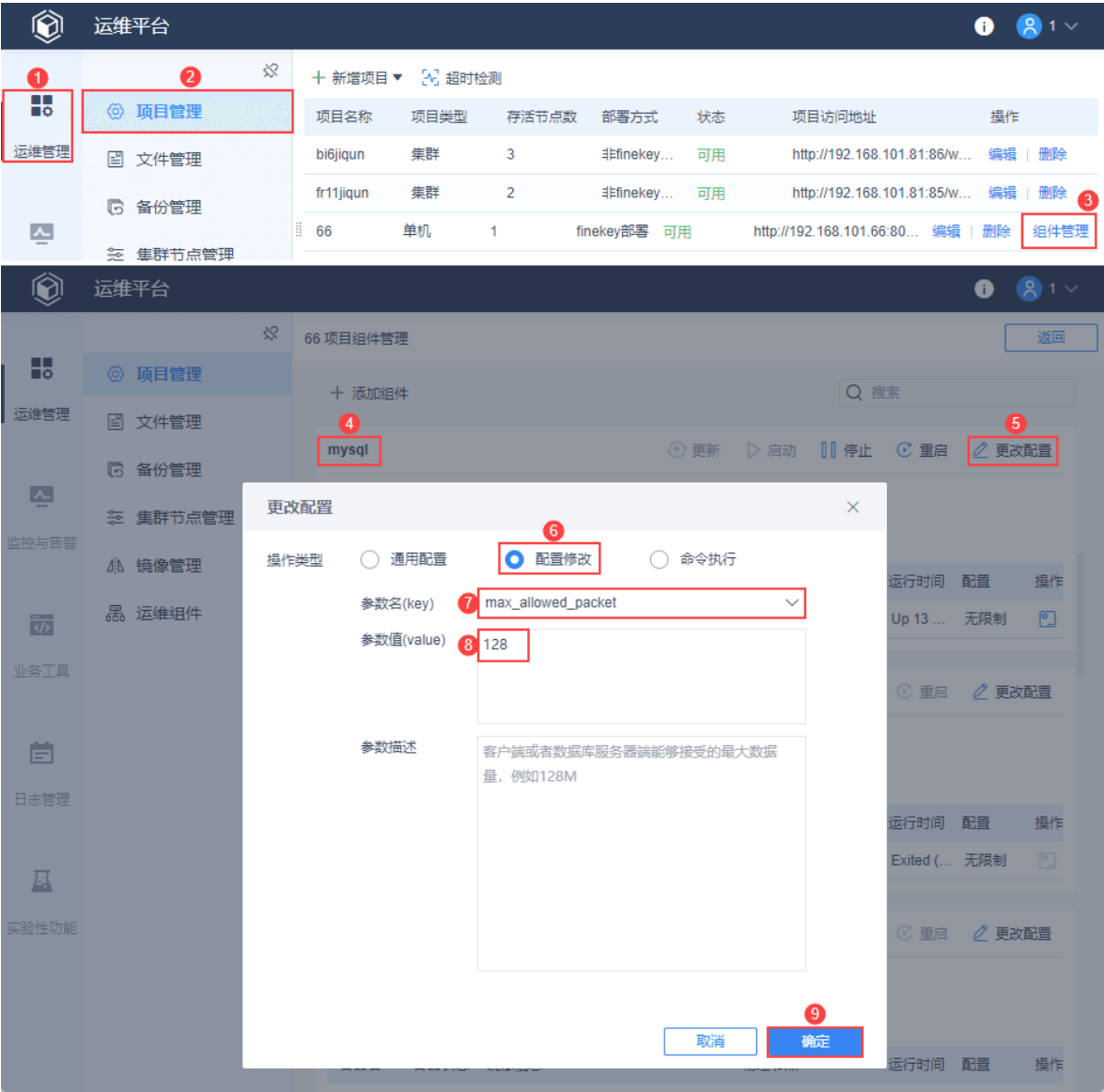
2) 修改 FR/BI 项目组件配置

管理员登录运维平台, 点击「运维管理>项目管理」, 点击目标项目对应的「组件管理」。

找到 MySQL 组件, 点击「更改配置>配置修改」, 找到参数「max\_allowed\_packet」。

根据参数描述, 填入参数值。

点击「确定」, 保存后, 会提示重启生效。

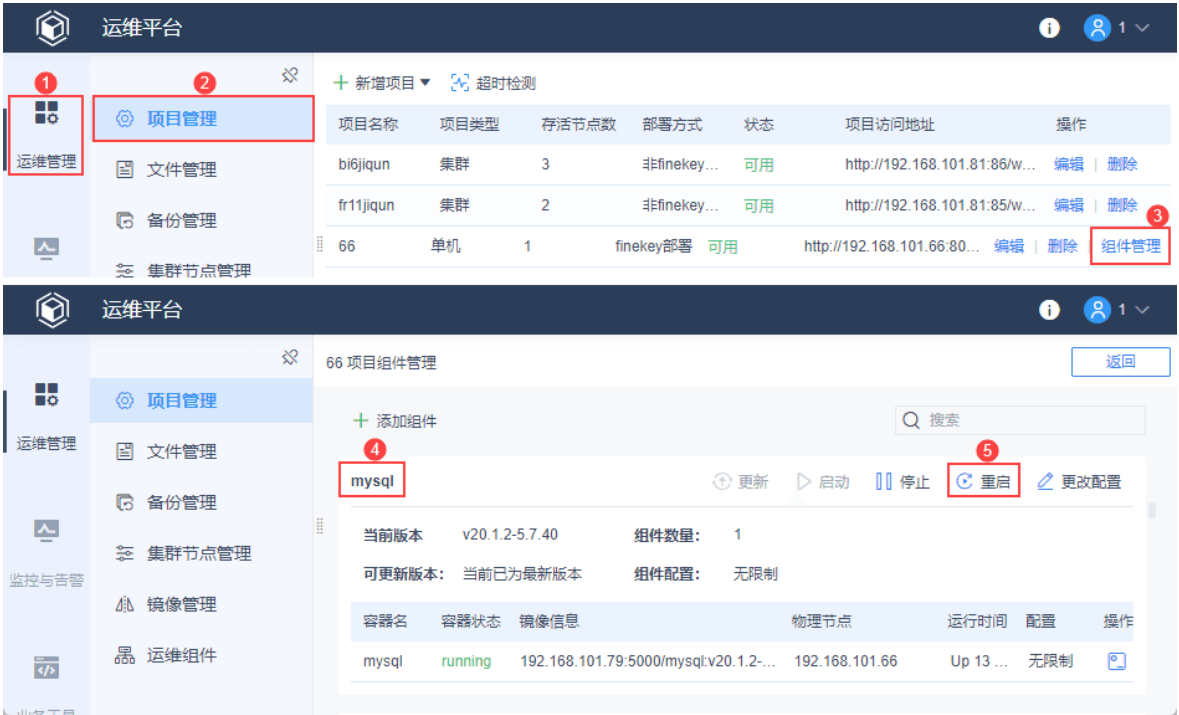


3) 重启 FR/BI 项目组件

管理员登录运维平台, 点击「运维管理>项目管理」, 点击目标项目对应的「组件管理」。

找到 MySQL 组件, 点击「重启」, 即可界面化重启 MySQL 组件。





4) 效果预览

重启后，可查看 max\_allowed\_packet 配置，修改成了指定值

```
mysql> select @@max_allowed_packet;
+-----+
| @@max_allowed_packet |
+-----+
|          146800640 |
+-----+
1 row in set (0.00 sec)

mysql> show variables like 'max_allowed_packet';
+-----+
| Variable_name | Value |
+-----+
| max_allowed_packet | 146800640 |
+-----+
1 row in set (0.01 sec)
```

3. 案例二：修改 FR/BI 组件配置

1) 发现问题

FineBI 数据决策系统默认不支持按照职务分配权限，但是集团权限配置方案中，有一部分需要根据职务分配权限。

因此需要修改 FineBI 工程的「FSConfig.authorizeAttr.postAuthority」配置值为 true。

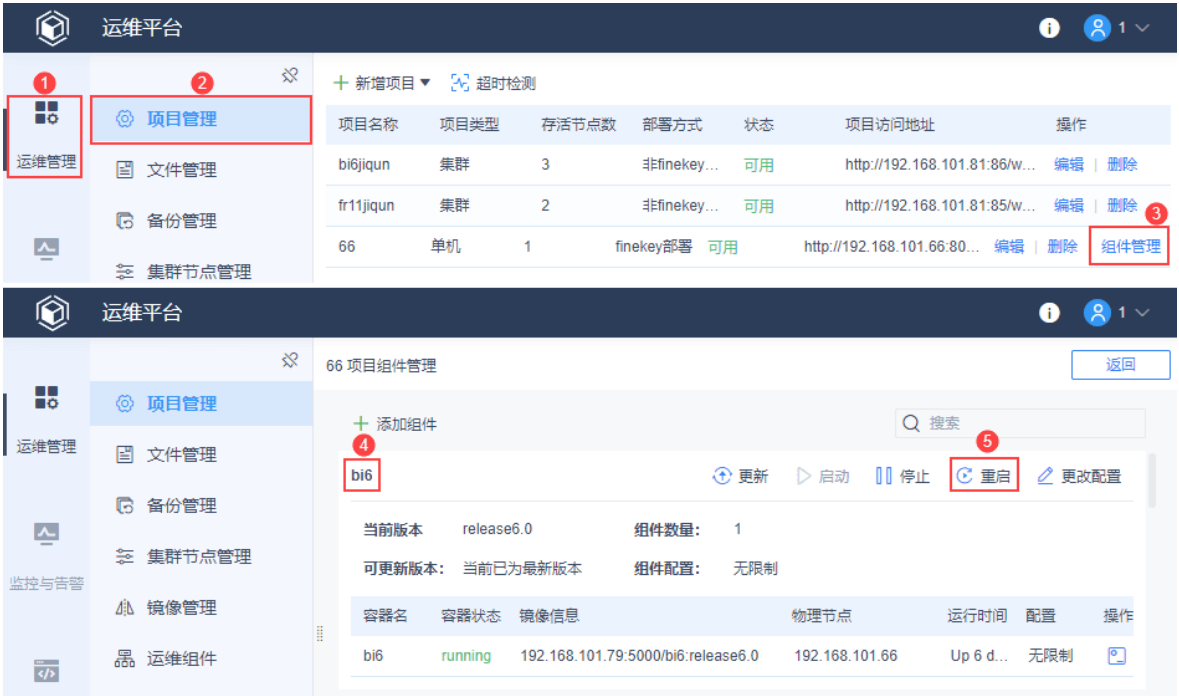
2) 修改 FR/BI 组件配置

管理员登录运维平台，点击「业务工具>业务配置更改」，选择需要修改配置的项目。选择配置方法为「选择参数配置」，找到参数「FSConfig.authorizeAttr.postAuthority」，填入参数值 true。点击「保存」，保存后，会提示重启生效。



3) 重启 FR/BI 组件

管理员登录运维平台, 点击「运维管理>项目管理」, 点击目标项目对应的「组件管理」。  
找到 FineBI 组件, 点击「重启」, 即可界面化重启 FR 组件。



4) 效果预览

重启后, 可查看 FSConfig.authorizeAttr.postAuthority 参数, 修改成了指定值。

4. 注意事项

如果修改 Nginx 配置, 重启过程中, 运维平台会出错, 因为 FineOps 运维平台通过 Nginx 转发访问。

待 Nginx 重启好后, 刷新运维平台查看即可。



## 2.3 场景案例-应用监控与告警

### 1. 概述

#### 1.1 用户痛点

- 一个 FR/BI 集群项目，在多台服务器上分布着各个节点，管理员需要监控每一个节点正常运行
- 一个 FR/BI 项目，不仅只有 FR/BI 组件，还有外接数据库、负载转发、状态服务器、文件服务器等组件，管理员需要监控每一个组件正常运行
- 除了确保以上服务器、工程、组件的正常运行，管理员需要监控他们的配置是否正常，是否有异常现象
- 当出现异常之后，类似于负载异常或组件连接异常，管理员需要能及时收到相关通知信息，而不是事后补救
- 为了分析这些异常，管理员需要登录到这些服务器后台去拿对应的日志，获取很困难

#### 1.2 解决思路

通过一个运维平台可以对接多个节点/应用，同时能够监控多个环境，任意一个节点/应用出现问题都可以统一预警，也能够通过运维平台方便的获取到信息。

- 1) 监控项目正常运行：项目是否可用、节点是否正常、组件是否启动
- 2) 界面化查看项目配置：服务器监控、应用监控、组件监控
- 3) 项目出现异常时预警：企业微信预警、邮件预警、webhook 预警
- 4) 获取日志分析异常：支持在线查看日志分析结果，支持下载日志到本地自行分析

### 2. 案例详情

#### 2.1 添加项目

管理员登录运维平台，点击「运维管理>项目管理」，可添加多个项目进行统一管理。



## 2.2 监控项目正常运行

### 1) 监控项目正常运行

管理员登录运维平台，点击「运维管理>项目管理」，可见各个项目整体的状态。

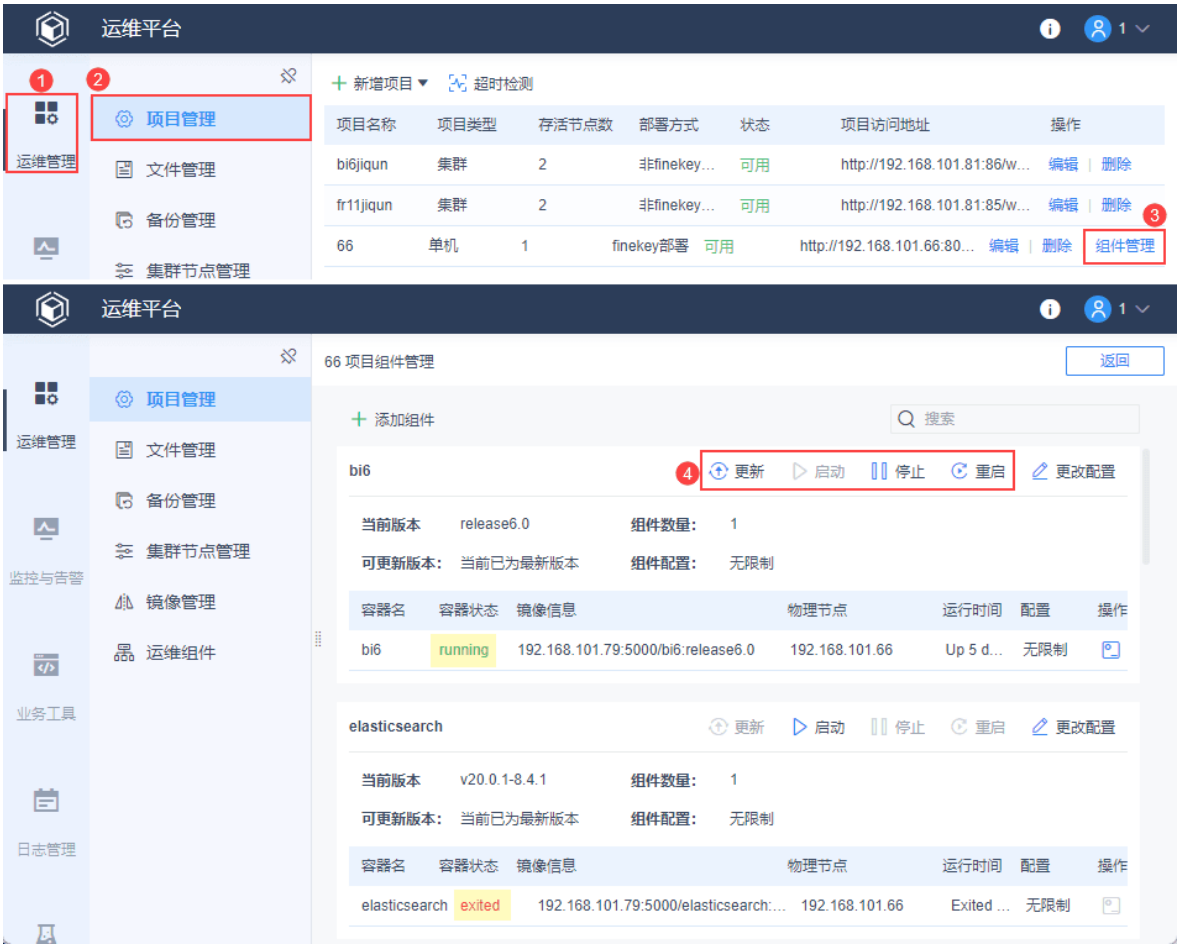
- 可用：该工程可用，至少有一个节点存活着
- 不可用：该工程完全不可用，每个节点都未存活



### 2) 监控工程相关组件正常运行

管理员登录运维平台，点击「运维管理>项目管理>组件管理」，可见该项目中各个组件的状态，支持快速启停组件。

- running：该组件正常运行中
- exited：该组件未启用，需要启动后方可使用



## 2.3 界面化查看项目配置

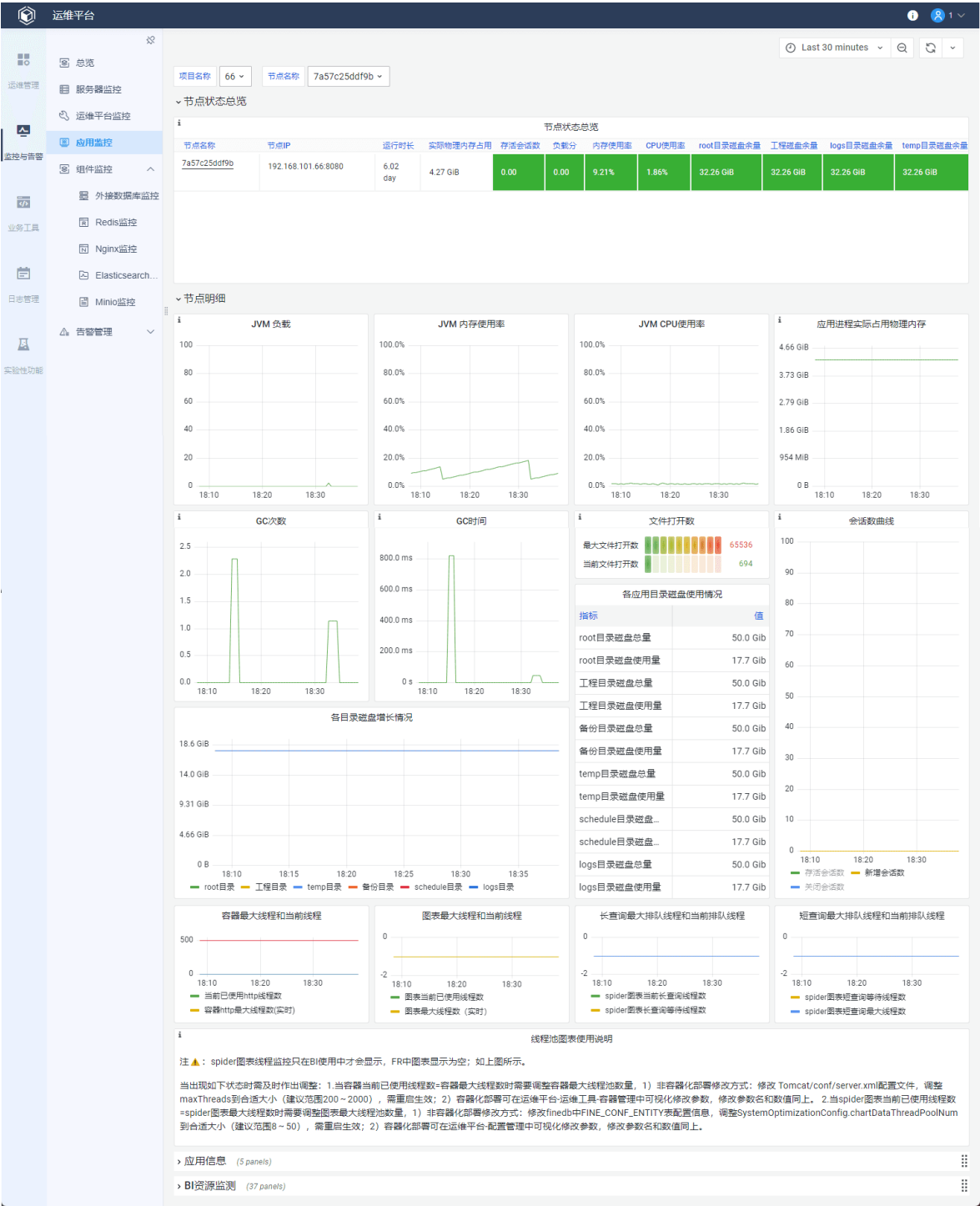
### 1) 展示服务器具体信息

在「监控与告警>服务器监控」界面，监控帆软应用及相关组件运行的服务器的状态，方便及时关注服务器的各项指标情况，避免影响应用的正常运行。



## 2) 展示应用具体信息

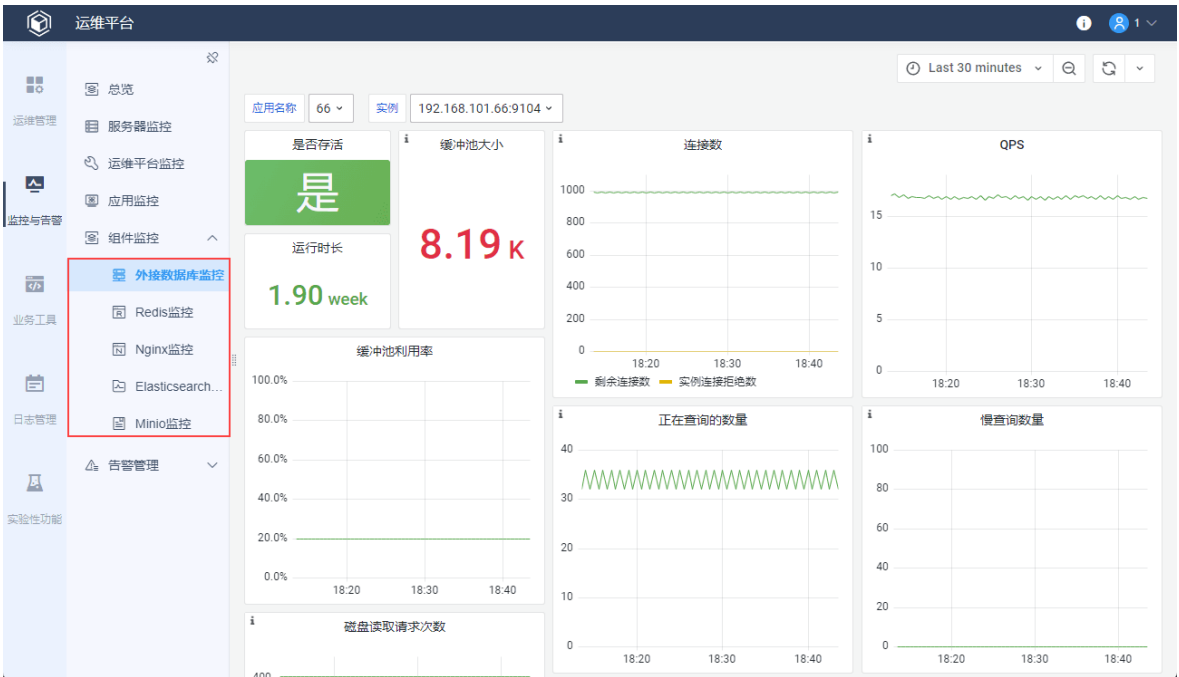
在「监控与告警>应用监控」界面，展示帆软应用的运行情况，可以针对单个节点的运行状态的详细数据进行查询。



### 3) 展示项目其他组件信息

在「[监控与告警>组件监控](#)」界面，展示项目中各类组件的使用信息。

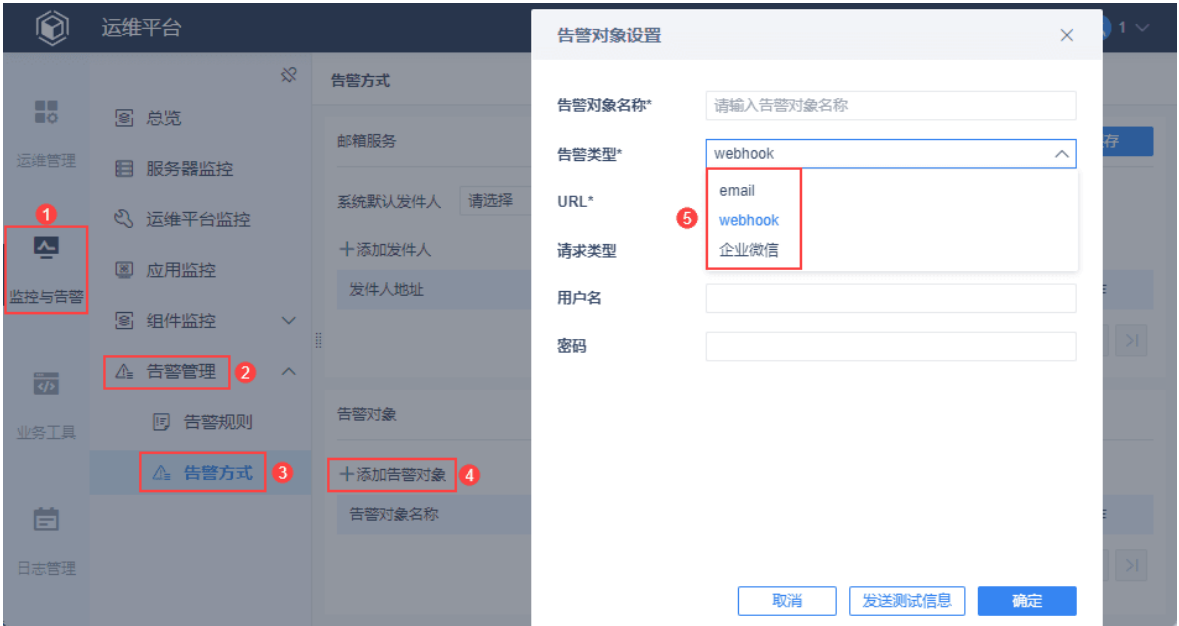
包括：[外接数据库监控](#)、[Redis 监控](#)、[Nginx 监控](#)、[Elasticsearch 监控](#)、[Minio 组件](#)



## 2.4 项目出现异常时预警

当应用出现高负载、宕机风险、磁盘空间不足等情况时，需要及时通知管理员。

- 1) 告警规则：运维平台内置了很多告警规则，基本无需用户自行设置。
- 2) 告警方式：在「监控与告警>告警管理>告警方式」中设置告警对象，支持「email、webhook、企业微信」三种方式。



- 3) 发送告警：配置的收件人邮箱收到告警邮件

[FIRING:1] fine\_alert\_v1 (disk\_space\_root\_alerting 1601692107 449bd88d-0904-69d4-7463-f220066e304c 942ee055a06b /webroot/tomcat Wdliosq7k hC6\_Kwl7z origin\_app\_schedule fanruan1) 外部

发给

3 分钟前

详情

实例信息:

disk\_space\_root\_alerting  
1601692107  
449bd88d-0904-69d4-7463-f220066e304c

详情:

您好, fanruan1系统942ee055a06b节点 root目录剩余空间不足, 仅剩余8.005744934082031GB, 请尽快

2.5 获取日志分析异常

为了分析异常时，管理员需要下载工程的日志。

在「日志管理」中，支持在线查看各个项目各个节点的日志分析结果，支持下载日志到本地自行分析。

运维平台

运维管理

性能堆栈

dump文件

日志管理

性能堆栈

当前运维项目: bijiqun

分析

日志分析

分析记录

项目名	节点名	操作时间	操作者	操作IP	分析结果
bijiqun	e169166b55ad	2023/06/12 17:53:17			查看 删除
bijiqun	e169166b55ad	2023/06/12 17:45:29	1	192.168.5.2	查看 删除
bijiqun	679ee02090ae	2023/06/12 17:45:29	1	192.168.5.2	查看 删除
bijiqun	679ee02090ae	2023/06/12 17:15:50			查看 删除
bijiqun	679ee02090ae	2023/06/12 17:13:35			查看 删除

## 2.4 场景案例-应用资源分配

### 1. 概述

#### 1.1 用户痛点

用户场景：

- 公司中有多个部门，可能存在高并发访问，需要优先确保一些重要业务/用户优先使用资源（例如领导使用，对外大屏，会议使用等）。
- 系统中存在几张模板性能不好（大数据量导出/填报/预览），业务部门不希望对模板进行限制，但又不想影响到其他业务。
- 系统原本的模板正常运行，新增了一些功能业务后，希望先在单独环境中使用一段时间，不要影响到原有业务。
- 公司没有测试系统，测试和生产环境在同一个应用中，需要将他们隔离到不同节点，不互相影响。

需求梳理：

- 保障核心业务性能和稳定性、保障核心用户/模板性能稳定性
- 按照部门设置业务组，业务级别故障隔离，避免不同部门业务相互影响
- 开发发布环境统一，用户权限一致需求，避免测试/新发布模板影响业务运行

#### 1.2 功能简介

功能说明：

- 资源优先级管理：管理员可对系统业务（目录、模板、仪表板、公共链接）和用户（支持按照部门、角色、用户筛选）进行优先级排序，确保高优先级业务优先调用系统资源
- 资源隔离：集群多节点应用，可以根据节点拆分为多个业务组，将不同类型的业务（按照测试/生产、原有/新增、预览/填报）放置到不同业务组中，确保互相不影响。



- 调度管理：使用调度管理功能，可检查系统中不同时间段的调度压力，合理设置业务定时任务的触发时间。
- 组件管理：一个 FR/BI 应用中，有很多组件，运维平台可以限制每个组件对系统 CPU 和内存的占用，防止某个组件大量占用服务器资源，导致其他组件性能差

使用前提：

- 集群节点数不得小于 3 个。其中有 1 个节点需要作为共享节点，不被任何业务组独占；其余至少 2 个节点用于业务划分。
- FineReport11.0、FineBI5.1.22 以上

2. 按业务重要程度划分资源

应用场景

FineReport/FineBI 系统中，有明确的核心业务资源，要求最大程度保证这部分资源的稳定使用。例如：

分类	建议
项目	对外：公司对外展示的大屏模板
	对上层：领导层每日查看的目标看板
	对会议：每周/每月定期会议需要查看的看板
	定期：上班签到、下班打卡、月度工资单、季度绩效单相关模板
用户	部门：对外接触客户的部门
	角色：领导层、股东
	用户：模板制作用户

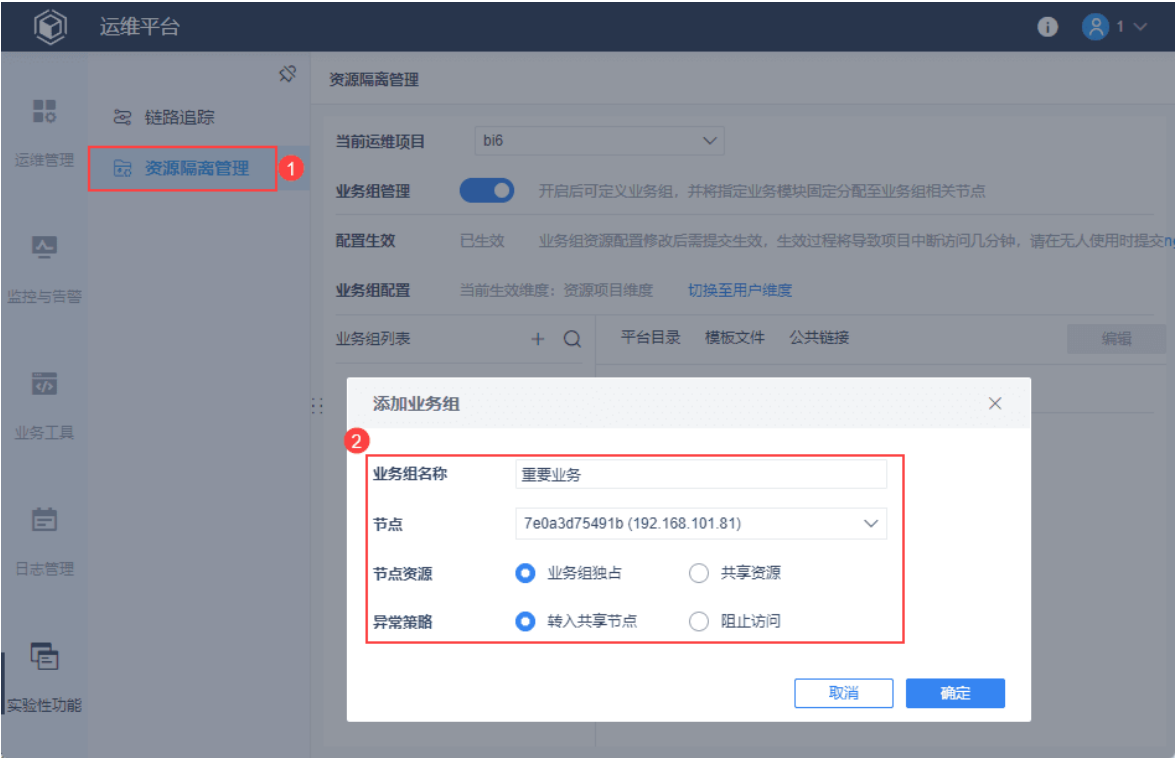
解决方案

1) 设置资源隔离

将核心的模板/用户设置到一个业务组中，业务组分配 2 个或以上的独占节点，业务组配置为：

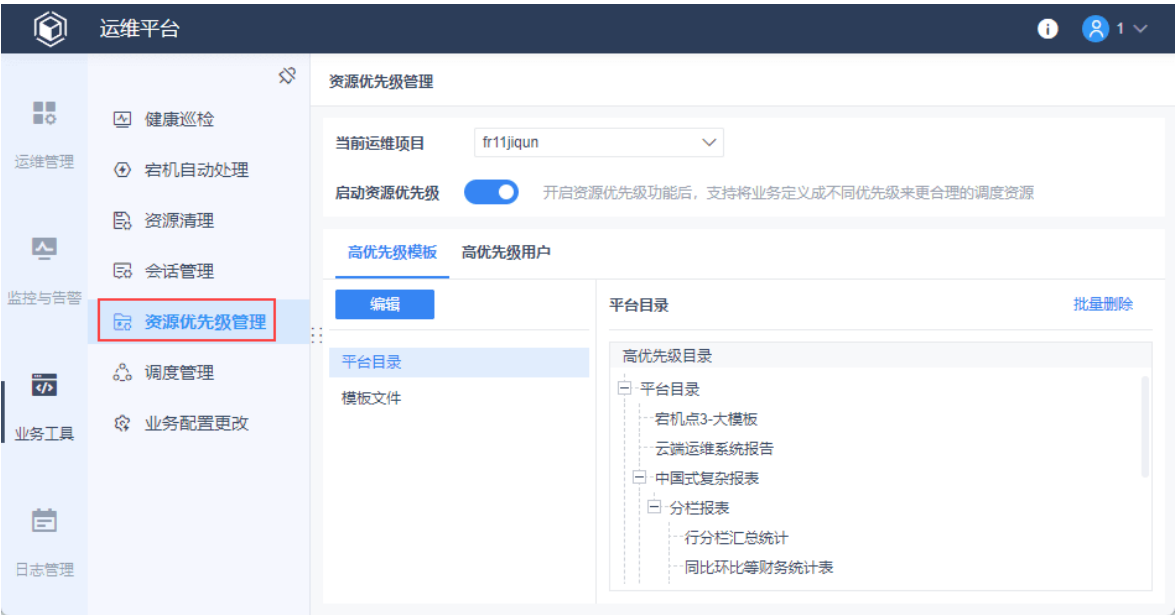
- 节点资源：业务组独占
- 异常策略：转入共享节点

不重要的业务可以整合起来到一个业务组中，或者不分配业务组使用共享节点资源。



## 2) 设置资源优先级

为核心的模板/用户设置高优先级。



## 使用效果

优点	说明
	核心业务组中的模板/用户的请求会被固定转发到指定的几个节点上
资源优先	其他未配置的资源请求转发到其他节点上
	请求级别隔离核心业务和其他业务，避免资源影响

核心业务组中的节点全部异常后，模板/用户会转发到其他节点进行访问，支持高可用

在其他节点，核心模板和用户仍然优先占用资源

资源均衡 核心业务组的节点间负载均衡

### 3. 按业务相关性划分资源

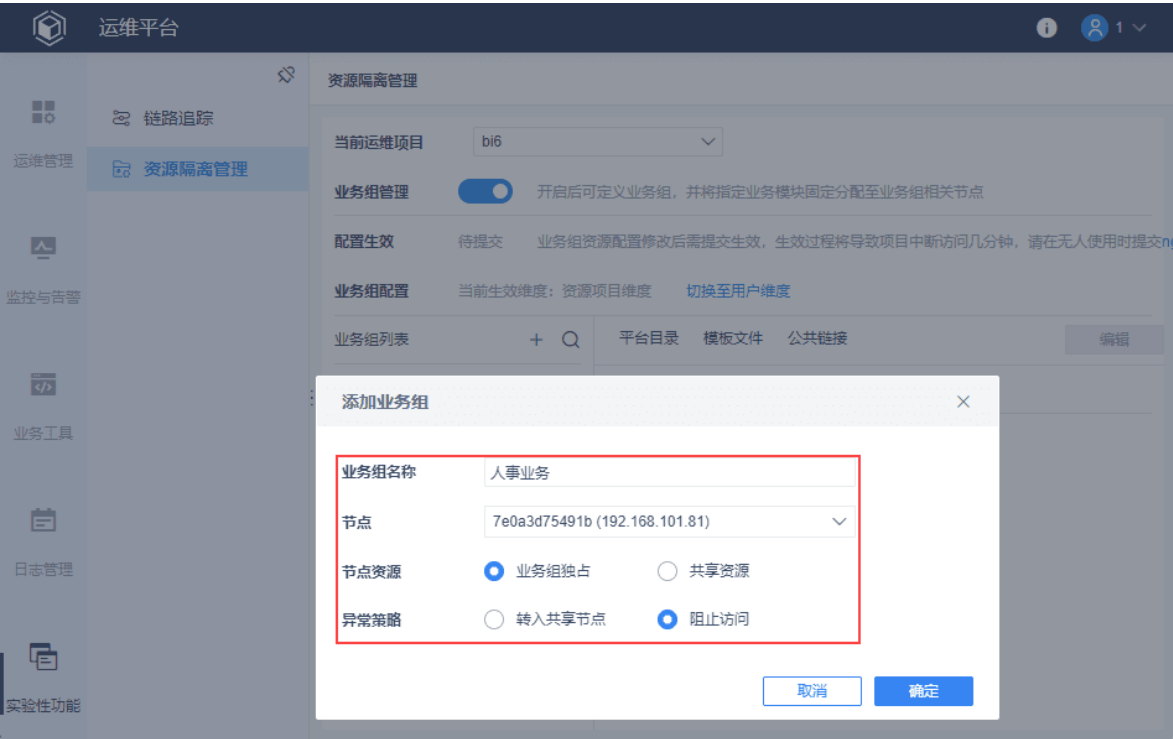
#### 应用场景

FineReport/FineBI 系统中的资源本身就是由多个业务部门使用，每一个业务部门使用的资源互不相关，不希望相互受到影响。

- 同时使用查询和填报，集中填报阶段对于查询会产生资源影响，影响查询性能和稳定性。
- A 和 B 两个部门完全不相关，各自硬件资源角度做资源分割，互不影响。
- 在原有的业务基础上，新增其他的业务，同时不希望影响到原有的业务使用。

#### 解决方案

- 将业务和业务相关用户匹配，同一组业务和用户，隔离到同一个业务组中。
- 这些业务组的配置为：
  - 节点资源：业务组独占
  - 异常策略：阻止访问



## 使用效果

- 1) 不同的业务组汇总的模板/用户请求，会固定转发到指定的节点上。
- 2) 请求级别隔离不同业务，避免资源影响
- 3) 不同业务组硬件级别的故障隔离，影响范围只在自己的业务组内
- 4) 新增业务的情况下，老业务的请求资源完全不受影响，安全稳定的横向扩展。

## 4. 按使用程度划分资源

### 应用场景

当无法根据上述三点划分出具体的隔离模板时，

- 1) 可根据数据决策系统「管理系统>智能运维>平台日志>模板热度」中，被操作总次数最多的模板来判断优先级

定时调度

移动平台

注册管理

智能运维

负载管理

集群配置

备份还原

平台日志

云端运维

网络检测

信息通知

宕机处理

健康巡检

资源异常

磁盘运维

数据连接

目录

我的分析

公共数据

数据开发

管理系统

回收站

访问统计

用户行为

模板热度

性能监控

管理日志

模板报错

下载日志

操作时间

2023-02-01

-

2023-04-03

模板名称

请选择

查询

导出为log

导出为excel

序号	模板名称	被访问资源	查看次数	导出次数	打印次数	总操作次数
1	功能应用/交互操作展示	我的分析/功能演示/FineBI交互操作展示/FineBI交互...	44	0	0	44
2	功能应用/图表组件汇总	我的分析/功能演示/图表组件汇总/图表组件汇总	36	0	0	36
3	五分钟上手 FineBI	我的分析/功能演示/五分钟上手FineBI/五分钟上手...	35	0	0	35
4		分片与冻结.cpt	29	0	0	29
5		行分栏.cpt	28	0	0	28
6	功能应用/如何清晰的展示数据信息	我的分析/功能演示/突出显示/突出显示	23	0	0	23
7	业务应用/库存主题/库存周转率分析	我的分析/数据分析方法/库存周转率分析/库存周...	22	0	0	22
8		TT-51393.cptx	17	0	0	17
9	功能应用/FineReport高级报表/人员变动...	我的分析/功能演示/FineReport高级报表/人员变...	16	0	0	16
10	新手入门/首页导航	我的分析/功能演示/首页导航/首页	15	0	0	15

- 2) 可根据数据决策系统「管理系统>智能运维>平台日志>用户行为」中，总操作次数最多的用户来判断优先级

定时调度

移动平台

注册管理

智能运维

负载管理

集群配置

备份还原

平台日志

云端运维

网络检测

信息通知

宕机处理

健康巡检

资源异常

磁盘运维

数据连接

目录

我的分析

公共数据

数据开发

管理系统

回收站

访问统计

用户行为

模板热度

性能监控

管理日志

模板报错

下载日志

模板业务

操作时间

2023-02-01

-

2023-04-03

操作用户

请选择

查询

导出为log

导出为excel

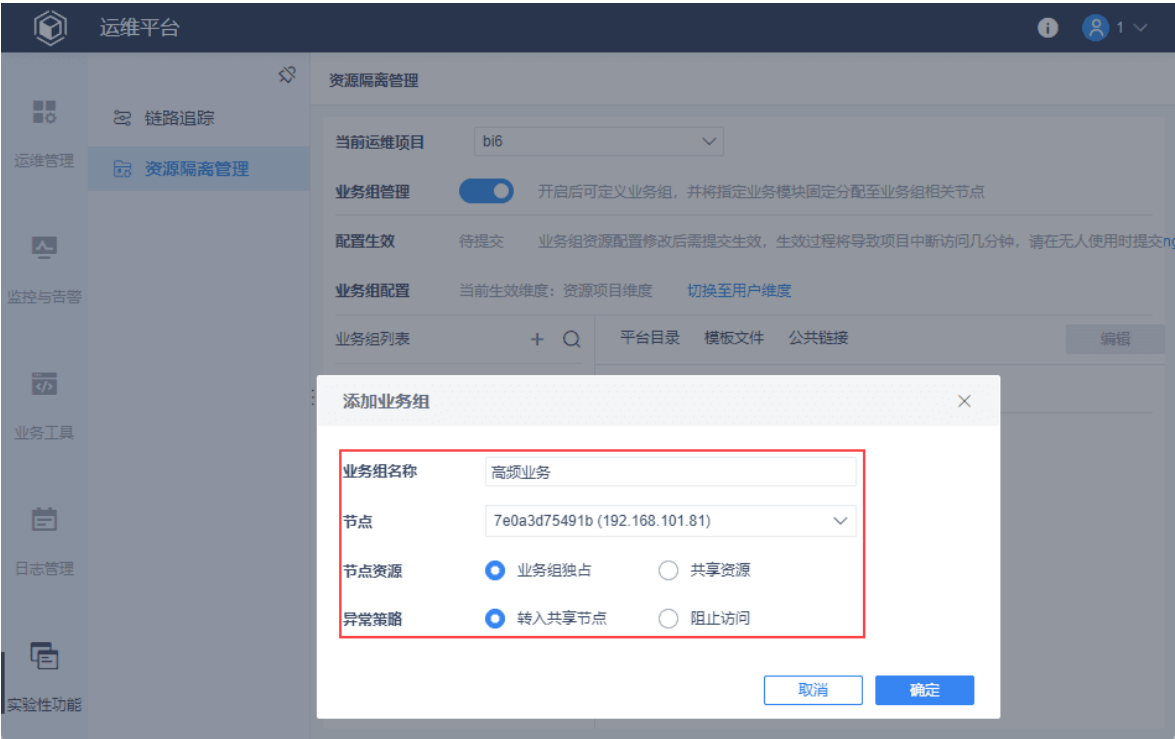
序号	操作用户	查看次数	导出次数	打印次数	总操作次数
1	1(1)	651	0	0	672
2	毛安薇(abby)	110	0	0	110
3	A(A)	33	0	0	33
4	B(B)	22	0	0	22
5	吴姬(Abby)	14	0	0	14
6	次管次管次管	22	0	0	10
7	#英文特殊符号2(#英文特殊符号2)	7	0	0	7

## 解决方案

- 1) 将模板热度中识别出来的高频使用模板设置到一个业务组当中

2) 该业务组的配置为：

- 节点资源：业务组独占
- 异常策略：转入共享节点



使用效果

优点	说明
	高频使用的模板请求会被固定转发到指定的几个节点上
资源优先	其他未配置的资源请求转发到其他节点上
	请求级别隔离核心业务和其他业务，避免资源影响
异常处理	核心业务组中的节点全部异常后依旧会有其他节点进行转发，支持高可用
	核心业务组的节点间负载均衡

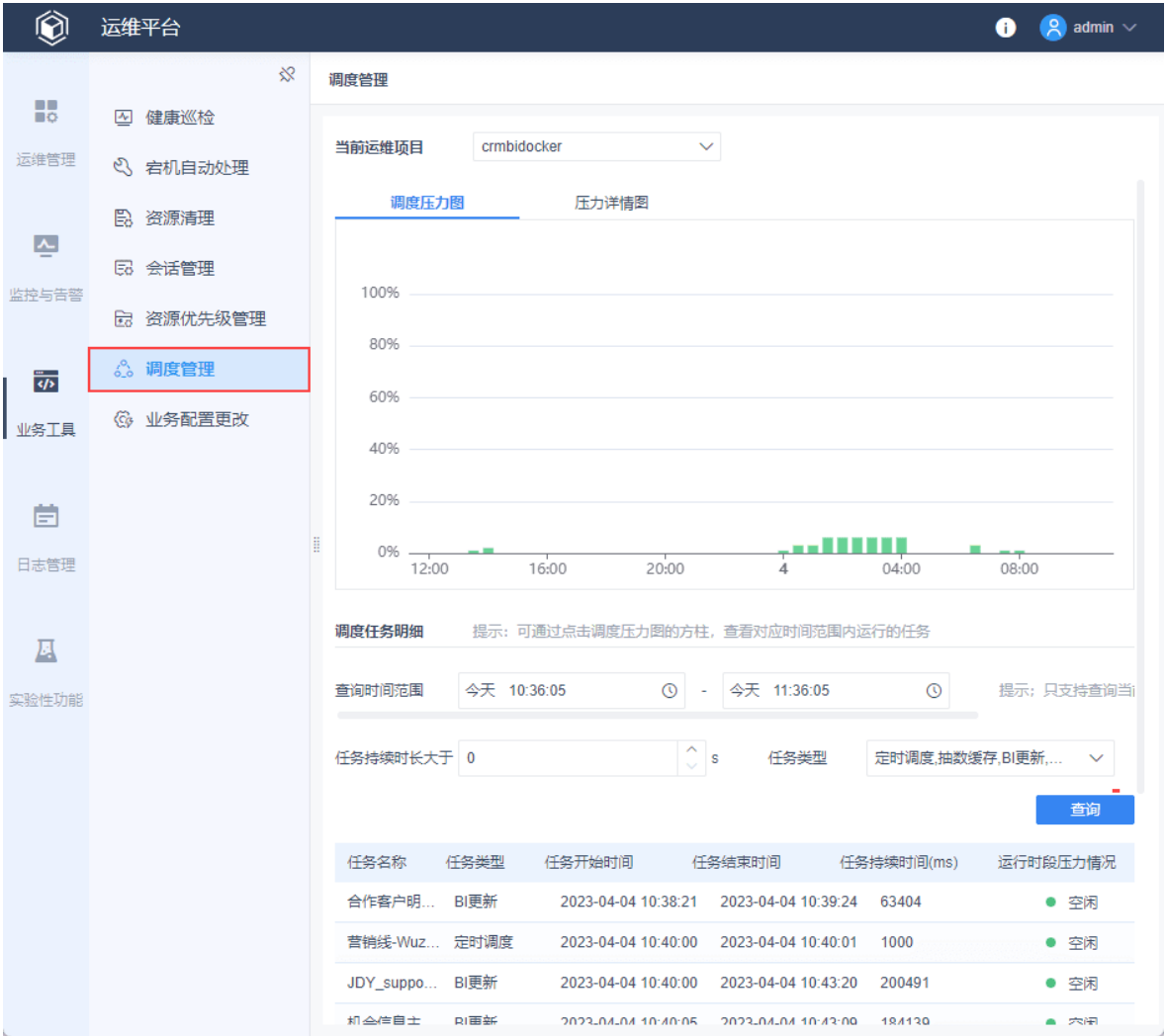
5. 其他功能

调度管理

在新增业务调度任务之前，先查看下系统自带的内置调度任务压力（这些任务包括定期清理日志，定期检测系统情况等）

从调度压力图可以看出来，下午 4 点到凌晨 4 点，系统的调度压力比较低。

管理员设置 BI 更新任务、定时调度任务、抽数缓存任务时，应当尽量安排在此时间段。

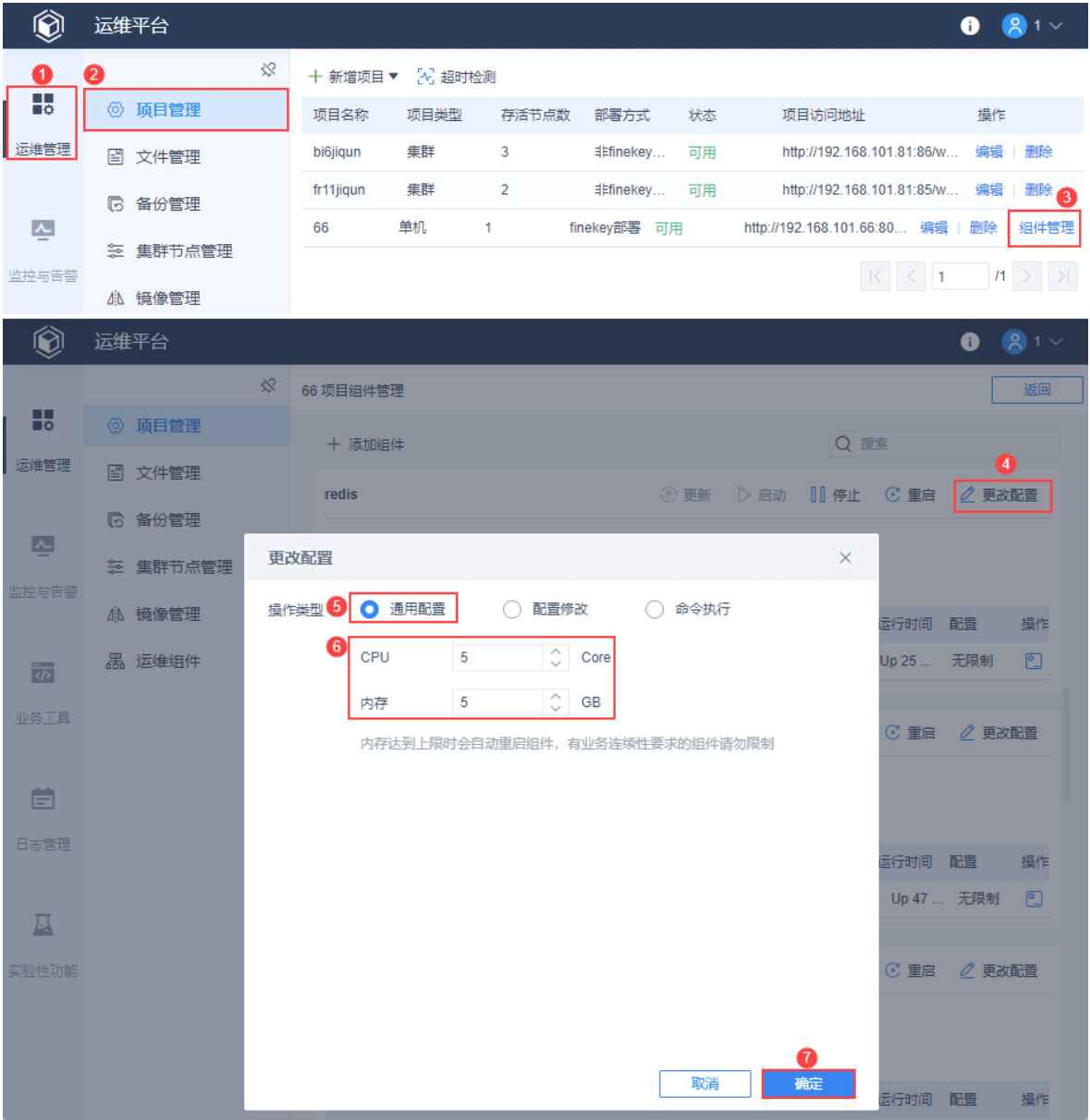


## 组件管理

小明在服务器上部署了一个容器化项目,但是他发现 elasticsearch 组件运行性能较差。经过检查,发现是 redis 组件占据了全部 CPU,导致服务器无法给 elasticsearch 组件提供富裕内存。

管理员登录运维平台,点击「运维管理>项目管理>组件管理」,点击 redis 组件对应的「更改配置>通用配置」,限制 CPU 和内存。

限制生效后,内存达到上限时会自动重启组件。



## 2.5 最佳实践-备份还原

### 1. 概述

#### 1.1 用户痛点

小明同学负责一个 FineReport 工程的运维，为了防止工程出现意外，需要定期对工程进行备份。

FineReport/FineBI 提供了「备份还原」功能，但是在使用该功能的过程中，小明又爱又恨。

- 1) 无法异地备份：工程部署在 A 服务器上，备份的内容也在 A 服务器上，结果小明误删除的时候，把工程和备份都删除了。
- 2) 备份速度缓慢：随着工程中的模板、数据越来越多，应用备份的速度越来越慢了。
- 3) 还原逻辑复杂：JAR 包、模板、配置都是分开备份的，但是大多时候需要将这些内容按照时间点统一备份还原。
- 4) 备份内容不全：一些重要内容不在备份还原范围内，会影响工程的正常运行，例如 classes、resources 等。

#### 1.2 解决思路

针对备份还原的相关诉求，运维平台提供「[备份还原](#)」功能，帮助用户对工程进行异地、整体、快速的备份还原。

#### 1.3 备份内容

在了解具体场景方案之前，用户应当先了解下，一个 FineReport/FineBI 工程中，需要备份的内容有哪些：

备份内容	说明	位置
平台配置	备份内容：工程的数据决策系统配置，包括：使用内置数据库：web	root\WEB-INF\embed\finedb
	系统设置、用户、权限、BI 模板等	
	备份必要性：没有备份的话，工程还原后，平台中的所有用户、权限、目录等配置都会丢失	
报表模板	备份内容：FineReport 的报表模板文件	单机：webroot\WEB-INF\reportlets



	<p>备份必要性：FineReport 最重要的模板文件， 集群：文件服务器\reports</p> <p>没有备份的话，就无法访问任何一张模板</p>	ortlets
	<p>备份内容：工程的 JAR 包</p>	
JAR 包	<p>备份必要性：没有备份的话，工程还原后，工程无法正常运行</p>	webroot\WEB-INF\lib
	<p>备份内容：工程定制/外部引入的 JAR 包</p>	
自定义 JAR 包	<p>备份必要性：没有备份的话，工程还原后，工程无法正常运行</p>	webroot\WEB-INF\customlib
	<p>备份内容：工程安装的插件</p>	
插件	<p>备份必要性：没有备份的话，工程还原后，依赖插件运行的模板和功能将无法正常使用</p>	webroot\WEB-INF\plugins
	<p>备份内容：工程中自定义的一些程序数据集等</p>	
classes	<p>编译好的可执行 class 文件</p> <p>备份必要性：没有备份的话，工程还原后，依赖这些 class 文件的模板，无法正常使用</p>	webroot\WEB-INF\classes
	<p>备份内容：设计器注册文件、模板加密文件</p>	
resources	<p>备份必要性：没有备份的话，工程还原后，通过 lic 方式注册的工程需要重新注册，使用了模板加密会导致所有模板都打不开</p>	单机： webroot\WEB-INF\resources 集群：文件服务器\resources
	<p>备份内容：FineBI 的 Excel 原始文件信息、FineReport 模板备份文件、通用的共享持久化目录</p>	
通用属性配置	<p>备份必要性：工程正常运行所需要的文件，如不拷贝会影响工程正常使用</p>	webroot\WEB-INF\assets
	<p>备份内容：集群的一些配置信息，例如外接数据库配置信息，仅当记录过集群配置信息后才</p>	单机： webroot\WEB-INF\config 集群：文件服务器\config

会出现		fig
备份必要性: 如果工程配置了外接库或开启了		
集群, 没有备份的话, 工程还原后是单机+内		
置, 集群的连接信息都会丢失, 需要重新配置		
外接库		
备份内容: FineDataLink 任务相关的配置文件		
备份必要性: 如不备份, 工程还原后, 会影响		
FineDataLink 任	FineDataLink 任务使用	webroot\WEB-INF\dpwor
务配置		ks
(如不存在该文件夹, 说明工程没有使用 FD		
L 相关功能, 略过)		

2. 最佳实践

对于不同情况下部署的 FineReport/FineBI 工程, 工程的最佳备份还原方案不完全相同。

1) 非容器化部署工程

部署方式说明: 未使用 FineKey 工具/运维平台部署的 FineReport、FineBI 工程。

备份还原方案:

- 本机备份还原: 通过数据决策系统的功能, 可以进行本机备份还原。备份内容包括 JAR 包、模板文件、插件、配置库。
- 异地备份还原: 通过运维平台对接非容器化工程, 可以实现异地备份, 需要手动还原。备份内容包括 JAR 包、模板文件、插件、配置库。
- 集群备份还原: 对于集群组件备份的场景, 基本都需要管理员自行手动备份还原。

2) 容器化部署工程自有配置库

部署方式说明: 使用 FineKey 工具/运维平台部署的 FineReport、FineBI 工程, 且工程的 finedb 配置库是后期自行接入的, 工程使用的集群组件都是用户自行安装配置。

备份还原方案: 运维平台仅能将工程的 JAR 包、插件、集群配置和 classes 备份到运

维平台所在服务器，其余都需要管理员自行处理。

3) 容器化部署工程（推荐）

部署方式说明：使用 FineKey 工具/运维平台部署的 FineReport、FineBI 工程，且工程的 finedb 配置库、集群组件等是部署时全新安装接入的。

备份还原方案：运维平台可以直接实现全部功能功能化备份还原，包括异地备份

2.1 单机工程-本地备份

		容器化部署工程	
备份内容	非容器化部署工程	自有配置库	容器化部署工程
平台配置	工程未配置外接数据库：		
	库：		
	使用数据决策系统的备份还原功能		
	对平台配置进行备份还原	工程配置了外接数据库：用户自行手动备份还原	使用运维平台的备份还原功能，对整体工程进行备份还原
报表模板	工程配置了外接数据库：		
	库：		1) 备份内容包括：
	用户自行手动备份还原		assets reportlets resources dashboards config plugins classes customlib drivers
	使用数据决策系统的备份还原功能	使用运维平台的备份还原功能，对整体工程进行备份还原	2) 其中 assets 下排除了：
JAR 包	对报表模板进行备份还原		temp_attach/default temp_attach/session temp_attach/query temp_attach/restart
	使用数据决策系统的备份还原功能	1) 备份内容包括：	
	对工程 JAR 包进行备份还原	assets reportlets resources dashboards config plugins classes customlib	
	使用数据决策系统的备份还原	drivers	
插件			

	份还原功能	2) 其中 assets 下排除了:
	对插件进行备份还原	temp_attach/default
		temp_attach/session
	用户自行手动备份	temp_attach/query
classes	用户自行手动还原	temp_attach/restart
	工程未配置外接数据库:	
	无需备份	
集群配置	工程配置了外接数据库:	
	用户自行手动备份	
	用户自行手动还原	
	用户自行手动备份	
resources	用户自行手动还原	

2.2 单机工程-异地备份

备份内容	容器化部署工程	
	非容器化部署工程	容器化部署工程
	工程未配置外接数据库:	自有配置库
	运维平台和 FR/BI 工程异地部署（部署在不同服务器上）	运维平台和 FR/BI 工程异地部署（部署在不同服务器上）
	使用运维平台的备份还原功能，即可对平台配置进行异地备份	使用运维平台的备份还原功能，即可对整体工程进行异地备份还原
	用户自行手动异地备份	1) 备份内容包括： assets reportlets resources dashboards config plugins classes customlib
	用户自行手动还原	
	工程配置了外接数据库:	
	无需备份	

	用户自行手动异地备份	drivers
	用户自行手动还原	2) 其中 assets 下排除了:
报表模板	运维平台和 FR/BI 工程异地	temp_attach/default
JAR 包	部署 (部署在不同服务器上)	temp_attach/session
	使用运维平台的备份还	temp_attach/query
	原功能, 即可进行异地	temp_attach/restart
插件	备份	
	用户自行手动还原	
	用户自行手动异地备份	
classes	用户自行手动还原	
	工程未配置外接数据	1) 备份内容包括:
	库:	assets
	无需备份	reportlets
集群配置	工程配置了外接数据	resources
	库:	dashboards
		config
		plugins
		classes
		customlib
		drivers
	用户自行手动异地备份	2) 其中 assets 下排除了:
	用户自行手动还原	temp_attach/default
	用户自行手动异地备份	temp_attach/session
resources	用户自行手动还原	temp_attach/query
		temp_attach/restart

2.3 集群工程-本地备份

容器化部署工程			
备份内容	非容器化部署工程	自有配置库	容器化部署工程
平台配置	用户自行手动备份	用户自行手动备份	使用运维平台的备份还
	用户自行手动还原	用户自行手动还原	原功能, 对整体工程进行
	使用数据决策系统的备	使用数据决策系统的备	备份还原
报表模板	份还原功能	份还原功能	1) 备份内容包括:
	对报表模板进行备份还	对报表模板进行备份还	assets
			reportlets

	原	原	resources dashboards config plugins classes customlib drivers
	备份到文件服务器	备份到文件服务器	
	使用数据决策系统的备份还原功能	使用运维平台的备份还原功能，对整体工程进行备份还原	2) 其中 assets 下排除了：
JAR 包	对工程 JAR 包进行备份还原	备份还原	temp_attach/default temp_attach/session temp_attach/query temp_attach/restart
	<a href="#">备份到文件服务器</a>	1) 备份内容包括： assets reportlets	
	使用数据决策系统的备份还原功能	resources dashboards config plugins classes customlib drivers	
插件	对插件进行备份还原		
	<a href="#">备份到文件服务器</a>		
	用户自行手动备份	2) 其中 assets 下排除了： temp_attach/default temp_attach/session temp_attach/query temp_attach/restart	
classes	用户自行手动还原		
	用户自行手动备份		
集群配置	用户自行手动还原		
	用户自行手动备份	用户自行手动备份	
resources	用户自行手动还原	用户自行手动还原	
			使用运维平台的备份还原功能，对以下集群组件进行备份还原
	用户自行手动备份	用户自行手动备份	外接数据库：MySQL 单机
集群组件	用户自行手动还原	用户自行手动还原	文件服务器：MinIO 单机
			负载均衡：nginx 单机
			状态服务器：Redis 单机

2.3 集群工程-异地备份

备份内容	非容器化部署工程	容器化部署工程	容器化部署工程
------	----------	---------	---------

自有配置库

	运维平台和 FR/BI 工程		
	异地部署（部署在不同		
	服务器上）	用户自行手动异地备份	
平台配置	使用运维平台的备份还	用户自行手动还原	
	原功能，即可对平台配		
	置进行异地备份		
	用户自行手动还原		
报表模板	运维平台和 FR/BI 工程	用户自行手动异地备份	使用运维平台的备份还
	异地部署（部署在不同	用户自行手动还原	原功能，对整体工程进行
JAR 包	服务器上）		备份还原
	使用运维平台的备份还		1) 备份内容包括：
插件	原功能，即可进行异地	使用运维平台的备份还	assets
	备份	原功能，对整体工程进行	reportlets
	用户自行手动还原	备份还原	resources
	运维平台和 FR/BI 工程	1) 备份内容包括：	dashboards
	异地部署（部署在不同	assets	config
	服务器上）	reportlets	plugins
classes	使用运维平台的备份还	resources	classes
	原功能，即可进行异地	dashboards	customlib
	备份	config	drivers
	用户自行手动还原	plugins	
	用户自行手动异地备份	classes	2) 其中 assets 下排除了：
集群配置	用户自行手动还原	customlib	temp_attach/default
	用户自行手动异地备份	drivers	temp_attach/session
	用户自行手动还原		temp_attach/query
resources	用户自行手动异地备份		temp_attach/restart
	用户自行手动还原		
集群组件	用户自行手动异地备份	用户自行手动异地备份	使用运维平台的备份还

用户自行手动还原	用户自行手动还原	原功能，对以下集群组件进行备份还原
		外接数据库：MySQL 单机
		文件服务器：MinIO 单机
		负载均衡：nginx 单机
		状态服务器：Redis 单机

3. 下一步操作

对比完本文各种类型工程的备份还原方案后，推荐进行下一步操作：

推荐步骤	功能说明
	未部署 FineReport/FineBI 工程
<a href="#">部署新项目</a>	可参考本文，使用运维平台容器化部署单机/集群工程

4. 相关推荐

本文主要讲述的是备份还原的场景案例，推荐你进行阅读并了解运维平台的备份还原功能：

文档索引	方法简介
	运维平台提供「备份管理」功能。
<a href="#">备份管理</a>	<ul style="list-style-type: none"><li>支持一键备份，支持自动备份，支持异机备份。</li><li>支持还原备份，支持管理备份版本，支持查看备份失败记录</li></ul>



## 2.6 最佳实践-应用更新

### 1. 概述

#### 1.1 方案简介

为满足客户的新需求以及完善之前版本某些功能的不足之处，FineReport/FineBI 在不断地更新迭代。

本文将带大家了解，不同部署方式下，FineReport/FineBI 小版本升级的最佳方案。

#### 1.2 使用前提

无论何种场景，在对 FineReport/FineBI 进行升级之前，都必须对工程进行备份，尽量对除了状态服务器和负载均衡服务以外的工程进行完整备份。

因为可能存在一些意外情况，导致更新对配置库进行了更改，或部分资源文件出现问题，必须保证升级失败后可以快速回退。

不同场景下的备份升级方案请参考：[最佳实践-备份还原](#)

### 2. 最佳实践

本文简单介绍，对于内网/外网、FineReport/FineBI、单机/集群、容器化部署/非容器化部署等工程，不同情况下的最佳升级方案。

注 1：本文针对的是小版本升级方案（例如 FR11.0.5 升级 FR11.0.13），不涉及跨版本更新（例如 FR10.0 升级 FR11.0）。

注 2：本文不考虑只使用容器化部署但不使用运维平台功能的情况，因此不推荐基于 FineKey 工具更新 FR/BI。

	FineReport 非容器化部署	FineBI 非容器化部署	FR/BI 容器化部署
更新到最新版本	1) 联系官方获取最新/指定版本的 JAR 包	更新到最新版本	1) 管理员登录运维平台，点击「运维管理>项目管理」, 选择待更新应用的「组件管理>更新」, 获取最新版本并更新
单机-外网	1) 管理员登录 FR 数据决策系统，点击「系统管理>智能运维>备份还原>更新升级>立即更新」	2) 手动关闭 BI 应用	2) 更新及启动过程自动化，若
		3) 手动替换 JAR 包	
		4) 手动重启 BI 应用	

2) 升级完成后, 手动重 启 FR 应用	更新失败会自动回滚
<b>更新到指定版本</b>	<b>更新到指定版本</b>
1) 联系官方获取指定 版本的 JAR 包	1) 联系官方获取指定版本的镜 像及镜像版本号
2) 手动关闭 FR 应用	2) 将镜像 push 到运维平台的镜 像仓库中（此处所指的镜像仓 库, 需要提前在「运维平台>运 维管理>镜像管理>全局设置」 中配置好）
3) 手动替换 JAR 包	3) 管理员登录运维平台, 点击 「运维管理>项目管理」, 选择待 更新应用的「组件管理>更新」, 在版本下拉框中输入并新增此 版本
4) 手动重启 FR 应用	4) 更新及启动过程自动化, 若 更新失败会自动回滚
<b>更新到最新版本</b>	<b>更新到指定版本</b>
1) 管理员登录数据决 策系统, 点击「系统管 理>智能运维>备份还 原>更新升级>立即更 新」	1) 联系官方获取最 新版本号触发更新
2) 手动关闭 FR 所有节 点	2) 手动关闭 BI 所有 节点
3) 手动重启 FR 所有节 点	3) 手动替换所有节 点的 JAR 包
<b>更新到指定版本</b>	<b>更新到指定版本</b>
1) 联系官方获取指定 版本的 JAR 包	1) 联系官方获取指定版本 的 JAR 包
2) 手动关闭 FR 所有节 点	2) 手动关闭 BI 所有 节点
3) 手动替换所有节点 的 JAR 包	3) 手动替换所有节点 的 JAR 包
4) 手动重启 FR 所有节 点	4) 手动重启 BI 所有 节点

集群-外  
网

	1) 联系官方获取最	
单机-内网	1) 联系官方获取指定版本的 JAR 包	1) 联系官方获取指定版本的镜像及镜像版本号
	2) 手动关闭 FR 应用	2) 将镜像 push 到运维平台的镜像仓库中（此处所指的镜像仓库，需要提前在「运维平台>运维管理>镜像管理>全局设置」
	3) 手动替换 JAR 包	
	4) 手动重启 FR 应用	
集群-内网		1) 联系官方获取最新/指定版本的 JAR 包
	1) 联系官方获取指定版本的 JAR 包	中配置好)
	2)手动关闭 FR 所有节点	3) 管理员登录运维平台，点击「运维管理>项目管理」, 选择待更新应用的「组件管理>更新」, 在版本下拉框中输入并新增此版本号触发更新
	2)手动关闭 BI 所有节点	
	3) 手动替换所有节点的 JAR 包	
	3) 手动替换所有节点的 JAR 包	
	4)手动重启 FR 所有节点	4) 更新及启动过程自动化，若更新失败会自动回滚
	4)手动重启 BI 所有节点	

## 3. 部署升级 FineOps

## 3.1 部署运维平台

### 1. 概述

#### 1.1 版本说明

运维平台版本	功能变动
V1.5.5	-
V1.5.6	运维平台瘦身，取消内置和安装部分冗余运维组件镜像：loki、promtail、sherlock、nacos-server、skywalking_ui
V1.5.7	elasticsearch、skywalking_oap 组件安装后默认启动
V1.11.0	新增浏览器版本检测提醒
V1.12.0	部署成功提示工程地址，方便用户访问
V1.14.0	若检测挂载磁盘空间不足，可快速更换部署路径，无需手动更改 yaml  调整部分运维组件默认端口，防止与运维项目冲突
V1.15.0	<ul style="list-style-type: none"><li>node_exporter 组件默认端口由 9100 调整为 9101</li><li>ops_agent 组件默认端口由 9070 调整为 9071</li></ul>

#### 1.2 应用场景

FineOps 运维平台是帆软软件推出的，为企业提供帆软应用稳定运行和全方位运维管理的解决方案。

帮助用户解决，管理帆软应用（FineReport、FineBI）的过程遇到一些痛点和挑战，详情请参见：[运维平台简介](#)

#### 1.3 功能简介

本文介绍，如何部署 FineOps 运维平台。

本文主要提供给服务器资源较为紧张的客户，如能提供全新服务器用于部署，可参考：[快速入门](#)

[helpvideo]5513[/helpvideo]

### 2. 准备部署环境

#### 2.1 确认服务器配置条件

部署运维平台的服务器应当满足下表条件：

配置	要求
----	----

建议运维平台**独占**一台服务器，即部署运维平台的服务器上不部署其他工程或内容，  
服务器数量 此时可以确保运维的独立性

如不能提供独立服务器，请至少确保该服务器中，空闲物理内存超过 16G

CentOS 7.3~7.9 版本

系统版本 Ubuntu 18.04.4 及以上版本

redhat 7.6 及以上版本

3.10 及以上

系统内核

最低 4 核

系统架构 仅支持 X86\_64

1) 服务器中存在一个分区剩余可用空间大于 **200G**，最低 **100G**

注：如准备全新磁盘，建议至少 **120G**。因为系统会占用部分空间，导致剩余不足 100G。

剩余空闲磁盘

2) 如对接的运维项目较多，请根据需要对接的项目（FR/BI/FDL）数量准备磁盘，服务器中存在一个分区剩余可用空间≥**项目数量\*50G**

注：可使用命令「df -h」查询磁盘空间，必须要有任一分区满足条件，不可将分区容量加和

剩余空闲物理内存

如运维平台可**独占**一台服务器，建议剩余空闲物理内存 16G 以上，最低 12G

如不能提供独立服务器，请确保该服务器中，剩余空闲物理内存超过 16G

后续需要使用 FineOps 运维平台对接帆软应用（FineReport/FineBI/FineDataLink）

网络要求

请确保 FineOps 运维平台所在服务器、帆软应用所在服务器、外接数据库所在服务器、集群组件所在服务器，网络环境内网互通

之所以需要上表的配置环境，是因为 FineOps 运维平台由很多组件构成。每个组件对应的功能和配置如下表所示：

注：下表列出的为组件正常运行的最低配置要求。推荐配置一般为最低配置的两倍，请根据服务器实际情况自行调整。

类型	组件	组件说明	CPU 使用上限	内存使用上限	内存配置 xmx
运维平台	ops	作用：运维平台容器，对其他容器进行	2Core	-	-

相关	运维				
	影响：必须启用该组件				
	作用：每个运维平台配套一个 ops_agent，与物理节点绑定，用于管理组件，容器化升级也是由该组件完成				
	ops-agent	1Core	-	1G	
	影响：必须启用该组件				
	作用：nginx 服务的容器，提供负载转发服务，所有服务的入口、使用都从这个入口访问				
	nginx	1Core	-	-	
	影响：必须启用该组件				
	作用：用于展示监控仪表板				
	grafana	1Core	1G	-	
	影响：如不启用该组件，「监控与告警」功能中的仪表板无法查看				
	作用：收集 exporter 的数据，提供给监控仪表板				
监控告警	prometheus	1Core	1G	-	
	影响：如不启用该组件，「监控与告警」功能中的仪表板无数据，「监控与告警」功能中的告警规则无法使用				
	作用：告警容器，用于告警规则，修改、触发等				
	alertmanager	1Core	1G	-	
链路追踪	影响：如不启用该组件，「监控与告警」功能中的告警功能无法使用				
	作用：对运维项目中 FineReport、FineBI、外接数据库、业务库和其他外置组件的请求进行采集				
	elasticsearch	1Core	-	2G	
	影响：如不启用该组件，「链路追踪」功能无法使用				

指标 收集	作用：链路追踪的数据上报和分析整理				
	服务端				
	skywalking_oap		1Core	-	2G
	影响：如不启用该组件，「链路追踪」功能无法使用				
	作用：服务器物理节点的 exporter，收集物理服务器的各种指标信息，运维平台所在节点安装一个 node_exporter				
	node_exporter		0.1Core	0.1G	-
	影响：如不启用该组件，「监控与告警」功能中的仪表板无数据				
	作用：nginx 服务的 exporter，收集 nginx 的指标信息				
	nginx_exporter		0.1Core	0.1G	0.1G
	影响：如不启用该组件，「监控与告警」功能中的仪表板无数据				
	作用：elasticSearch 的 exporter，收集 elasticSearch 的指标				
	elasticsearch_exporter		0.1Core	0.1G	-
	影响：如不启用该组件，「监控与告警」功能中的仪表板无数据				
	作用：存放 docker 镜像的仓库				
	影响：若用户自有仓库，无需启用该组件；若用户无镜像仓库，可启用该组件，作为「镜像管理」的仓库				
	registry		1Core	-	-

2.2 确认服务器端口可用性

请确认服务器中，FineOps 运维平台组件的默认端口是否可用。

- 请确认默认的映射端口端口是否已被使用，如已被使用，请安排好其他空闲端口，并在 3.4 节步骤中修改 yaml 文件。



- nginx 的映射端口将作为 FineOps 运维平台访问端口，因此请确认准备使用的 nginx 映射端口是否开放了防火墙、安全组等，确保可连接。

注 1：端口 portMapping 为{映射端口:容器内端口}，其中映射端口可改，容器内端口修改无效。

注 2：检查端口是否被占用的命令：lsof -i:端口

组件	安装默认配置
ops	端口：8081
ops-agent	端口：9071
	root 用户：80
nginx	非 root 用户：8090
grafana	端口：3000
prometheus	端口：9090
alertmanager	端口：9093
	端口：9200
elasticsearch	用户：elastic（不支持修改）
	密码：elasticadmin
skywalking_oap	端口：11800、12800
node_exporter	端口：9101
nginx_exporter	端口：9113
elasticsearch_exporter	端口：9114
registry	端口：5000

2.3 确认 docker 状态

若服务器中未安装 docker，请直接跳过本节内容，安装包中自带 docker。

若服务器中已安装 docker，请确认 docker 版本和运行状态。部署时会将 Dockerfile 和 镜像资源放到已有的 docker 环境下（不会影响已有的其他容器）。

1) 确认 docker 版本：

执行命令：docker version

若安装了 docker，则会返回版本。请确保版本在 20.10.9 及以上。

2) 确认 docker 运行：

执行命令：systemctl status docker

确认 docker 是 running 状态

2.4 确认用户权限

推荐使用 root 用户上传、解压、运行 FineKey 容器化部署工具。

如需使用非 root 用户上传解压 FineKey 容器化部署工具, 至少需要拥有目标文件夹上传权限, 有执行 tar 命令的权限

如需使用非 root 用户运行 FineKey 容器化部署工具, 至少需要拥有执行以下命令的权限: cp、groupadd、gpaswd、systemctl、kill、sysctl

可使用 visudo 命令修改用户的 sudo 权限 (dev 修改为非 root 用户的)

```
root    ALL=(ALL)        ALL
dev                                           ALL=(root)NOPASSWD:
/bin/dockerd,/bin/mkdir,/bin/grep,/bin/sh,/usr/bin/wget,/usr/bin/tar,/bin/rm,/bin/ls,/bin/cd,/bin/cp,/bin/systemctl,/bin/kill,/usr/sbin/sysctl,/usr/bin/gpasswd,/usr/sbin/groupadd,/usr/bin/c
hown,/usr/bin/echo,/usr/bin/sed
```

3. 准备 FineKey 工具

3.1 获取 FineKey 安装包

点击可跳转至下载页下载安装包: [FineOps 运维平台下载](#)

有三种类型的安装包, 用户可根据服务器环境和自身诉求自行选择:

安装包类型	说明
在线版安装包	1) 服务器可访问外网, 支持下载运维组件镜像, 安装包体积小
	2) 支持从云端下载 FineReport/ FineBI 项目组件镜像, 可通过 FineOps 可视化部署工程
	3) 可对接已有 FineReport/ FineBI 工程
纯净版离线安装包	1) 服务器为内网, 内置全部运维组件镜像, 安装包体积大
	2) 未内置 FineReport/ FineBI 项目组件镜像, 无法通过 FineOps 可视化部署工程
	3) 用于对接已有 FineReport/ FineBI 工程
全量版离线安装包	1) 服务器为内网, 内置全部运维组件镜像, 安装包体积大
	2) 内置 FineReport/ FineBI 项目组件镜像, 可通过 FineOps 可视化部署工程
	3) 可对接已有 FineReport/ FineBI 工程

注: 如希望使用在线版安装包, 且只开放指定外网网址, 请确保镜像仓库所在服务器支持访问以下外网地址:

(运维平台自带的镜像仓库与运维平台在同一服务器下)

功能	网址
获取镜像/更新运维平台 /更新项目组件	域名：registry.cn-hangzhou.aliyuncs.com，端口：443
	域名：fineops-registry.cn-hangzhou.cr.aliyuncs.com，端口：443
获取更新信息	四选一，有一个能联通即可：
	https://cloud.fanruan.com/config
	https://cloud.fanruan.com:8443/config
	http://cloud.fanruan.com/config
	http://cloud.fanruan.com:8090/config

### 3.2 上传 FineKey 安装包

获得工具包后，需要将它上传到 Linux，上传文件的方法请自行查阅相关资料。

本文示例是将工具包上传到了 CentOS 7.6.1810 的 /home 路径下，如下图所示：

```
[root@localhost home]# cd /home
[root@localhost home]# ls
finekey-operation.tar.gz
[root@localhost home]# ls
```

### 3.3 解压 FineKey 安装包

- 1) 先使用 cd 命令进入工具包所在文件夹下。
- 2) 根据下载的安装包类型，键入命令进行解压：

安装包类型	解压语句
在线版安装包	tar -xvf finekey-operation-online.tar
纯净版离线安装包	tar zxvf finekey-operation-lite.tar.gz
全量版离线安装包	tar zxvf finekey-operation-all.tar.gz

```
[root@localhost home]# cd /home
[root@localhost home]# ls
finekey-operation.tar.gz
[root@localhost home]# cd /home
[root@localhost home]# tar zxvf finekey-operation.tar.gz
finekey/
finekey/bin/
finekey/bin/finekey
finekey/bin/fine_net_test
finekey/conf/
finekey/conf/finekey.yaml
finekey/conf/finekey.push.yaml
finekey/resources/
finekey/resources/images/
```

解压后的文件如下表所示：

文件夹/文件	文件说明
finekey/bin	运行 finekey 的可执行文件
finekey/conf	finekey.yaml 新部署运维平台配置文件
	finekey.push.yaml 拉取镜像配置文件
finekey/resources	资源文件，镜像包放在其中 images 文件夹中

### 3.4 配置 finekey.yaml 文件

yaml 文件指的是工具解压目录下 conf 文件夹里的 finekey.yaml，该文件是控制后面整

个自动部署流程的。

- 如完全采用默认部署配置（包括安装目录、组件端口，仓库配置等），用户可不对 yaml 文件做任何配置，可以直接在当前宿主机本地部署。
- 如需更改运维平台相关配置、组件占用端口等，可以打开 yaml 文件自行配置，yaml 文件的配置需要在启动 FineKey 之前完成。

1) 文件内容说明

配置	配置项	说明
基础配置		值为部署的 FineOps 运维平台相关配置
	nodes	包括：ip, port, user, password  如果 ssh 方式为密钥，则将 password 配置注释，新增 keypath: 密钥绝对路径/  根挂载目录  请确保挂载目录所在分区，磁盘空闲空间≥100G  默认安装目录为~/data，即安装用户主目录下的 data 文件夹  例如使用 root 用户部署，则~/data 代表的目录为/root/data
	repo	1) 全新安装一个 registry 镜像仓库，可指定仓库端口，默认 5000  2) 如已有镜像仓库，可输入指定仓库的 url、username、password、ssl
	online	是否在线安装：  false：离线安装  true：在线安装
		不建议修改该配置项，该配置项与下载的安装包类型有关，详情请参见 3.1 节
组件配置		components 下，配置的是 2.1 节列举的待安装的组件
		每个组件都可以自定义端口，参数为：portMapping: {访问用的端口: 镜像中国
		注 1：安装的组件和默认端口如 2.2 节，请排查端口是否可用，是否开放，并  文件中的端口。  注 2：端口 portMapping 为 {映射端口:容器内端口}，其中映射端口可改，容器  改，不要删除。

注 3: 如果修改了 components 下的组件配置, 需要将 components 和组件前面  
掉

例如修改 ops 的端口 8081 为 8088

- 修改前:

```
#组件端口配置
#components:
#   nginx:
#       portMapping: {80: 8080}
#
#   ops:
#       portMapping: {8081: 8080}
```

- 修改后:

```
#组件端口配置
components:
#   nginx:
#       portMapping: {80: 8080}
#
#   ops:
#       portMapping: {8088: 8080}
```

2) 文件示例

```
node:
  ip: localhost          #机器 ip
  port: 22               #机器 ssh 端口
  user:                  #节点 ssh 用户
  password:              #节点 ssh 密码
dataRootPath: ~/data
repo:
  port: 5000             #待安装的 registry 仓库端口
  #指定已有 repo
  url:                   #已有 repo 的 url
  username:              #已有 repo 的用户名
  password:              #已有 repo 的密码
  ssl: true              #已有 repo 是否安全仓库

online: false           #在线安装
#组件端口配置
#components:
#   nginx:
#       portMapping: {80: 8080}
#
#   ops:
#       portMapping: {8081: 8080}
#
#   ops_agent:
#       portMapping: {9070: 9070}
```

```
#
# grafana:
#   portMapping: {3000: 3000}
#
# prometheus:
#   portMapping: {9090: 9090}
#
# alertmanager:
#   portMapping: {9093: 9093}
#
# nginx_exporter:
#   portMapping: {9113: 9113}
#
# skywalking_oap:
#   portMapping: {11800: 11800, 12800: 12800}
#
# elasticsearch:
#   portMapping: {9200: 9200}
#   password: elasticadmin
#
# elasticsearch_exporter:
#   portMapping: {9114: 9114}
```

## 4. 部署操作

### 4.1 启动部署

1) 使用 `cd` 命令进入工具解压目录下 `bin` 文件夹

2) 键入如下命令执行自动化部署：

```
./finekey
```

注：请严格按照文档所述命令执行，不要自行变更为 `sudo ./finekey` 之类的命令。否则

可能导致部署失败。

3) 整个部署过程需要花费一段时间，请耐心等待

```
[root@localhost bin]# cd /home/finekey/bin/
[root@localhost bin]# ./finekey
INFO : 2023/03/28 11:33:05 begin install use config: /home/finekey/conf/finekey.
yaml
INFO : 2023/03/28 11:33:05 use default ip:192.168.101.53
INFO : 2023/03/28 11:33:05 192.168.101.53 :use default user:root
INFO : 2023/03/28 11:33:05 deploymentType:2
INFO : 2023/03/28 11:33:05 begin check config valid!
INFO : 2023/03/28 11:33:05 The deployment mode is offline
INFO : 2023/03/28 11:33:05 192.168.101.53 :use local exec
INFO : 2023/03/28 11:33:05 over check config valid!

INFO : 2023/03/28 11:33:05 begin check net and port~
INFO : 2023/03/28 11:33:05 begin start server and bind ports in all nodes~
INFO : 2023/03/28 11:33:06 192.168.101.53 :time: 2023-03-28/11:33:06 +0800
INFO : 2023/03/28 11:33:06 No migration related configuration

INFO : 2023/03/28 11:33:06 192.168.101.53 :begin install docker~
INFO : 2023/03/28 11:33:06 finekey home:/home/finekey
INFO : 2023/03/28 11:33:08 unzip docker success!
INFO : 2023/03/28 11:33:09 copy docker to /usr/bin success!
INFO : 2023/03/28 11:33:10 docker daemon is starting~
INFO : 2023/03/28 11:33:13 docker daemon start success! docker ps:
```

注 1：如果部署过程中出现报错，请参考 [容器化部署常见问题](#)，查看有无对应的解决

方案。

请注意~/data 目录不代表/data 目录，而是对应用户主目录下的/data 目录。例如使用 root 用户部署，则~/data 代表的目录为/root/data

注 2：如果部署过程中，检测到挂载目录磁盘可用空间不足 100G，会提示「The current path capacity check failed, Please re-enter the path」

用户无需退出部署环节，手动修改 yaml 文件。可直接输入满足条件的目录，继续部署。

```
[root@k8s ops]# cd /home/ops/finekey/bin/
[root@k8s bin]# ./finekey
INFO : 2023/11/27 16:22:02 begin install use config: /home/ops/finekey/conf/finekey.yaml
INFO : 2023/11/27 16:22:02 localhost :use local exec
ERROR: 2023/11/27 16:22:04 node localhost path ~/data disk available capacity:30658212, less
then 100G
INFO : 2023/11/27 16:22:04 The current path capacity check failed, Please re-enter the path
Enter the folder path and press ENTER to confirm, or enter q then ENTER to exit:
```

### 4.2 完成部署

当出现 deployment success! bye~ 提示时，表示已经成功部署。

在该行提示中，会展示运维平台的工程地址，可复制，用于下一步访问。如下图所示：

注：地址中的端口，是部署时设置的 Nginx 组件占用的端口。

- 若为 root 用户部署，默认端口为 80，运维平台访问地址中可以省略端口，直接用 IP 访问，访问地址为 http://ip/ops/decision
- 若为非 root 用户部署，默认端口为 8090，访问地址为 http://ip:8090/ops/decision
- 若 finekey.yaml 文件中，手动修改了 Nginx 端口，访问地址为 http://ip:Nginx 端口/ops/decision

```
INFO : 2023/10/31 14:48:24 192.168.101.198 :begin install ops_agent
INFO : 2023/10/31 14:48:24 docker image 192.168.101.198:5000/ops_agent:v1.12.0.1
exist
INFO : 2023/10/31 14:48:27 192.168.101.198 :begin install skywalking_oap
INFO : 2023/10/31 14:48:28 docker image 192.168.101.198:5000/skywalking_oap:v20.
1.4.3-9.1.0 exist
INFO : 2023/10/31 14:48:31 deployment success The access address is: http://19
2.168.101.198:80/ops/decision . bye~
[root@k8s bin]#
[root@k8s bin]#
```

### 4.3 访问工程


部署完成后，服务会自动启动，此时可以在浏览器地址栏输入上一步获取的工程地址，访问 FineOps 运维平台。

#### 1) 选择浏览器


请使用以下版本的浏览器访问运维平台， 否则可能导致界面展示异常。

浏览器类型	推荐版本
Chrome	88 以上版本
Edge	88 以上版本
Safari	14 以上版本
Firefox	78 以上版本
IE	不推荐

如使用不符合版本要求的浏览器进行访问，部分界面会出现弹窗提醒「检测到当前浏览器版本过低，可能导致部分组件显示异常，建议升级浏览器版本」。



检测到当前浏览器版本过低，可能导致部分组件显示异常  
建议升级浏览器版本



当前版本：71.0.3578.98 (MacIntel)  
最低版本要求: 88

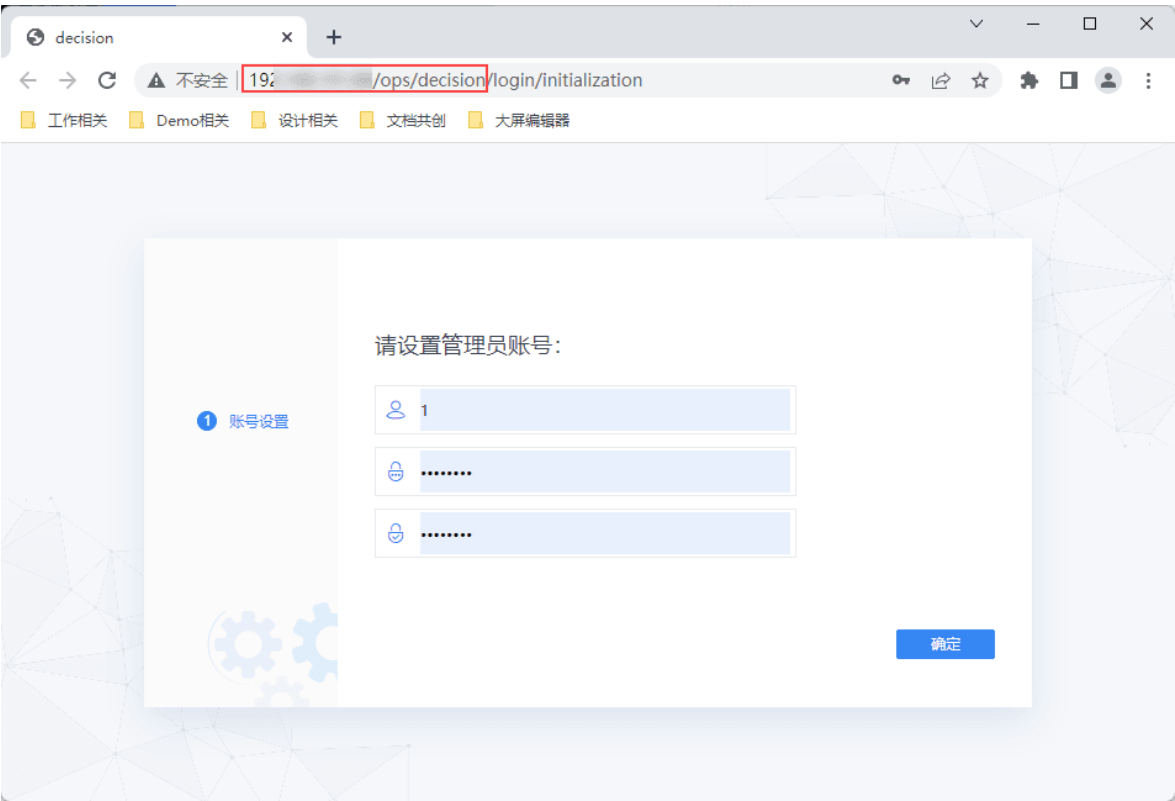
不再出现此提示

我知道了

2) 访问工程

在浏览器地址栏输入上一节获取的工程地址，访问 FineOps 运维平台。

例如本文示例，浏览器地址栏键入：`http://IP/ops/decision`，即可访问运维平台，如下图所示：



5. 下一步操作

当你完成本文「部署运维平台」操作后，推荐进行下一步操作：



推荐步骤	功能说明
<a href="#">部署新项目</a>	若您未部署 FineReport、FineBI、FineDataLink 工程 可通过该功能快速部署工程并接入运维平台
<a href="#">接入已有项目</a>	若您已部署好 FineReport、FineBI 工程 可通过该功能将已有工程接入运维平台

## 3.2 登录运维平台

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.5.5	-

#### 1.2 功能简介

本文将介绍，运维平台部署完成后，用户第一次登录运维平台时的操作流程，以及用户相关配置。

### 2. 初始化配置

一个运维平台，只允许设置一个管理员，不可增加其他用户，该管理员默认可使用所有功能。

只有第一次打开运维平台时，需要对运维平台进行初始化配置。以后再打开运维平台，直接进入登录页面。

#### 2.1 打开运维平台

##### 1) 选择浏览器

请使用以下版本的浏览器访问运维平台，否则可能导致界面展示异常。

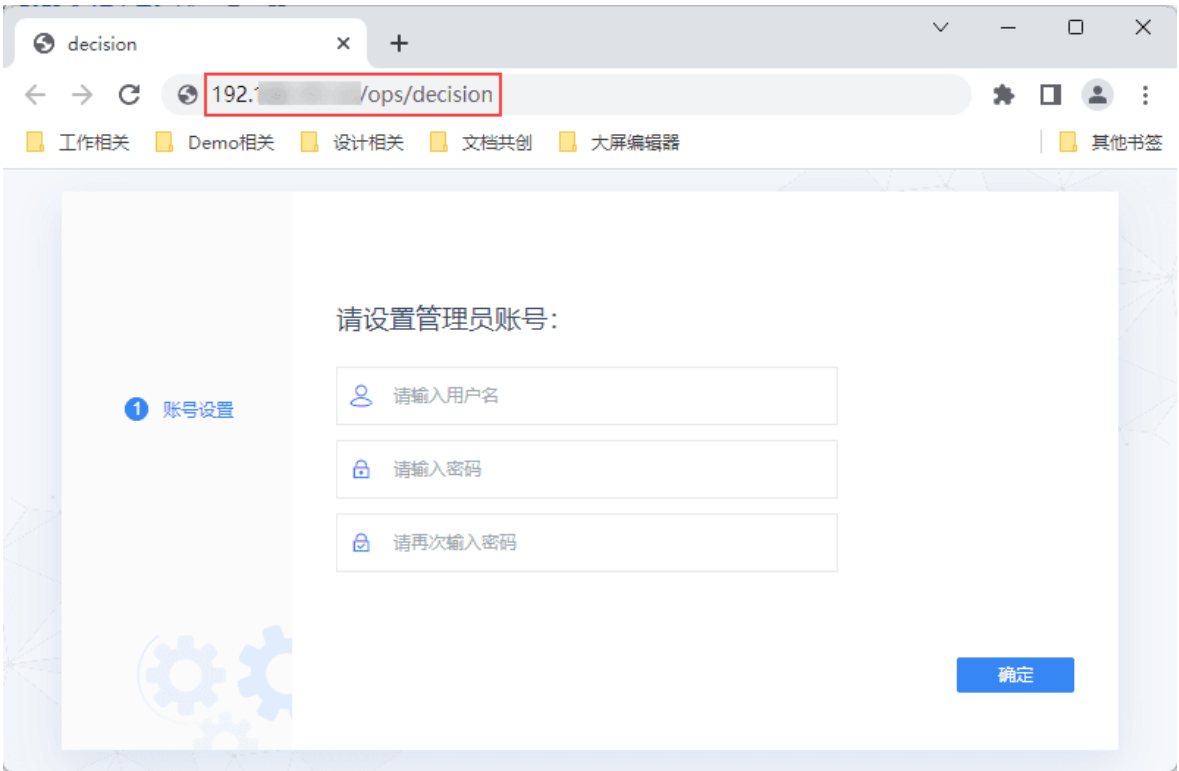
浏览器类型	推荐版本
Chrome	88 以上版本
Edge	88 以上版本
Safari	14 以上版本
Firefox	78 以上版本
IE	不推荐

如使用不符合版本要求的浏览器进行访问，部分界面会出现弹窗提醒「检测到当前浏览器版本过低，可能导致部分组件显示异常，建议升级浏览器版本」。

##### 2) 访问工程

容器化部署完成之后，在浏览器地址栏输入：`http://{ip}/ops/decision`（请把`{ip}`换成自己的主机 IP），即可打开运维平台初始化配置页，如下图所示：

注：由于默认 Nginx 端口为 80，所以运维平台访问地址中才能省略端口，直接用 IP 访问。  
若为非 root 用户部署，默认端口为 8090，访问地址为 `http://ip:8090/ops/decision`  
若 `finekey.yaml` 文件中，手动修改了 Nginx 端口，访问地址为 `http://ip:Nginx 端口/ops/decision`



## 2.2 设置管理员账号

在初始化配置页面中设置管理员账号的用户名和密码，点击「确定」，点击「下一步」按钮。如下图所示：

注：密码需要包含小写字母、数字，且长度超过 8 位，否则在登录时会强制修改密码。



请牢记您设置的用户名和密码。这里设置的超管密码，原则上无法重置，请千万不要忘记密码！



### 2.3 登录运维平台

配置好数据库后，在登录页面输入设置好的管理员账号和密码，点击「登录」。如下图所示：



注：若密码强度低，不包含小写字母、数字，或长度不超过 8 位，在登录时会强制修改密码。如下图所示：



3. 用户设置

3.1 账号设置

管理员登录运维平台后，点击右上角的用户名，点击「账号设置」，可修改姓名、密码、绑定邮箱、绑定手机。如下图所示：



支持的修改项和注意事项如下表所示：

修改项	说明
姓名	点击「姓名」右侧的「修改」按钮，输入新的姓名，点击右上角显示的就是新的姓名。 注：此处改的是展示的姓名，而非登录账号名。

手机 点击手机后面的「绑定/修改」，输入有效手机号码，点击「完成」，手机号绑定/修改成功。

注：支持绑定中国大陆、中国台湾、中国香港、土耳其、韩国、日本、新加坡、马来西亚手机号。

台湾，韩国，日本，马来西亚等国家和地区，区号后、手机号前都不需要加 0 。

邮箱 点击邮箱后面的「绑定/修改」按钮，输入有效邮箱，点击「完成」，邮箱绑定/修改成功。

注：所添加邮箱名中支持包含符号‘#’和‘&’。

密码 在修改密码时需要输入旧密码作为更新密码的验证。

3.2 退出登录

管理员登录运维平台后，点击右上角的用户名，点击「退出」按钮，退出运维平台，直接跳转至登录页面，如下图所示：



3.3 上次登录信息

管理员登录运维平台后，会显示上一次登录的时间、IP 地址、地点，如下图所示：



## 3.3 升级运维平台

### 3.3.1 外网升级运维平台

#### 1. 概述

##### 1.1 版本

运维平台版本	功能变更
V1.5.5	-
V1.5.6	运维平台更新成功后，自动进行运维组件更新操作
V1.9.0	在线升级检测 docker 引擎版本，不符合要求则提示使用手动升级方案，以防由于 docker 引擎版本过低导致功能不可用
V1.11.0	新增新版本更新强提醒

##### 1.2 功能简介

为满足客户的新需求以及完善之前版本某些功能的不足之处，FineOps 运维平台在不断地更新迭代。

外网用户，可使用运维平台内置的一键升级功能，对运维平台进行更新升级。

注 1: 内网用户，推荐借助 FineKey 工具升级运维平台，详情请参见：[内网使用 FineKey 升级运维平台](#)

注 2: 升级运维平台前，可查阅近期功能变更，详情请参见：[运维平台更新日志](#)

#### 2. 示例

##### 2.1 使用前提

如需使用在线升级功能，必须确保运维平台对应的镜像仓库所在服务器支持访问以下外网地址。否则无法获取最新版本进行升级。


功能	网址
获取镜像/更新运维平台/更新项目组件	域名：registry.cn-hangzhou.aliyuncs.com，端口：443
	域名：fineops-registry.cn-hangzhou.cr.aliyuncs.com，端口：443
获取更新信息	四选一，有一个能联通即可：
	https://cloud.fanruan.com/config
	https://cloud.fanruan.com:8443/config
	http://cloud.fanruan.com/config
	http://cloud.fanruan.com:8090/config

##### 2.2 新版本提醒

当存在新版本时，管理员登录运维平台，会自动弹出「发现新版本」弹窗，点击「查看更新」即可跳转到更新入口。





或者点击右上角「」按钮，即可查看当前运维平台版本和检查是否有新版本。

提示	说明
有新版本，查看更新	存在更新版本的运维平台，可点击查看更新日志
当前已是最新版本	当前已是最新版本的运维平台
	当前运维平台无法连接外网，无法使用在线升级功能
网络连接失败	推荐借助 FineKey 工具升级运维平台，详情请参见： <a href="#">内网使用 FineKey 升级运维平台</a>



2.3 立即更新

点击「有新版本，查看更新」按钮，即可进入更新页面。可查看新版本的更新日志。

点击「立即更新」按钮，即可开始更新运维平台。

关于运维平台

×



运维平台

当前版本：v1.12.0

最新版本：v1.12.1

1.12.1更新日志

1) 部署运维平台/运维项目成功后，提示工程地址，方便用户访问

2) 支持导出容器化部署信息，方便后期运维

3) 文件管理支持新建文件夹、批量上传、上传.fvs和.mfvs格式文件、新增同名覆盖提示

4) 备份还原磁盘空间检查优化，升级和还原前备份检查优化，防止备份占用磁盘过高

5) 性能堆栈优化，支持手动上传分析，支持展示历史生成记录，使用顺序优化

6) 系统健康观测、首屏链路、调用链路跳转优化，方便查询和快速定位

暂不更新

立即更新

更新过程中「立即更新」按钮置灰，提示「正在更新」。

更新过程中无进度条提示，用户可关闭窗口去进行其他工作，不影响更新进度。如下图所示：

关于运维平台

×



运维平台

当前版本：v1.12.0

最新版本：v1.12.1

1.12.1更新日志

1) 部署运维平台/运维项目成功后，提示工程地址，方便用户访问

2) 支持导出容器化部署信息，方便后期运维

3) 文件管理支持新建文件夹、批量上传、上传.fvs和.mfvs格式文件、新增同名覆盖提示

4) 备份还原磁盘空间检查优化，升级和还原前备份检查优化，防止备份占用磁盘过高

5) 性能堆栈优化，支持手动上传分析，支持展示历史生成记录，使用顺序优化

6) 系统健康观测、首屏链路、调用链路跳转优化，方便查询和快速定位

暂不更新

正在更新

注：由于 docker 与 FineOps 运维平台有匹配关系，若检测到 docker 引擎版本不符合升级需求，则无法进行在线升级。

会出现弹窗提醒「当前 docker 引擎版本过低，暂不支持在线升级至更高版本，请使用工具离线升级」。

请移步使用该方案升级：[内网使用 FineKey 升级运维平台](#)



## 2.4 重启工程

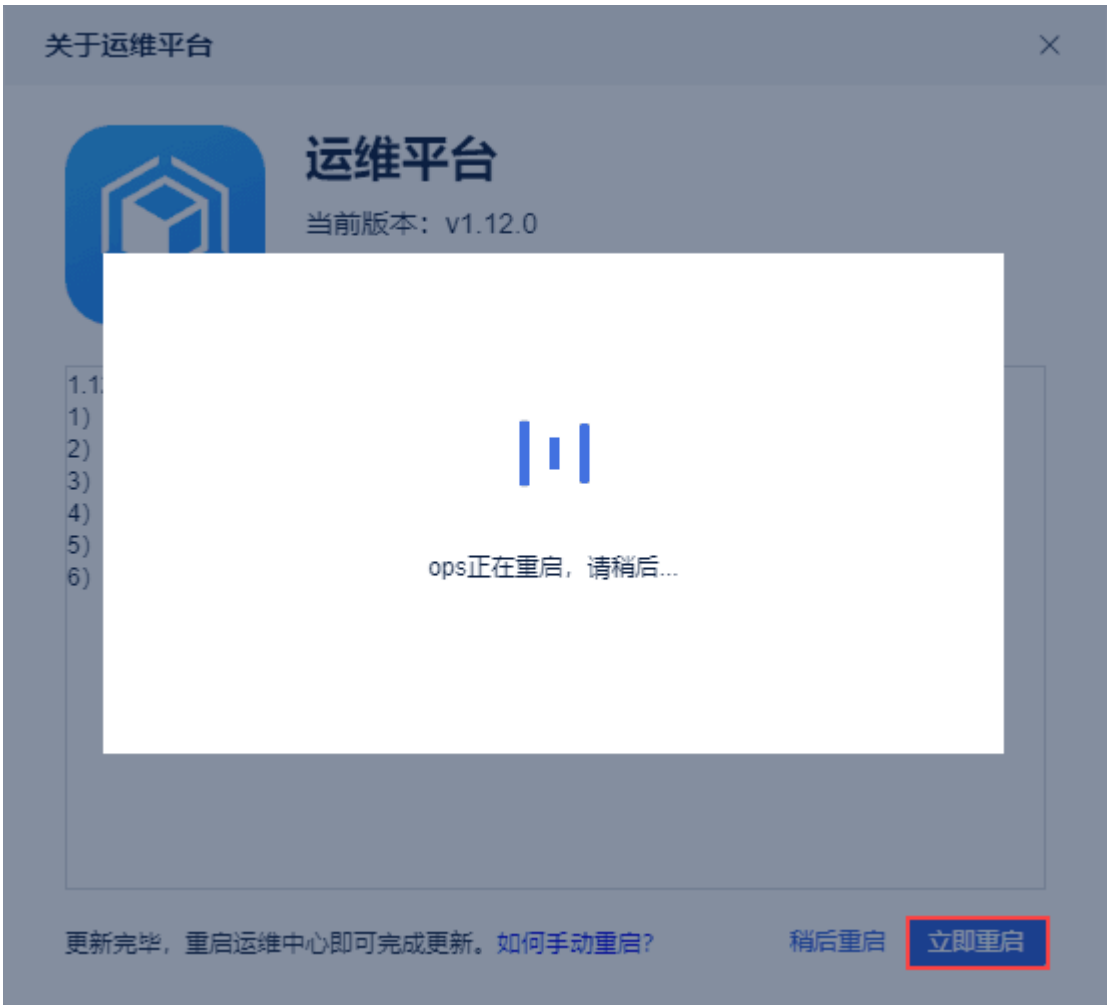
更新完毕后，出现提示「更新成功」，按钮从灰化变为「立即重启」。用户可在线重启工程/稍后手动重启工程。



### 2.4.1 在线重启

点击「立即重启」按钮，即可开始重启运维平台。

重启成功前无法访问运维平台。重启成功后，需要重新登录。



2.4.2 手动重启

重启容器指令：docker restart 容器名，例如：docker restart ops

2.5 更新运维组件

FineOps 运维平台升级完成后，会自动获取该版本依赖的最新运维组件镜像。运维组件也全部成功更新完成后，才能算作运维平台整体升级成功。

2.5.1 自动更新

V1.5.6 及之后版本，当成功在线升级并重启运维平台后，管理员登录运维平台时，运维平台会自动检测并升级运维组件。

管理员登录运维平台，若存在需要升级的运维组件，自动进行一键升级，弹窗提示「运维平台正在完成最后一步更新，更新结束后将自动刷新此页面，更新期间请勿关闭运维平台！」



### 2.5.2 手动更新

V1.5.6 之前版本，运维平台升级成功后，用户需要手动更新运维组件，方可完成整体升级成功。

V1.5.6 及之后版本，若自动更新运维平台失败，会弹窗提示用户进行手动更新。

管理员登录运维平台，点击「运维管理>运维组件」的「一键更新」按钮，即可手动升级所有运维组件。



### 2.6 判断是否升级成功

管理员登录运维平台，点击右上角「」按钮，查看版本，提示「当前已是最新版本」，即代表运维平台升级成功。



## 3.3.2 内网升级运维平台

### 1. 概述

#### 1.1 版本说明

运维平台版本	功能变动
V1.5.5	-
V1.9.0	手动升级过程新增检测并升级 docker 引擎版本，以防由于 docker 引擎版本过低导致功能不可用

#### 1.2 应用场景

为满足客户的新需求以及完善之前版本某些功能的不足之处，FineOps 运维平台在不断地更新迭代。

内网用户，可通过容器化部署工具对运维平台和运维组件进行更新升级。

注 1：外网用户，推荐使用在线升级功能，详情请参见：[更新运维平台](#)

注 2：升级运维平台前，可查阅近期功能变更，详情请参见：[运维平台更新日志](#)

### 2. 升级准备

#### 2.1 确认节点联通

请确保待升级的运维平台所在服务器，与 3.1 节待上传的容器化部署工具 FineKey 所在服务器，互相联通。

若两者不连通，则无法使用本文升级方法。（可将 Finekey 上传至运维平台所在服务器）

#### 2.2 确认剩余磁盘空间

确认待升级节点剩余磁盘空间是否大于 100GB。

如不足 100GB，则无法使用本文升级方法。

#### 2.3 确认是否需要升级

确认待升级节点是否存在需要升级的组件。

如不存在需要升级的组件，则无法使用本文升级方法。

#### 2.4 保存原 yaml 文件

从原容器化部署工具 finekey/conf 目录下获取 finekey.yaml 文件，下文需要用到。



### 3. 准备 FineKey 工具

#### 3.1 获取 FineKey 安装包

点击可跳转至下载页：[FineOps 运维平台下载](#)。

有三种类型的安装包，内网用户请下载「纯净版/全量版离线安装包」：

安装包类型	说明
在线版安装包	1) 服务器可访问外网，支持下载运维组件镜像，安装包体积小
	2) 支持从云端下载 FineReport/ FineBI 项目组件镜像，可通过 FineOps 可视化部署工程
	3) 可对接已有 FineReport/ FineBI 工程
纯净版离线安装包	1) 服务器为内网，内置全部运维组件镜像，安装包体积大
	2) 未内置 FineReport/ FineBI 项目组件镜像，无法通过 FineOps 可视化部署工程
	3) 用于对接已有 FineReport/ FineBI 工程
全量版离线安装包	1) 服务器为内网，内置全部运维组件镜像，安装包体积大
	2) 内置 FineReport/ FineBI 项目组件镜像，可通过 FineOps 可视化部署工程
	3) 可对接已有 FineReport/ FineBI 工程

#### 3.2 上传 FineKey 安装包

获得工具包后，需要将它上传到 Linux，上传文件的方法请自行查阅相关资料。

本文示例是将工具包上传到了 CentOS 7.6.1810 的 /home 路径下，如下图所示：

```
[root@localhost home]# cd /home
[root@localhost home]# ls
finekey-operation.tar.gz
[root@localhost home]# ls
```

#### 3.3 解压 FineKey 安装包

- 1) 先使用 cd 命令进入工具包所在文件夹下。
- 2) 键入如下命令进行解压：

tar zxvf finekey-operation.tar.gz

```
[root@localhost home]# cd /home
[root@localhost home]# ls
finekey-operation.tar.gz
[root@localhost home]# cd /home
[root@localhost home]# tar zxvf finekey-operation.tar.gz
finekey/
finekey/bin/
finekey/bin/finekey
finekey/bin/fine_net_test
finekey/conf/
finekey/conf/finekey.yaml
finekey/conf/finekey.push.yaml
finekey/resources/
finekey/resources/images/
```

解压后的文件如下表所示：

文件夹/文件	文件说明
finekey/bin	运行 finekey 的可执行文件
finekey/conf	
finekey.yaml	新部署运维平台配置文件
finekey.push.yaml	拉取镜像配置文件
finekey/resources	资源文件，镜像包放在其中 images 文件夹中

### 3.4 配置 finekey.yaml 文件

yaml 文件指的是工具解压目录下 conf 文件夹里的 finekey.yaml，该文件是控制后面整个升级流程的。

#### 替换 yaml 文件

取出 2.4 节从原 finekey 部署工具 finekey/conf 目录下获取 finekey.yaml 文件，将文件中的配置值，一项一项的复制替换到新 FineKey 部署工具 finekey/conf 目录下的 finekey.yaml 文件。

注 1：请勿直接替换 yaml 文件，或一键复制粘贴文件内容，因为新旧版本中，yaml 文件设置可能存在微小差异，建议逐项拷贝配置值即可。

注 2：如果 finekey 部署工具和运维平台不在同一台服务器下，yaml 文件中需要新增以下待升级节点的 ip、用户名、密码，其他均由 finekey 自动识别。

```
update:
nodes:
  - ip: localhost                                #机器 ip
    port: 22                                    #机器 ssh 端口
    user:                                       #节点 ssh 用户
    password:                                  #节点 ssh 密码
#  - ip: 192.168.101.145
#    port: 22
#    user: xxx
#    password: xxx
```

## 4. 升级运维平台

4.1 启动升级

- 1) 先使用 cd 命令进入工具包所在文件夹的 bin 目录下。
- 2) 使用语句启动升级： ./finekey update

4.2 执行升级流程

启动升级后，FineKey 升级运维平台的流程如下表所示，请耐心等待升级完成。

步骤	内容
	FineKey 工具会优先检查第二章的升级前提是否达成。
检查升级前提	如不达成升级前提， 则直接结束升级步骤。  如达成升级前提， 则进行下一步。
	FineKey 工具会自动检测待升级组件和 FineKey 内镜像版本是否一致。
检查组件版本	如版本一致， 则没有升级必要， 直接结束升级步骤。  如版本不一致， 则进行下一步。
自动备份组件	FineKey 工具会自动备份需要升级的组件挂载的数据 注： 升级指定组件时， 可能会关闭/备份关联组件。
替换升级镜像	FineKey 工具会自动更换镜像进行升级  FineKey 工具会自动重启运维平台
重启运维平台	若重启成功， 则升级成功  若重启失败， 则自动回滚， 取消升级

4.3 更新运维组件

FineOps 运维平台升级完成后， 会自动获取该版本依赖的最新运维组件镜像。运维组件也全部成功更新完成后， 才能算作运维平台整体升级成功。

4.3.1 自动更新

V1.5.6 及之后版本， 当成功在线升级并重启运维平台后， 管理员登录运维平台时， 运维平台会自动检测并升级运维组件。

管理员登录运维平台， 若存在需要升级的运维组件， 自动进行一键升级， 弹窗提示「运维平台正在完成最后一步更新， 更新结束后将自动刷新此页面， 更新期间请勿关闭运维平台！」



### 4.3.2 手动更新


V1.5.6 之前版本，运维平台升级成功后，用户需要手动更新运维组件，方可完成整体升级成功。

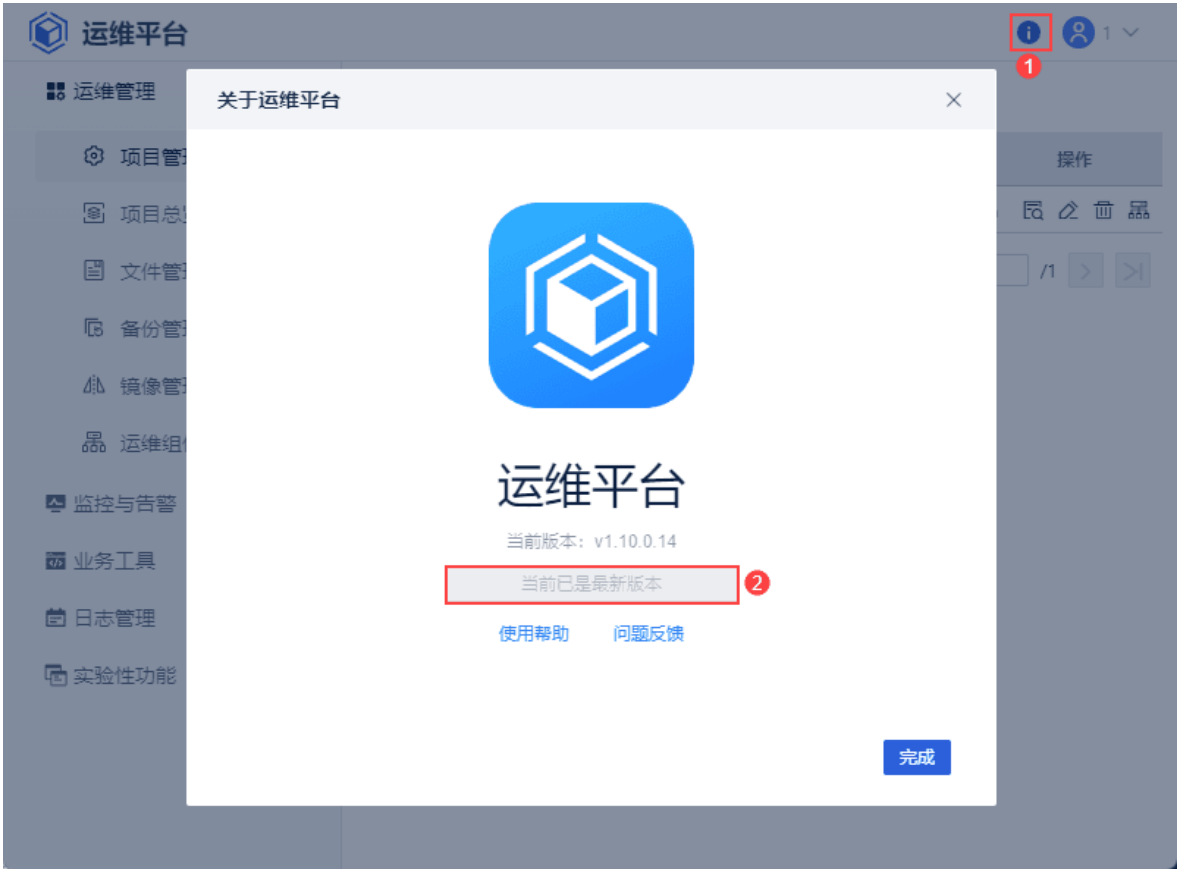
V1.5.6 及之后版本，若自动更新运维平台失败，会弹窗提示用户进行手动更新。

管理员登录运维平台，点击「运维管理>运维组件」的「一键更新」按钮，即可手动升级所有运维组件。



### 4.4 检查升级是否成功

升级完成后，管理员登录运维平台，点击右上角「」按钮，查看版本，提示「当前已是最新版本」，即代表升级成功。



## 5. 注意事项

### 问题描述

运维平台升级过程中，提示 docker engine 相关报错

### 原因分析

由于 docker 引擎、FineOps 运维平台、项目的 ops\_agent 组件存在对应关系。

如需手动升级运维平台，必须确保运维平台所在服务器、运维平台对接的项目所在服务器，均支持升级 docker 引擎。

如在执行 finekey update 过程中，检测到系统禁止升级 docker 引擎，就会出现该报错，中断升级。

### 解决方案

- 1) 依次分别登录运维平台所在服务器、运维平台对接的项目所在的每一台服务器
- 2) 执行语句，升级 docker 引擎：./finekey engine --update
- 3) 再次参考上文使用语句启动升级：./finekey update

### 注意事项

在升级 FineOps 运维平台时，如触发了 docker 引擎升级，运维平台和运维平台对接的项目，均会自动重启，影响终端用户使用。

因此请选择非工作时间进行升级，降低业务影响。



## 3.4 常见问题

### 3.4.1 部署运维平台问题排查

#### 1. 概述

##### 1.1 版本说明

运维平台版本	功能变动
V1.5.5	-

##### 1.2 应用场景

本文将介绍部署 FineOps 运维平台过程中出现的常见问题和解决方案。

#### 2. FineKey 启动失败

##### 2.1 finekey is already running

报错语句：finekey is already running

原因分析：finekey 工具正在运行，需要关闭工具占用的进程，方可重启。

解决方案：通过语句 `ps -ef|grep finekey`，查看相关的进程号，`kill -9 进程号` 即可。

##### 2.2 disk avaiillable capacity:xxx, less then 500G

报 错 语 句：ERROR: 2022/05/18 17:08:36 node 192.168.1.24 disk avaiillable capacity:3789952, less then 500G

原因分析：安装时会自动检测 finekey.yaml 文件中配置的挂载卷目录所在磁盘分区空闲空间要≥500G，默认的~/data 代表的目录是/root/data。

解决方案：这是容器化部署的前提，必须满足。请联系相关运维人员对服务器进行扩容。

##### 2.3 memory avaiillable capacity: xxx, less then 16G

报错语句：ERROR: 2022/04/20 14:12:55 node 192.168.101.144 memory avaiillable capacity: 7302952, less then 16G

原因分析：服务器内存容量不足

解决方案：这是容器化部署的前提，必须满足。请联系相关运维人员对服务器进行扩容。

##### 2.4 connection refused

报错语句：ERROR: 2022/05/18 17:33:07 finedb connect error:dial tcp 192.168.1.24:3306: connect: connection refused

原因分析：finekey.yaml 文件中配置的相关端口，没有开放，无法连接。



解决方案：请检查服务器相关端口的防火墙、安全组等设置是否开放。

### 3. finekey.yaml 文件配置错误

#### 3.1 mapping values are not allowed in this context

报 错 语 句 : ERROR: 2022/04/20 13:56:28 parse config file /Users/carly/finekey/conf/finekey.yaml error: yaml: line 14: mapping values are not allowed in this context

报错原因：yaml 文件对格式的要求非常高，出现该错误，大概率都是由于内容格式存在错误。

解决方案：根据报错提示中的 line 号修改配置文件格式。

#### 3.2 address already in use

报错语句: ERROR: 2022/05/18 17:15:23 192.168.1.24 :try bind port error:listen tcp :12100: bind: address already in use

原因分析：端口冲突，已有的组件占用了默认端口。

解决方案：

- 1) 端口确实冲突：打开安装目录下 conf 文件夹里的 finekey.yaml 文件，找到对应组件，修改 portMapping 中映射端口，保存退出后重新运行即可。
- 2) 端口没有冲突：仔细排查安装目录下 conf 文件夹里的 finekey.yaml 文件，是否是 finekey.yaml 中各处字符格式不对，比如漏加空格或用了中文符号等。

#### 3.3 can't use localhost

报错语句: ERROR: 2022/04/20 13:58:14 can't use localhost, 127.0.0.1 etc as node.ip in cluster mode.. must ensure ensure IP connectivity of all nodes

报错原因：集群下不能使用 localhost 等回环地址作为 IP

解决方案：修改 IP 为其他节点都能访问的真实 IP

#### 3.4 add user to docker group error

报错语句: add user to docker group error: sudo: 1 次错误密码尝试

解决方案：

- 1) 确保 finekey.yaml 文件配置的 ssh 用户和运行 finekey 的用户保持一致
- 2) 检查 finekey.yaml 文件中配置 ssh 用户名密码是否正确
- 3) 远程连接工具问题，关闭重启远程工具并再次尝试

### 3.5 ssh connect error

报错语句：ERROR: 2022/04/20 13:58:14 node:192.168.101.144 ssh connect error:ssh: handshake failed: ssh: unable to authenticate, attempted methods

报错原因：SSH 连接失败，通常报错明细中出现 unable to authenticate 则代表密码有问题；出现 Connection refused 则代表网络不通等可能

解决方案：

- 1) 确保 finekey.yaml 文件配置的 ssh 用户和运行 finekey 的用户保持一致
- 2) 检查 finekey.yaml 文件中配置 ssh 用户名密码是否正确
- 3) 远程连接工具问题，关闭重启远程工具并再次尝试

### 4. 非 root 用户运行 FineKey

推荐使用 root 用户运行 FineKey 容器化部署工具。

如需使用非 root 用户运行 FineKey 容器化部署工具，需要注意以下三点：

#### 1) 用户权限

该用户需要至少拥有以下命令的权限

命令	使用场景
cp	安装 docker 时，cp 到/usr/bin 等目录下
groupadd	添加 docker 用户组
gpaswd	将非 root 用户添加到 docker 用户组
systemctl	安装 docker 时将 docker 添加到开机自启动服务
	非必须项，如果没有权限则跳过
kill	使 docker 引擎信任该非安全仓库
	kill 发送信号至 docker 引擎，使其热加载
sysctl	修改 max_map_count

#### 2) 修改 yaml 文件

finekey.yaml 文件中必须填写该非 root 用户的密码。

#### 3) 执行部署命令

执行./finekey 命令时，不要对 finekey 本身使用 sudo, 即不要使用 sudo ./finekey 命令。

### 5. 容器化运维指令

列出一些常用的容器化运维指令：

- 重启容器指令：docker restart 容器名，例如：docker restart fanruan\_fr

- 查看所有容器运行情况： docker ps -a
- 启动指定容器： docker start 容器名， 例如： docker start fanruan\_fr
- 停止指定容器： docker stop 容器名， 例如： docker stop fanruan\_fr

6. 运维平台外网地址

FineOps 运维平台若部署在内网环境下，某些需要外网的场景功能无法使用。

本文将简单介绍部分功能需要开通的外网端口，用户可根据需要自行开通网络白名单。

功能	网址
获取镜像/更新运维平台/ 更新项目组件	域名： registry.cn-hangzhou.aliyuncs.com  端口： 443
	四选一， 有一个能联通即可：
获取更新信息	https://cloud.fanruan.com/config https://cloud.fanruan.com:8443/config http://cloud.fanruan.com/config http://cloud.fanruan.com:8090/config

### 3.4.2 端口占用检查与防火墙配置

#### 1. 概述

##### 1.1 问题描述

- 部署运维平台报错：connection refused

报错语句：ERROR: 2022/05/18 17:33:07 finedb connect error:dial tcp 192.168.1.24:3306: connect: connection refused

- 部署运维项目报错：Server IP try to bind ports failed!

报错语句：Server IP try to bind ports 9114,9100,9200,9080,9070 failed!



##### 1.2 原因分析

待部署的运维平台组件/运维项目组件容器，需要占用一些端口。

如果这些端口已被占用/没有开放，则无法成功部署，会出现以上报错。

##### 1.3 解决方案

请依次检查以下三者：

###### 1) 检查该端口是否被服务器中其他进程占用

- 如果端口被占用，则需要更换一个未被占用的端口，再检查新端口的下两项。
- 如果端口未被占用，则需要检查该端口的下两项。

###### 2) 检查服务器防火墙是否放行端口

- 如果服务器防火墙未开启，建议开启。
- 如果服务器防火墙开启，需要开放所需端口。

- 如果端口已开放，请检查下一项。

3) 如为云服务器，请检查服务器安全组是否放行端口

- 云服务器安全组端口放行方式本文不作介绍，请联系你的云服务器厂商咨询。

2. 检查端口占用情况

本文提供多种方法检查 Linux 系统中的端口是否被其他进程占用。

需要注意的是，执行以下命令通常需要以 root 用户或具有足够权限的用户身份运行，以获取完整的进程信息。

2.1 使用 netstat 命令

**命令语句**  
netstat -tuln | grep <端口号>  
将 <端口号> 替换为您要检查的具体端口号

这条命令将显示监听该端口的进程信息

**返回说明**  
如果有进程正在使用该端口，您将能够看到相关的输出  
如果该端口未被占用，则无任何输出

示例

```
[root@192 ~]# netstat -tuln | grep 8080
[root@192 ~]# netstat -tuln | grep 80
```

8080端口未被占用

80端口被占用

tcp	0	0	192.168.101.58:80	0.0.0.0:*	LISTEN
tcp	0	0	192.168.101.58:8081	0.0.0.0:*	LISTEN
tcp	0	0	192.168.101.58:11800	0.0.0.0:*	LISTEN
tcp	0	0	192.168.101.58:6080	0.0.0.0:*	LISTEN
tcp	0	0	192.168.101.58:12800	0.0.0.0:*	LISTEN

2.2 使用 lsof 命令

**命令语句**  
lsof -i :<端口号>  
将 <端口号> 替换为您要检查的具体端口号

这条命令将显示打开该端口的进程及其相关信息。

**返回说明**  
如果有进程正在使用该端口，您将能够看到相关的输出  
如果该端口未被占用，则无任何输出

示例

```
[root@192 ~]#
[root@192 ~]#
[root@192 ~]# lsof -i :9999 9999端口未被占用
[root@192 ~]#
[root@192 ~]#
[root@192 ~]# lsof -i :80 80端口被占用
COMMAND  PID  USER  FD   TYPE    DEVICE  SIZE/OFF  NODE  NAME
docker-pr 18003  root   3u    IPv4  446516966      0t0  TCP k8s.node0.fanruan:http->172.17.0.11:51362 (ESTABLISHED)
docker-pr 18003  root    4u    IPv4  443187061      0t0  TCP k8s.node0.fanruan:http (LISTEN)
docker-pr 18003  root   13u    IPv4  443201053      0t0  TCP k8s.node0.fanruan:http->172.17.0.9:39918 (ESTABLISHED)
docker-pr 18003  root   21u    IPv4  443243366      0t0  TCP k8s.node0.fanruan:http->172.17.0.17:33076 (ESTABLISHED)
docker-pr 18003  root   31u    IPv4  443244634      0t0  TCP k8s.node0.fanruan:http->172.17.0.19:54718 (ESTABLISHED)
docker-pr 18003  root   37u    IPv4  445064555      0t0  TCP k8s.node0.fanruan:http->172.17.0.10:56934 (FIN_WAIT2)
[root@192 ~]#
```

3. 检查防火墙放行情况

- 1) 建议服务器防火墙长期启用。
- 2) 建议开放且只开放必要的端口，防止不必要的网络暴露。

3.1 常用语句

场景	语句
检查防火墙状态	systemctl status firewalld
启动防火墙	systemctl start firewalld
关闭防火墙	systemctl stop firewalld
防火墙开机启动	systemctl enable firewalld
防火墙开机禁用	systemctl disable firewalld
查看所有放行端口	firewall-cmd --zone=public --list-ports
查看指定端口是否放行	firewall-cmd --zone=public --query-port=<端口>/<协议>
放行指定端口	firewall-cmd --zone=public --add-port=<端口>/<协议> --permanent
取消放行指定端口	firewall-cmd --zone=public --remove-port=<端口>/<协议> --permanent
更新防火墙规则	firewall-cmd --reload

3.2 示例

本示例介绍如何开放防火墙并放行 8080 端口。

1) 启动防火墙

- 检查防火墙状态：systemctl status firewalld。返回值为「not running」，说明未启动防火墙
- 启动防火墙：systemctl start firewalld
- 再次检查防火墙状态：systemctl status firewalld。返回值为「running」，说明已启动防火墙

```
[root@192 ~]#
[root@192 ~]# firewall-cmd --state
not running
[root@192 ~]# systemctl start firewalld
[root@192 ~]# firewall-cmd --state
running
[root@192 ~]#
```

## 2) 开放 8080 端口

- 检查端口是否放行: `firewall-cmd --zone=public --query-port=8080/tcp`。返回值为「no」, 说明未放行 8080 端口
- 开放端口: `firewall-cmd --zone=public --add-port=8080/tcp --permanent`。返回值为「success」, 说明已放行 8080 端口
- 更新防火墙规则: `firewall-cmd --reload`。返回值为「success」, 说明已更新规则
- 再次检查端口是否放行: `firewall-cmd --zone=public --query-port=8080/tcp`。返回值为「yes」, 说明已放行 8080 端口

```
[root@192 ~]# firewall-cmd --state
running
[root@192 ~]# firewall-cmd --zone=public --query-port=8080/tcp
no
[root@192 ~]# firewall-cmd --zone=public --add-port=8080/tcp --permanent
success
[root@192 ~]# firewall-cmd --reload
success
[root@192 ~]# firewall-cmd --zone=public --query-port=8080/tcp
yes
[root@192 ~]#
```

### 3.4.3 docker 常用操作指南

#### 1. 概述

容器化部署后，用户需要掌握一些基础的 docker 运维知识。

#### 2. 查找容器 ID/容器名/容器状态

注：推荐优先使用 FineOps 「[运维组件](#)」和「[组件管理](#)」功能查看容器信息。

命令语句：docker ps -a

返回值说明：可以查看所有容器的状态、端口映射以及其他重要信息，从而更好地管理和调试容器化应用程序。

返回值	说明
CONTAINER ID	容器的唯一标识符 通常以大写字母和数字的组合形式表示
IMAGE	容器使用的镜像名称
COMMAND	用于启动容器的命令
CREATED	容器创建的时间戳
STATUS	容器的运行状态和健康状况 如正在运行（Up）、已停止（Exited）、正在重启等
PORTS	容器所暴露的端口和绑定到主机的端口
NAMES	指定的容器名称或自动生成的容器名称

示例：

在示例服务器中，存在一个 ops 容器，容器 ID 为「flac156d4db9」，使用了「ops:v1.6.0」镜像，并以「fanruan230718121827\_ops」为容器名称正在运行。

容器于「6 天前」创建，从创建至今已运行了 6 天，容器被标记为「healthy」，表示容器内部的应用程序正常运行并且健康。

它将主机「192.168.101.58」的「8081」端口映射到容器的「8080」端口，也就是用户可以通过访问主机的「192.168.101.58:8081」地址来与 OPS 通信。

```
root@192 ~# docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                                                                                                     NAMES
b6c254140c9   skywalking_ops:v20.0.12-9.1.0.53   "/docker-entrypoint.    6 days ago    Up 6 days    192.168.101.58:11800->11800/tcp, 1234/tcp, 192.168.101.58:12800->12800/tcp   fanruan230718121827_skywalking_ops
flac156d4db9   ops_agent:v1.6.0                    "java -jar ops-docke    6 days ago    Up 6 days    192.168.101.58:9070->9070/tcp                                           fanruan230718121827_ops_agent
flac156d4db9   ops:v1.6.0                          "/docker-entrypoint.    6 days ago    Up 6 days (healthy) 192.168.101.58:8081->8080/tcp                                           fanruan230718121827_ops
4c9a959e98e   grafana:v20.0.9-9.2.2              "/grafana-entrypoint.    6 days ago    Up 6 days    3000/tcp                                                              fanruan230718121827_grafana
8e5d343b5b7   prometheus:v20.0.7-2.34.0          "/docker-entrypoint.    6 days ago    Up 6 days    9090/tcp                                                              fanruan230718121827_prometheus
9c3d526b4e0   alertmanager:v20.0.5-0.24.0        "/docker-entrypoint.    6 days ago    Up 6 days    9093/tcp                                                              fanruan230718121827_alertmanager
69eb51582e   node_exporter:v20.0.1-1.3.1        "/bin/node_exporter -    6 days ago    Up 6 days    192.168.101.58:9113->9113/tcp                                           fanruan230718121827_node_exporter
f1a85e4e0ec   nginx_exporter:v20.0.0-0.10.0       "/usr/bin/nginx-prom.    6 days ago    Up 6 days    80/tcp, 192.168.101.58:80->8080/tcp                                     fanruan230718121827_nginx_exporter
d8254f9226   nginx:v20.0.7-1.21.4               "/docker-entrypoint.    6 days ago    Up 6 days    7979/tcp, 192.168.101.58:9114->9114/tcp                                   fanruan230718121827_nginx
898d74143eba   elasticsearch_exporter:v20.0.0-1.5.0 "/bin/elasticsearch_     6 days ago    Up 6 days    192.168.101.58:9200->9200/tcp, 9380/tcp                                fanruan230718121827_elasticsearch_exporter
a7eddai4f532   elasticsearch:v20.0.2-0.4.1         "/bin/tini -- /usr/l.    6 days ago    Up 6 days    192.168.101.58:9200->9200/tcp, 9380/tcp                                fanruan230718121827_elasticsearch
76acc38c385e   registry:v20.0.1-2.0.1             "/entrypoint.sh /etc-    6 days ago    Up 6 days    0.0.0.0:5000->5000/tcp, :::5000->5000/tcp                               fanruan_registry
```

#### 3. 重启/关闭/启动容器

命令语句：容器 ID/容器名称请根据第二章方法查找。



场景	语句
重启容器	docker restart 容器 ID/容器名
关闭容器	docker stop 容器 ID/容器名
启动容器	docker start 容器 ID/容器名

示例：本节示例如何关闭和启动 ops 容器。

### 3.1 关闭 ops 容器

- 1) 查找名称中包含 ops 的容器：docker ps -a | grep ops

根据返回值确定容器 ID 为：flac156d4db9

- 2) 关闭该容器：docker stop flac156d4db9

- 3) 再次查找名称中包含 ops 的容器：docker ps -a | grep ops

确认容器状态为 Exited，代表容器成功关闭

```
[root@192 ~]# docker ps -a | grep ops
baec254140c9    ops_agent:v1.6.0    "java -jar ops-docke..."    6 days ago    Up 6 days    fanruan230718121827_ops_agent
192.168.101.58:9070->9070/tcp
flac156d4db9    ops:v1.6.0          "/docker-entrypoint..."    6 days ago    Up 18 minutes (health)    fanruan230718121827_ops
hy) 192.168.101.58:8081->8080/tcp
[root@192 ~]# docker stop flac156d4db9
flac156d4db9
[root@192 ~]# docker ps -a | grep ops
baec254140c9    ops_agent:v1.6.0    "java -jar ops-docke..."    6 days ago    Up 6 days    fanruan230718121827_op
agent
flac156d4db9    ops:v1.6.0          "/docker-entrypoint..."    6 days ago    Exited (0) About a m
inute ago
fanruan230718121827_op
[root@192 ~]#
```

### 3.2 启动 ops 容器

- 1) 查找名称中包含 ops 的容器：docker ps -a | grep ops

此时容器状态为 Exited，代表容器未启动。根据返回值确定容器 ID 为：flac156d4db9

- 2) 启动该容器：docker start flac156d4db9

- 3) 查看全部容器状态：docker ps -a

确保 ops 容器及其关联组件等（例如 ops\_agent）均启动

```
[root@192 ~]# docker ps -a | grep ops
baec254140c9    ops_agent:v1.6.0    "java -jar ops-docke..."    6 days ago    Up 6 days    fanruan230718121827_ops_agen
t
192.168.101.58:9070->9070/tcp
flac156d4db9    ops:v1.6.0          "/docker-entrypoint..."    6 days ago    Exited (0) 3 minutes
ago
fanruan230718121827_ops
[root@192 ~]# docker start flac156d4db9
flac156d4db9
[root@192 ~]# docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS
PORTS
adb03c83633    skywalking_oap:v20.0.12-9.1.0.53   "/docker-entrypoint..."    6 days ago    Up 6 days
192.168.101.58:11800->11800/tcp, 1234/tcp, 192.168.101.58:12800->12800/tcp    fanruan230718121827_skywalking_oap
baec254140c9    ops_agent:v1.6.0                    "java -jar ops-docke..."    6 days ago    Up 6 days
192.168.101.58:9070->9070/tcp    fanruan230718121827_ops_agent
flac156d4db9    ops:v1.6.0                          "/docker-entrypoint..."    6 days ago    Up 7 minutes (health)
192.168.101.58:8081->8080/tcp    fanruan230718121827_ops
4c6b895869b6    grafana:v20.0.9-9.2.2              "/grafana-entrypoint..."    6 days ago    Up 6 days
3000/tcp    fanruan230718121827_grafana
8e52d343bb07    prometheus:v20.0.7-2.34.0          "/docker-entrypoint..."    6 days ago    Up 6 days
9090/tcp    fanruan230718121827_prometheus
```

## 4. 查看容器配置项

操作步骤：

- 1) 参考第一章查找容器 ID：docker ps -a

2) 查看容器配置项: docker inspect 容器 ID/容器名

4.1 查看 fr 容器配置

1) 查找名称中包含 fr 的容器: docker ps -a | grep fr

根据返回值确定容器 ID 为: 0b69cd526d44

2) 查看该容器配置项: docker inspect 0b69cd526d44

```
[root@192 bin]# docker ps -a | grep fr
0b69cd526d44 192.168.101.58:5000/fr:11.0-20230711 "/docker-entrypoint..."
3 minutes ago Up 3 minutes (healthy) 192.168.101.58:12100->12100/tcp, 192.168.101.58:608
0->8080/tcp fanruan230724160433_fr
[root@192 bin]# docker inspect 0b69cd526d44
[
  {
    "Id": "0b69cd526d4414c777793633efb76941ea4b60646a8414379030ace17a30372c",
    "Created": "2023-07-24T08:06:01.370101707Z",
    "Path": "/docker-entrypoint.sh",
    "Args": [
      "catalina.sh",
      "run"
    ]
  }
]
```

3) 此处摘选部分常用配置项，帮助用户查看。

- Binds 中，为工程外挂目录。

冒号前面为宿主机（容器所在机器）上目录，冒号后面为容器内部文件目录

例如如需上传 FineReport 模板到该工程中，应当上传到 /home/fr/fanruan230724160433/fr/reportlets 路径下

```
"Name": "/fanruan230724160433_fr",
"RestartCount": 0,
"Driver": "overlay2",
"Platform": "linux",
"MountLabel": "",
"ProcessLabel": "",
"AppArmorProfile": "",
"ExecIDs": null,
"HostConfig": {
  "Binds": [
    "/home/fr/fanruan230724160433/fr/customlib:/usr/local/tomcat/webapps/webroot/WEB-INF/customlib",
    "/home/fr/fanruan230724160433/fr/help:/usr/local/tomcat/webapps/webroot/help",
    "/home/fr/fanruan230724160433/fr/resources:/usr/local/tomcat/webapps/webroot/WEB-INF/resources",
    "/home/fr/fanruan230724160433/fr/config:/usr/local/tomcat/webapps/webroot/WEB-INF/config",
    "/home/fr/fanruan230724160433/fr/local:/usr/local/tomcat/webapps/webroot/WEB-INF/local",
    "/home/fr/fanruan230724160433/fr/treasures:/usr/local/tomcat/webapps/webroot/WEB-INF/treasures",
    "/usr/bin/docker:/usr/bin/docker",
    "/home/fr/fanruan230724160433/fr/reportlets:/usr/local/tomcat/webapps/webroot/WEB-INF/reportlets",
    "/home/fr/fanruan230724160433/fr/logs:/usr/local/tomcat/webapps/webroot/logs",
    "/var/run/docker.sock:/var/run/docker.sock",
    "/home/fr/fanruan230724160433/fr/plugins:/usr/local/tomcat/webapps/webroot/WEB-INF/plugins",
    "/home/fr/fanruan230724160433/fr/assets:/usr/local/tomcat/webapps/webroot/WEB-INF/assets",
    "/home/fr/fanruan230724160433/fr/schedule:/usr/local/tomcat/webapps/webroot/WEB-INF/schedule",
    "/home/fr/fanruan230724160433/fr/backup:/usr/local/tomcat/webapps/webroot/backup",
    "/home/fr/fanruan230724160433/fr/mount:/usr/local/tomcat/mount",
    "/home/fr/fanruan230724160433/fr/tomcat_logs:/usr/local/tomcat/logs",
    "/home/fr/fanruan230724160433/fr/classes:/usr/local/tomcat/webapps/webroot/WEB-INF/classes"
  ],
  "ContainerIDFile": "",
  "LogConfig": {
    "Type": "json-file",
    "Config": {
      "max-file": "2",
      "max-size": "64m"
    }
  }
},
}
```

- 在 Env 中，可以查看到容器内目录、内存、端口、工程名、JVM 参数等信息

```
"Config": {
  "Hostname": "0b69cd526d44",
  "Domainname": "",
  "User": "",
  "AttachStdin": false,
  "AttachStdout": false,
  "AttachStderr": false,
  "ExposedPorts": {
    "12100/tcp": {},
    "8080/tcp": {}
  },
  "Tty": true,
  "OpenStdin": true,
  "StdinOnce": false,
  "Env": [
    "CONTEXT_PATH=webroot",
    "GROUP_ID=0",
    "AGENT_ADDRESSES=http://192.168.101.58:6070/",
    "TZ=Asia/Shanghai",
    "CATALINA_OPTS=-Xmx12288m",
    "USER_ID=0",
    "jgroups.external_addr=192.168.101.58",
    "EXTERNAL_HTTP_PORT=6080",
    "ops_sign_key=c607f7684ab04f78",
    "PATH=/usr/local/tomcat/bin:/usr/local/openjdk-8/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin",
    "JAVA_HOME=/usr/local/openjdk-8",
    "LANG=zh_CN.utf8",
    "JAVA_VERSION=8u332",
    "CATALINA_HOME=/usr/local/tomcat",
    "TOMCAT_NATIVE_LIBDIR=/usr/local/tomcat/native-jni-lib",
    "LD_LIBRARY_PATH=/usr/local/tomcat/native-jni-lib",
    "GPG_KEYS=48F8E69F6390C9F25CFEDCD268248959359E722B A9C5DF4D22E9998D9875A5110C01C5A2F6059E7 DCFD35E0BF8CA7344752DE8B6FB21E8933C60243",
    "TOMCAT_MAJOR=9",
    "TOMCAT_VERSION=9.0.63",
    "TOMCAT_SHA512=4b905018164026756bd36ab9fde8f6b21c886acb8e5255d93f8938491e4d375dd18b9fc58ee23e3d78b16e8b81271c1c998e5592beedcac632567c2ca9411c69",
    "JAVA_OPTS=-server -Duser.country=CN -Duser.language=zh -Djava.awt.headless=true\\t-XX:+HeapDumpOnOutOfMemoryError -XX:PerBytecodeRecompilationCutoff=-1 -XX:PerMethodRecompilationCutoff=-1 -XX:MaxDirectMemorySize=2048M -XX:ReservedCodeCacheSize=250M -Dfineio.read_mem_limit=2 -Dfineio.write_mem_limit=1 -Dsun.zip.disableMemoryMapping=true",
    "DEBIAN_FRONTEND=noninteractive",
    "MALLOC_ARENA_MAX=1",
    "COPY_INITIAL_MAP=/usr/local/tomcat/initial_data/map:/usr/local/tomcat/webapps/webroot/WEB-INF/assets/map",
    "RUNTIME_ENV_fr_0=TZ=Asia/Shanghai",
    "RUNTIME_ENV_fr_1=CATALINA_OPTS=${CATALINA_OPTS}",
    "RUNTIME_ENV_fr_2=ops_sign_key=${ops_sign_key}",
    "RUNTIME_ENV_fr_3=EXTERNAL_HTTP_PORT=${EXTERNAL_HTTP_PORT}",
    "RUNTIME_ENV_fr_4=jgroups.external_addr=${jgroups.external_addr}",
    "RUNTIME_ENV_fr_5=CORE_EXTERNAL_PORT=${CORE_EXTERNAL_PORT}",
    "RUNTIME_ENV_fr_6=FILE_SYNC_EXTERNAL_PORT=${FILE_SYNC_EXTERNAL_PORT}",
    "RUNTIME_ENV_fr_7=GENERAL_CACHE_EXTERNAL_PORT=${GENERAL_CACHE_EXTERNAL_PORT}"
  ]
}
```

4.2 查看 Redis 容器配置

1) 查找名称中包含 redis 的容器： docker ps -a | grep redis

根据返回值确定容器 ID 为： 47dadb9d756f

2) 查看该容器配置项： docker inspect 47dadb9d756f

```
[root@iZ28x3egfhLZ ~]# docker ps -a | grep redis
bc6e25c93d8c 120.27.40.246:5000/redis_exporter:v20.0.0-1.33.0 "/redis_exporter"
5 days ago Up 5 days 120.27.40.246:9121->9121/tcp fanruan230719153907_re

dis_exporter
47dadb9d756f 120.27.40.246:5000/redis:v20.1.1-6.2.11 "docker-entrypoint.
s..." 5 days ago Up 5 days 120.27.40.246:6379->6379/tcp fanruan230719153907_re

dis
[root@iZ28x3egfhLZ ~]#
[root@iZ28x3egfhLZ ~]# docker inspect 47dadb9d756f
[
  {
    "Id": "47dadb9d756f261130c3389c7af2ee8fcf2231af0c162f7d6bbdaf948891a88b",
    "Created": "2023-07-19T07:44:39.140061579Z",
    "Path": "docker-entrypoint.sh",
    "Args": [
      "redis-server",
      "/data/redis.conf"
    ],
  }
]
```

3) 此处摘选部分常用配置项，帮助用户查看。

```
"HostConfig": {
  "Binds": [
    "/data/bi6/fanruan230719153907/redis:/data:/data"
  ],
  "ContainerIDFile": "",
  "LogConfig": {
    "Type": "local",
    "Config": {
      "compress": "true",
      "max-file": "5",
      "max-size": "20m"
    }
  },
  "NetworkMode": "default",
  "PortBindings": {
    "6379/tcp": [
      {
        "HostIp": "192.168.101.58",
        "HostPort": "6379"
      }
    ]
  },
  "RestartPolicy": {
    "Name": "on-failure",
    "MaximumRetryCount": 10
  }
},
```

5. 执行组件命令

操作步骤：

- 1) 参考第一章查找容器 ID： docker ps -a
- 2) 进入容器： docker exec -it 容器 ID/容器名 bash
- 3) 进入之后，所有指令和常规 Linux 终端指令无异

5.1 查看 MySQL 信息

- 1) 查找名称中包含 mysql 的容器： docker ps -a | grep mysql

根据返回值确定容器 ID 为： aa4cc27d65dd

- 2) 进入 mysql 容器： docker exec -it aa4cc27d65dd bash
- 3) 输入： env

即可返回 mysql 相关信息，例如下图 mysql 中 root 用户的密码为 12345678QWE

```
Last login: Mon Jul 24 17:37:08 2023 from 192.168.5.2
[root@192 ~]# docker ps -a | grep mysql
5f8c2d559112 192.168.101.58:5000/mysql_d_exporter:v20.0.1-0.13.0 "/bin/mysqld_exporter"
2 hours ago Up 2 hours 192.168.101.58:6104->9104/tcp
fanruan230724160433_mysqld_exporter
aa4cc27d65dd 192.168.101.58:5000/mysql:v20.2.2-5.7.42 "docker-entrypoint.s..."
2 hours ago Up 2 hours 33060/tcp, 192.168.101.58:6306->3306/tcp
fanruan230724160433_mysql
[root@192 ~]# docker exec -it aa4cc27d65dd bash
bash-4.2# env
HOSTNAME=aa4cc27d65dd
TERM=xterm
MYSQL_VERSION=5.7.42-1.el7 mysql版本
MYSQL_DATABASE=finedb 数据库名称
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
PWD=/
MYSQL_SHELL_VERSION=8.0.33-1.el7
TZ=Asia/Shanghai
SHLVL=1
HOME=/root
MYSQL_MAJOR=5.7
RUNTIME_CMD_mysql=
RUNTIME_PATHMAPPING_mysql=/var/lib/mysql:data;/etc/mysql/conf.d:conf;/var/log/mysql:log
GOSU_VERSION=1.16
MYSQL_ROOT_PASSWORD=12345678QWE root用户密码
RUNTIME_ENV_mysql_2=MYSQL_ROOT_PASSWORD=${MYSQL_ROOT_PASSWORD}
RUNTIME_ENV_mysql_0=TZ=Asia/Shanghai
RUNTIME_ENV_mysql_1=MYSQL_DATABASE=${MYSQL_DATABASE}
_/usr/bin/env
bash-4.2#
```

5.2 查看 elasticsearch 密码

1) 查找名称中包含 elasticsearch 的容器： docker ps -a | grep elasticsearch

根据返回值确定容器 ID 为： ae590e473ce0

2) 查看该容器外挂目录 secrets 文件夹的位置信息：

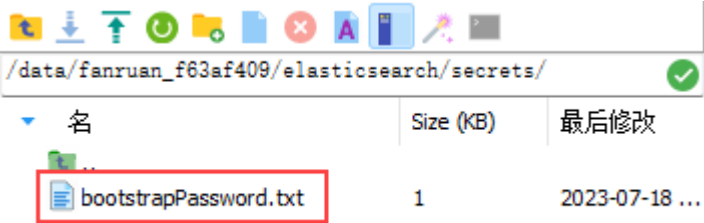
语句： docker inspect 容器 ID |grep elasticsearch/secrets

示例： docker inspect ae590e473ce0 |grep elasticsearch/secrets

根据返回值确定密码存放位置为： /data/fanruan\_f63af409/elasticsearch/secrets

```
[root@iz28x3egfhlz ~]# docker ps -a | grep elasticsearch
ae590e473ce0 120.27.40.246:5000/elasticsearch:v20.1.4-8.4.1 "/bin/tini -- /usr/l..." 2 months ago
Up 2 months 120.27.40.246:9200->9200/tcp, 9300/tcp fanruan230718201150_elasticsearch
ddb925b5eed5 120.27.40.246:5000/elasticsearch:v20.0.3-8.4.1 "/bin/tini -- /usr/l..." 3 months ago
Up 3 months 9300/tcp, 120.27.40.246:6200->9200/tcp fanruan230719153907_elasticsearch
8b05c79067e4 120.27.40.246:5000/elasticsearch_exporter:v20.0.0-1.5.0 "/bin/elasticsearch_..." 5 months ago
Up 5 months 7979/tcp, 120.27.40.246:6114->9114/tcp fanruan230719153907_elasticsearch_exporter
18bcd48fc235 elasticsearch_exporter:v20.0.0-1.5.0 "/bin/elasticsearch_..." 5 months ago
Up 5 months 7979/tcp, 120.27.40.246:9114->9114/tcp fanruan230718201150_elasticsearch_exporter
[root@iz28x3egfhlz ~]# docker inspect ae590e473ce0 |grep elasticsearch/secrets
"/data/fanruan_f63af409/elasticsearch/secrets:/run/secrets:rw",
"Source": "/data/fanruan_f63af409/elasticsearch/secrets",
```

3) 进入服务器密码存放位置，查看该位置下的 bootstrapPassword.txt 文件，里面存放的即为 elasticsearch 密码



6. 查看容器日志

当 FineOps 出现故障时，无法从前端查看各个应用和组件的日志。

本节介绍两种查看容器日志的方法。

6.1 在线实时查看 OPS 日志

1) 查找名称中包含 ops 的容器： docker ps -a | grep ops

根据返回值确定 ops 容器 ID 为： 7b396943b0da

2) 实时监控容器的日志： docker logs -f 7b396943b0da

```
[root@192 ~]#
[root@192 ~]# docker ps -a | grep ops
1d3fc721d927 ops_agent:v1.8.0-feature "/docker-entrypoint..." 3 days ago Up 3 days fanruan230815105759_ops_agent
192.168.101.58:9070->9070/tcp
7b396943b0da ops:v1.8.0.47-feature "/docker-entrypoint..." 3 days ago Up 2 days (healthy) fanruan230815105759_ops
192.168.101.58:8081->8080/tcp
[root@192 ~]# docker logs -f 7b396943b0da
no setenv.sh file
Using CATALINA_BASE: /usr/local/tomcat
Using CATALINA_HOME: /usr/local/tomcat
Using CATALINA_TMPDIR: /usr/local/tomcat/temp
Using JRE_HOME: /opt/java/openjdk
Using CLASSPATH: /usr/local/tomcat/bin/bootstrap.jar:/usr/local/tomcat/bin/tomcat-juli.jar
Using CATALINA_OPTS: -Dappname=webroot -Dtomcat.maxThreads=500 -Dtomcat.acceptCount=100
To enable GC log rotation, use -Xloggc:<filename> -XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=<num_of_files>
where num_of_file > 0
GC log rotation is turned off
```



## 6.2 下载 OPS 日志

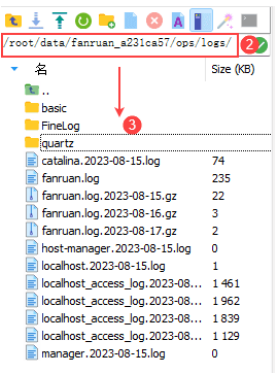
1) 查找名称中包含 ops 的容器：docker ps -a | grep ops

根据返回值确定 ops 容器 ID 为：7b396943b0da

2) 查看 ops 容器的配置：docker inspect 7b396943b0da

```
[root@192 ~]# docker ps -a | grep ops
1d3fc721d927      ops_agent:vi.8.0-feature      "/docker-entrypoint..." 3 days ago Up 3 days
192.168.101.58:9070->9070/tcp      fanruan230815105759_ops_agent
7b396943b0da      ops:vi.8.0.47-feature      "/docker-entrypoint..." 3 days ago Up 2 days (healthy)
192.168.101.58:8081->8080/tcp      fanruan230815105759_ops
[root@192 ~]# docker inspect 7b396943b0da
[
  {
    "Id": "7b396943b0daab09a5a0f7839c6863282e124ff3c279ad6f4f9faf0c45240c5a",
    "Created": "2023-08-15T03:00:54.479960911Z",
    "Path": "/docker-entrypoint.sh",
    "Args": [
      "catalina.sh",
      "run"
    ]
  }
]
```

3) 在返回值中，即可在 Binds 中查看到 logs 日志的存储位置，自行下载查看即可。



```
"RestartCount": 0,
"Driver": "overlay2",
"Platform": "linux",
"MountLabel": "",
"ProcessLabel": "",
"AppArmorProfile": "",
"ExecIDs": null,
"HostConfig": {
  "Binds": [
    "/root/data/fanruan_a231ca57/ops/assets:/usr/local/tomcat/webapps/ops/WEB-INF/assets",
    "/root/data/fanruan_a231ca57/ops/treasures:/usr/local/tomcat/webapps/ops/WEB-INF/treasures",
    "/root/data/fanruan_a231ca57/ops/embed:/usr/local/tomcat/webapps/ops/WEB-INF/embed",
    "/usr/bin/docker:/usr/bin/docker",
    "/var/run/docker.sock:/var/run/docker.sock",
    "/root/data/fanruan_a231ca57/ops/resources:/usr/local/tomcat/webapps/ops/WEB-INF/resources",
    ① "/root/data/fanruan_a231ca57/ops/logs:/usr/local/tomcat/logs"
  ],
  "ContainerIDFile": "",
  "LogConfig": {
    "Type": "local",
    "Config": {
      "compress": "true",
      "max-file": "5",
      "max-size": "20m"
    }
  },
  "NetworkMode": "default",
  "PortBindings": {
    "6379/tcp": [
      {
        "HostIp": "192.168.101.58",
        "HostPort": "6379"
      }
    ]
  },
  "RestartPolicy": {
    "Name": "on-failure",
    "MaximumRetryCount": 10
  },
  "SecurityOpt": null
}
```

## 7. 其他

### 7.1 查看 Redis 密码

1) 查找名称中包含 redis 的容器：docker ps -a | grep redis

根据返回值确定容器 ID 为：47dadb9d756f

2) 查看该容器配置项：docker inspect 47dadb9d756f

即可确定 Redis 的外挂目录为：/data/bi6/fanruan230719153907/redis/data

```
"HostConfig": {
  "Binds": [
    ② "/data/bi6/fanruan230719153907/redis/data:/data"
  ],
  "ContainerIDFile": "",
  "LogConfig": {
    "Type": "local",
    "Config": {
      "compress": "true",
      "max-file": "5",
      "max-size": "20m"
    }
  },
  "NetworkMode": "default",
  "PortBindings": {
    "6379/tcp": [
      {
        "HostIp": "192.168.101.58",
        "HostPort": "6379"
      }
    ]
  },
  "RestartPolicy": {
    "Name": "on-failure",
    "MaximumRetryCount": 10
  },
  "SecurityOpt": null
}
```

3) 进入/data/bi6/fanruan230719153907/redis/data，打开该目录下的 redis.conf 文

件

即可查看 Redis 的用户密码为 admin123

```
dir /data
replica-serve-stale-data yes
replica-read-only yes
repl-diskless-sync no
repl-diskless-sync-delay 5
repl-disable-tcp-nodelay no
replica-priority 100
requirepass admin123
lazyfree-lazy-eviction no
lazyfree-lazy-expire no
lazyfree-lazy-server-del no
replica-lazy-flush no
```

7.2 迁移拷贝文件

迁移工程时，经常需要拷贝文件夹。直接复制粘贴非常耗费时间，建议通过 cp 语句进行拷贝。

语句：

拷贝文件：scp /本地文件夹路径/文件名.后缀 远程 user@host:/远程目标路径

拷贝文件夹：scp -r /本地文件夹路径 远程 user@host:/远程目标路径

示例：

将「192.168.101.58 服务器/usr/local/tomcat/webapps/webroot/WEB-INF/lib 文件夹」，拷贝到「192.168.101.108 服务器/data 文件夹」下

请提前获取 192.168.101.108 服务器中，有增删目录文件的用户名和密码，此示例假定用户为 user

- 1) 打开终端，使用有拷贝文件权限的用户，访问 192.168.101.58 服务器（源服务器）
- 2 ) 执 行 命 令 ： scp -r /usr/local/tomcat/webapps/webroot/WEB-INF/lib user@192.168.101.108:/data/
- 3) 执行命令后，可能会提示您输入目标服务器的用户密码。输入密码后，复制过程将开始。
- 4) 等待复制完成。复制的时间取决于文件夹的大小和网络传输速度。

7.3 拷贝文件到容器内

配置工程 ssl 时，往往需要将证书文件上传到 nginx 中，建议通过 docker cp 语句进行拷贝。

容器 ID/容器名称请根据第二章方法查找。

**语句：**

docker cp 源文件路径 容器名:目标路径

**示例：**

- 将当前机器上/data 目录下的 a.txt 拷贝到容器内的  
/usr/local/tomcat/webapps/webroot/WEB-INF/lib 下

docker cp /data/a.txt 容器名:/usr/local/tomcat/webapps/webroot/WEB-INF/lib



## 4. 部署迁移升级项目

## 4.1 部署新项目

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.5.5	-
V1.8.0	1) 部署新项目入口位置调整，更醒目 2) 支持部署新 FineDataLink 集群项目
V1.9.0	1) 部署交互界面优化 2) 部署失败再次部署，支持自动填充信息

#### 1.2 功能简介

一个运维平台支持对接多个 FineReport、FineBI、FineDataLink 应用。

用户希望通过运维平台界面化部署 FineReport、FineBI、FineDataLink 应用。

[helpvideo]4192[/helpvideo]

### 2. 快速部署方案

本节介绍最简单的部署方案，因此对服务器相关要求较高，如需了解详细方案，请参考第三章索引。

容器化部署需要基于运维平台前端进行, 因此用户需要先部署一个 FineOps 运维平台。

步骤	说明
单机	1) 准备 1 台全新服务器，服务器不部署其他工程或内容，所有端口均未被占用
准备	2) 系统版本：CentOS 7.3~7.9 版本 / Ubuntu 18.04.4 及以上版本/ redhat 7.6 及以上版本
1 服	3) 系统内核：3.10 及以上，fr/bi 最低 8 核，fdl 最低 4 核
务	4) 系统架构：X86_64
器	5) 剩余空闲磁盘：使用命令「df -h」查询，服务器/root/data 目录所在分区剩余空闲磁盘大于：fr/fdl 50 G，bi 100G

6) 剩余空闲物理内存: fr 16G 以上、bi 32G 以上、fdl 8G 以上

7) 网络要求:

- 所有服务器与运维平台所在服务器内网互通
- 对外开放 8080 端口, 确保用户可通过 8080 端口访问帆软应用

8) 用户: 使用 root 用户完成部署

## 集群

1) 按照「工程节点数+1」准备全新服务器数量, 服务器不部署其他工程或内容, 所有端口均未被占用

2) 系统版本: CentOS 7.3~7.9 版本 / Ubuntu 18.04.4 及以上版本/ redhat 7.6 及以上版本

3) 系统内核: 3.10 及以上, frbi 最低 8 核, fdl 最低 4 核, 集群组件服务器最低 4 核

4) 系统架构: X86\_64

5) 剩余空闲磁盘: 使用命令「df -h」查询, 每台服务器/root/data 目录所在分区剩余空闲磁盘大于:

- fr/fdl 50 G, bi 100G, 集群组件服务器 50 G
- fdl 如需使用管道功能, 集群组件服务器剩余空闲磁盘建议大于 1T

6) 剩余空闲物理内存:

fr 16G 以上、bi 32G 以上、fdl 8G 以上、集群组件服务器 8G 以上

7) 网络要求:

- 所有服务器与运维平台所在服务器内网互通
- 集群组件服务器对外开放 80 端口, 确保用户可通过 80 端口访问帆软应用

8) 用户: 使用 root 用户完成部署

FDL 额外：

联系帆软技术支持，获取 FineDataLink、Kafka 相关镜像包

1) 管理员登录 FineOps 运维平台，点击「运维管理>项目管理」，点击「部署新项目」



2) 进入「项目设置」界面，设置以下信息，点击「下一步」

自定义项目名称、选择应用类型、全选自定义部署组件、其他无需变更

部署帆软项目



3) 进入「节点配置」

单机新增 1 个主应用节点，集群按照工程数量新增主应用节点和 1 个组件节点。

输入主机、端口、用户名密码，挂载路径无需修改。



4) 点击「开始部署」，进入部署过程。



通过运维平台成功部署的新项目， 自动接入运维平台的项目管理列表中

其中的访问地址， 即为该帆软项目的访问地址， 在浏览器新窗口访问该地址，  
即可进入工程初始化配置界面

访问帆软项目

3



### 3. 详细操作索引

运维平台支持部署各类帆软应用， 具体如下表所示：

帮助文档

部署内容

<a href="#">部署新 FineReport 单机项目</a>	运维平台支持容器化部署 FineReport 单机/集群工程
<a href="#">部署新 FineReport 集群项目</a>	FineReport 是一款用于报表制作，分析和展示的工具，用户可以轻松的构建出灵活的数据分析和报表系统，缩短项目周期，减少实施成本，解决企业信息孤岛问题。
<a href="#">部署新 FineBI 单机项目</a>	运维平台支持容器化部署 FineBI 单机/集群工程
<a href="#">部署新 FineBI 集群项目</a>	FineBI 旨在帮助企业的业务人员充分了解和利用他们的数据，用户只需简单拖拽便能制作出丰富多样的数据可视化信息，自由地对数据进行分析和探索。
<a href="#">部署新 FineDataLink 单机项目</a>	运维平台支持容器化部署 FineDataLink 单机/集群工程
<a href="#">部署新 FineDataLink 集群项目</a>	FineDataLink 数据平台，从高效专业的数据处理角度出发，帮助企业处理出质量更高、更利于展示与分析的数据。

## 4.1.1 部署新 FineReport 单机项目

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.5.5	-
V1.5.6	部署过程中，不再安装冗余组件 promtail、rocketmq_namesrv、rocketmq_broker
V1.6.0	1) 单机部署时，若选择安装 elasticsearch，同步安装 filebeat 组件
	2) 部署项目时，工程外挂目录新增 help 文件夹
V1.9.0	部署交互界面优化
V1.10.0	部署进度界面优化，细分展示镜像拉取和组件部署进度
V1.11.0	1) 项目配置中，项目名称限制放宽，新增支持英文大写、中文
	2) 项目配置中，访问路径支持为空
V1.14.0	部署的配置库组件，由 MySQL5 调整为 MySQL8
V1.15.0	1) 节点配置时，支持填写外网 IP，以适配不支持内网访问、仅可外网映射的服务
	器
	2) 支持在部署过程中「中止部署」，自动停止本次部署并回退已有操作

#### 1.2 功能简介

运维平台支持部署并直接接入全新 FineReport、FineBI、FineDataLink 应用。

本文将介绍，如何通过 [运维平台](#)，容器化部署全新的 **FineReport 单机**项目。

注 1：本文面向服务器资源不足或端口限制较高的用户，操作步骤较为详细，篇幅较长。

如能准备充足资源的全新服务器，可参考快速指南进行部署：[部署新帆软项目](#)

注 2：请在容器化部署前，先部署好 FineOps 运维平台。安装步骤请参见：[部署运维平台](#)。

### 2. 准备服务器

#### 2.1 确认需要安装的组件

通过运维平台容器化部署的 FineReport 单机项目，不仅仅包括 FineReport 工程，还包括配套的外接配置库、运维组件等等。

请先确认好需要安装的组件。本文按照安装全部组件来进行示例讲解。

服务 器	安装的组 件	是否 安装	说明
	FineRepo rt	必须 安装	<p>作用：FineReport 的工程节点组件，其中不带任何的 demo 数据</p> <p>说明：必须安装</p>
	ops_agen t	必须 安装	<p>作用：为服务器物理节点配套安装，用于管理组件，容器化升级也是由该组件完成</p> <p>说明：FineReport 项目涉及的每个服务器都会安装一个 ops_agent</p>
主应 用服 务器	node_exp orter	必须 安装	<p>作用：服务器物理节点的 exporter，收集物理服务器的各种指标信息</p> <p>说明：FineReport 项目涉及的每个服务器都会安装一个 node_exporter</p>
	filebeat	可选安 装	<p>作用：日志采集</p> <p>说明：与 elasticSearch 组件配套安装</p> <p>如不安装 elasticsearch 组件，则不安装 filebeat 组件</p> <p>如安装 elasticsearch 组件，则同步在 FineReport 工程节点所在服务器安装 filebeat 组件</p> <p>作用：作为工程的外接配置库</p> <p>说明：可选安装</p> <p>如果安装该组件，会安装一个 MySQL8 数据库，作为工程外接配置库</p>
组件 服务 器	mysql	可选 安装	<p>如不安装该组件，用户必须自行准备好一个外接配置库，否则无法部署</p> <p>支持的数据库类型包括：MySQL5、MySQL8、Oracle（V1.14.0 及之前版本不支持 MySQL8）</p> <p>如需使用其他类型外接库，可在项目部署成功后，登录工程切换</p>



mysqld_exporter	可选	作用：MySQL 服务的 exporter，收集 MySQL 的各种指标信息
	安装	说明：与 MySQL 组件配套安装
elasticsearch		作用：日志存储
		说明：可选安装
	可选安装	如果安装该组件，则取代 FineReport 中原有的 swift，作为日志存储
		如不安装该组件，将启用 FineReport 中原有的 swift（logdb）进行日志存储
elasticsearch_exporter	可选安装	作用：elasticSearch 的 exporter，收集 elasticSearch 的指标
		说明：与 elasticSearch 组件配套安装
node_exporter		作用：服务器物理节点的 exporter，收集物理服务器的各种指标信息
		说明：
	必须安装	如果准备了组件服务器，则自动安装一个 node_exporter 组件
		如果不安装 mysql 和 elasticsearch 套件，则无需准备组件服务器，则不会安装该组件

## 2.2 服务器数量要求

本文按照 2 个服务器进行下文的部署操作，分为主应用服务器 A、组件服务器 B。

请准备好服务器的内网 IP、用户名、密码。

注：FineOps 运维平台，和 FineReport 项目，不要部署在同一服务器下。

如果 FineOps 和项目处于同一环境下,当项目服务器发生异常时,FineOps 可能也出现故障,从而无法起到运维的作用。

类型	要求
	建议准备 <b>2 个</b> 服务器，这两个服务器中不再安装其他工程/产品
推荐配置	<ul style="list-style-type: none"><li>工程独占一个服务器：即主应用服务器，包括 FineReport、ops_agent</li><li>其他组件独占一个服务器：即组件服务器，如果不安装 mysql 和 elastic search 套件，则无需准备该服务器</li></ul>
最低配置	至少准备 <b>1 个</b> 服务器，这个服务器中不再安装其他工程/产品

- 整个 FineReport 项目独占一个服务器（只有 1 个服务器的情况下，所有组件均安装在该服务器中）

2.3 服务器网络要求

请确保 FineReport 项目涉及的主应用服务器 A、组件服务器 B，以及 FineOps 运维平台所在的服务器 C：

这些服务器内网互相联通，如不联通则无法成功部署。

注：如自行准备了外接配置库，外接配置库所在的服务器 D，也需要和 ABC 服务器内网互通。

2.4 服务器系统配置

请确保 FineReport 项目涉及的主应用服务器 A、组件服务器 B，满足此基础配置。

服务器	配置	要求
主应用服务器	系统版本	CentOS 7.3~7.9 版本
		Ubuntu 18.04.4 及以上版本
		redhat 7.6 及以上版本
	系统内核	3.10 及以上
		最低 8 核
	CPU 架构	仅支持 X86_64
	剩余空闲物理内存	16G 以上
	剩余空闲磁盘容量	50GB 以上
	组件服务器	系统版本
Ubuntu 18.04.4 及以上版本		
redhat 7.6 及以上版本		
系统内核		3.10 及以上
		最低 4 核
CPU 架构		仅支持 X86_64
剩余空闲物理内存		8G 以上
剩余空闲磁盘容量		50GB 以上

注：服务器之所以要满足上表所说的内存和磁盘，是由于 FineReport 项目中的各个组件有最低安装配置要求。

点击展开更多

服务器	组件	CPU	内存	xmx	磁盘
主应用服务器	FineReport	8Core	-	默认内存为：	50G

机器内存减掉 8

G 后的内存大小

最大不会超过 3

2G

组件服务器	ops_agent	1Core	-	1G	-
	node_exporter	0.1Core	0.1G	-	-
	filebeat	0.1Core	0.1G	-	-
	mysql	2Core	-	-	-
	mysqld_exporter	0.1Core	0.1G	-	-
	elasticsearch	1Core	-	2G	50G
	elasticsearch_expo	0.1Core	0.1G	-	-
	rtter				
	ops_agent	1Core	-	1G	-
	node_exporter	0.1Core	0.1G	-	-

2.5 服务器端口要求

请确保 FineReport 项目涉及的主应用服务器 A、组件服务器 B 中，待部署应用和组件的默认端口可用。

- 请确认默认的映射端口是否已被使用，如已被使用，请安排好其他空闲端口。
- FineReport 的映射端口将作为项目访问端口，因此请确认准备使用的 FineReport 映射端口是否开放了防火墙、安全组等，确保可连接。

注 1：端口占用和防火墙开放操作方法可参考：[端口占用检查与防火墙配置](#)。  
注 2：端口 portMapping 为 {映射端口:容器内端口}，其中映射端口可改, 容器内端口不支持修改。

服务器	组件	默认端口
主应用服务器	FineReport	端口：{8080: 8080, 12100: 12100}
	ops_agent	端口：{9070:9070}
	node_exporter	端口：{9100:9100}
	filebeat	不占用端口
组件服务器		端口：{3306:3306}
	mysql	用户名：root（不支持修改） 密码：随机生成，届时请务必修改
	mysqld_exporter	端口：{9104:9104}

	端口： {9200:9200}
elasticsearch	用户名： elastic （不支持修改）
	密码： elasticadmin （支持修改）
elasticsearch_exporter	端口： {9114:9114}
ops_agent	端口： {9070:9070}
node_exporter	端口： {9100:9100}

2.6 设置项目挂载目录

容器化部署的 FineReport 项目，重要文件均外挂。请提前在主应用服务器 A 和组件服务器 B 的空闲磁盘下，新建好挂载目录。

1) 查看服务器磁盘空间

命令：df -h，找出可用空间较大的挂载点

例如本示例中，服务器最合适的挂载目录为/home 目录下

```
Last login: Tue Jul 25 10:16:49 2023 from 192.168.5.2
[root@192 ~]# df -h
文件系统          容量  已用  可用 已用% 挂载点
/dev/mapper/centos-root    50G   14G   37G   28% /
devtmpfs              9.8G    0 9.8G    0% /dev
tmpfs                  9.8G    0 9.8G    0% /dev/shm
tmpfs                  9.8G  962M  8.8G   10% /run
tmpfs                  9.8G    0 9.8G    0% /sys/fs/cgroup
/dev/mapper/centos-home   142G   12G  130G    9% /home
/dev/sda1                1014M  146M  869M   15% /boot
tmpfs                   2.0G    0  2.0G    0% /run/user/0
```

2) 新建文件夹

- 使用 mkdir 命令创建文件夹（本示例在/home 目录下创建一个名叫 fanruan 的文件夹）：mkdir /home/fanruan
- 使用 chmod 命令设置文件夹的权限（本示例直接赋 777 权限）：  
chmod 777 /home/fanruan

```
[root@192 ~]#
[root@192 ~]# mkdir /home/fanruan
[root@192 ~]# chmod 777 /home/fanruan
[root@192 ~]#
[root@192 ~]#
[root@192 ~]# cd /home
[root@192 home]# ls
fanruan  finekey  finekey-operation-all.tar.gz  fr
[root@192 home]#
```

注：FineReport 单机项目部署成功后，项目挂载目录中的 fr 文件夹，存放着工程外挂内容，其中的文件夹和存放内容如下表所示。

点击展开更多

外挂文件夹	说明
-------	----

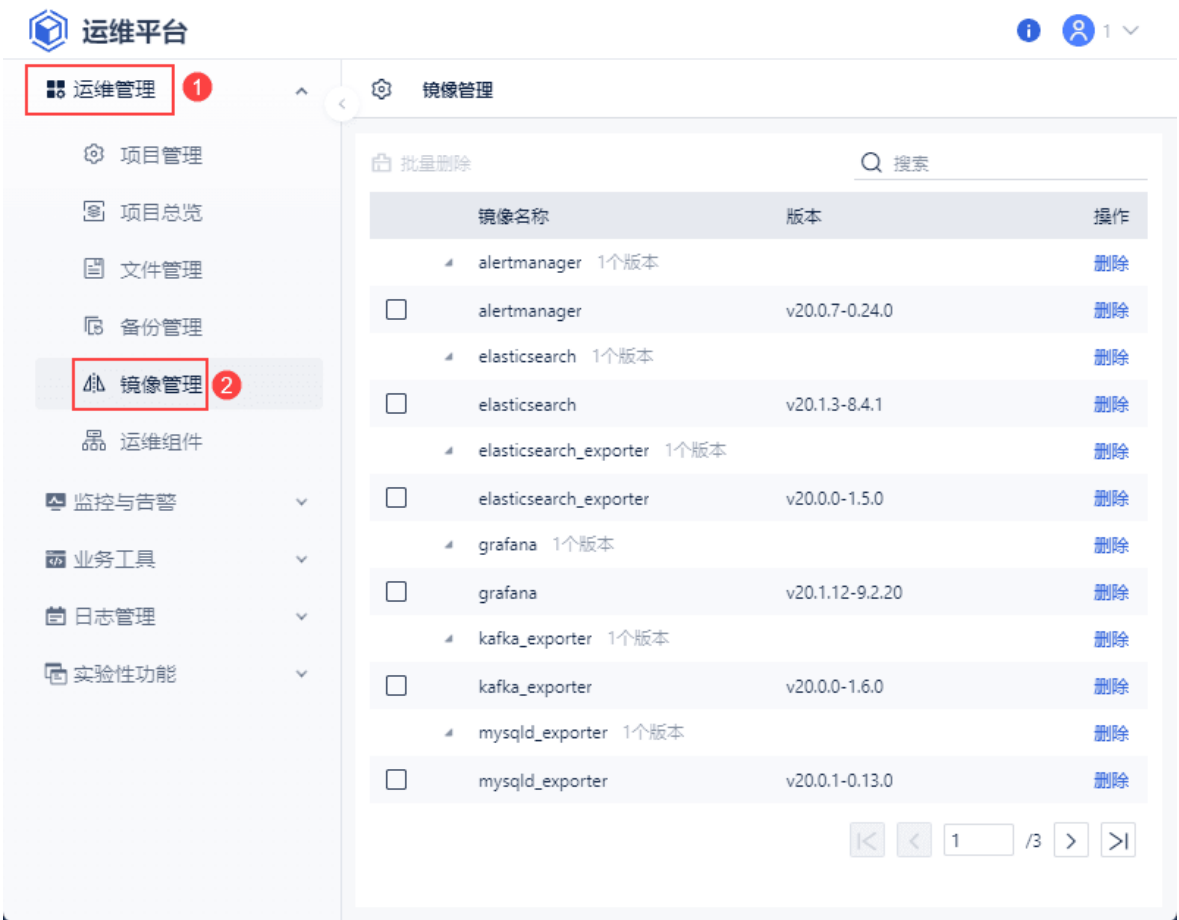
	temp_attach: 读写缓存存储路径
assets	vcs: FineReport 模板备份文件
	其他文件: 通用的共享持久化目录
backup	工程历史备份文件
classes	工程调用的默认和自定义 class 文件
config	平台调用的配置库 (finedb) 相关文件
customlib	工程定制/外部引入的 JAR 包
dpworks	FineDataLink 任务相关的配置文件 (如不存在该文件夹, 说明工程没有使用 FDL 相关功能)
help	工程需要调用的一些 css、js、html、图片等资源
local	存放新引擎的编译文件
logs	swift 日志 (即 logdb)
mount	存放 setenv.sh 文件, 用于设置工程的环境变量和启动参数
plugins	插件相关文件
reportlets	FineReport 模板存放目录
resources	存放工程相关的资源配置文件
schedule	定时调度生成的文件
tomcat_logs	Tomcat 通用日志
treasures	云端运维生成的数据包

### 3. 确认镜像仓库

部署新项目中的各个组件, 是需要通过镜像进行部署的。要么镜像仓库已有镜像, 要么支持从云端拉取镜像。

1) 管理员登录运维平台, 点击「运维管理>镜像管理」, 查看是否存在相关镜像。

- 必须安装: FineReport、node\_exporter、ops\_agent
- 可选安装: mysql、mysqld\_exporter、elasticsearch、elasticsearch\_exporter、filebeat



2) 如镜像仓库中不存在相关镜像，请查看镜像仓库地址，并确保地址可连通外网云端仓库。

- 查看镜像仓库地址：管理员登录运维平台，点击「运维管理>镜像管理>全局配置」，即可查看镜像仓库地址。
- 确保镜像仓库所在服务器支持访问以下外网地址：

域名：registry.cn-hangzhou.aliyuncs.com，端口：443

域名：fineops-registry.cn-hangzhou.cr.aliyuncs.com，端口：443

即可在部署项目时自动从云端拉取最新项目组件镜像。



#### 4. 部署新 FineReport 单机项目

本节正式进入部署 FineReport 单机项目的操作环节。请在开始部署前务必确认上文的所有内容，否则可能会部署失败！

##### 4.1 新增项目

管理员登录 FineOps 运维平台，点击「运维管理>项目管理」，点击「部署新项目」，如下图所示：



##### 4.2 项目设置

项目设置是配置项目的类型、需要安装的组件、占用的端口等信息。

进入「项目设置」界面，输入相关项目信息，点击「下一步」，如下图所示：

部署新项目

1 项目设置

2 节点配置

项目名称

1

fanruan1

应用类型

2

FineReport

FineBI

FineDataLink

项目类型

集群

3

单机

默认部署组件

☒ 运维基础套件

自定义部署组件

4

☒ 外接数据库(MySQL)

☒ ElasticSearch

部署配置

☐ 使用默认配置

☒ 自定义端口、组件用户名密码

去设置

访问路径

webroot

配置后应用的访问链接为{域名/IP}/{访问路径}/decision

取消

5

下一步

各配置项如下表所示：

信息	说明
自定义项目的名称	
项目名称不允许为空	
项目名称	项目名称仅支持中文、英文大小写、数字
	项目名称长度最大为 12 位
	项目名称不允许与已有项目重名
应用类型	选择「FineReport」
应用类型	选择「单机」
默认部署组件	即必须安装的组件，包括： FineReport、node_exporter、ops_agent
	即可选安装的组件，包括：
	1) 外接数据库
自定义部署组件	mysql、mysqld_exporter
	如不安装该组件，用户必须自行准备好一个外接配置库，否则无法部署
	支持的数据库类型包括：MySQL5、MySQL8、Oracle（V1.14.0 及之前版本不支持 MySQL8）



如需使用其他类型外接库，可在项目部署成功后，登录工程切换

2) Elasticsearch

elasticsearch、elasticsearch\_exporter、filebeat

如不安装该组件，将启用 FineReport 中原有的 swift (logdb)，用作日志存储

支持两种：

- 1) 使用默认配置
- 2) 自定义端口、组件用户名密码，即对 2.5 节的默认端口、密码进行调整  
请务必修改并记住 MySQL、elasticsearch 组件的密码，该密码默认随机生成

部署新项目

1 项目设置2 节点配置

部署配置

项目名称fanruan1

应用类型FineReportFineBIFineDataLink

应用类型集群单机

默认部署组件运维基础套件

自定义部署组件外接数据库(MySQL)ElasticSearch

部署配置使用默认配置自定义端口、组件用户名密码去设置

访问路径webroot  
配置后应用的访问链接为{域名/IP}/{访问路径}/decision

部署配置自定义

FineReport

主机端口8080设置端口8080

主机端口12100设置端口12100

mysql

主机端口3306设置端口3306

用户名root密码\*\*\*\*\*

mysql\_exporter

主机端口9104设置端口9104

elasticsearch

主机端口9200设置端口9200

取消完成

取消下一步

说明：

应用的访问链接为{域名/IP}/{访问路径}/decision

支持用户自定义应用的访问路径，访问路径默认为 webroot

举例：

一般应用访问链接为 http://IP:端口/webroot/decision

访问

路径 修改访问路径为 demo

则部署好的应用访问链接为 http://IP:端口/demo/decision

注意事项：

1.11.0 及之后版本，访问路径支持为空，即最终应用访问链接为 http://IP:端口/decision

但此时工程对应的 root 目录访问路径为：http://IP:端口/root

4.3 节点配置-新增主应用节点

配置完需要的组件和端口后，即可进入「节点配置」。

首先需要新增主应用节点，即为添加主应用服务器 A 的相关信息。

1) 点击「新增节点」，选择节点类型为「主应用」，填写相关设置项，点击「测试连接」。



具体节点设置如下图所示：

节点设置	说明
节点类型	请选择「主应用」 主应用节点要求最低配置 8 核 16G、磁盘剩余 50GB 以上 注 1：如未准备组件服务器，那么所有组件都将部署在该主应用节点下 注 2：帆软项目需要独占服务器，即该主应用服务器上不建议部署其他工程或内容，包括运维平台
主机	请输入节点主机 IP（内网 IP） 注：同一项目中，主机不支持重复添加多次
端口	请输入节点端口号，默认为 22
用户名	请输入有 sudo 权限的服务器用户名
验证方法	支持两种：密码、公钥
挂载路径	请输入节点安装目录的服务器路径，即 2.6 节设置的目录 默认为~/data，~代表所使用的服务器用户的主目录 注：可在终端使用该用户账号访问服务器，输入 echo \$HOME 命令查看该用户的主目录路径
外网 IP	选填项

如果服务器不支持内网访问，仅支持外网映射，可以填写可联通的外网 IP

2) 效果预览

若主机用户名等信息校验成功，即可添加一个主应用节点。如下图所示：

- 注 1：若网络环境不满足 2.3 节要求，节点连接失败，则出现报错提醒，无法正确新增节点。
- 注 2：若服务器配置不符合 2.4 节最低要求，则出现报错提醒，无法正确新增节点。



4.4 节点配置-新增组件节点

接着要新增组件节点，即为添加组件服务器 B 的相关信息。如未准备组件服务器，无需执行本章操作，所有组件都将部署在上一节主应用节点下。

- 1) 点击「新增节点」，选择节点类型为「组件」，填写相关设置项，点击「测试连接」。



具体节点设置如下图所示：

节点设置	说明
	请选择「组件」
节点类型	组件节点要求最低配置 4 核 8G、磁盘剩余 50GB 以上 注 1：如配置了多个组件节点，则组件随机部署在各个节点 注 2：帆软项目需要独占服务器，即该组件服务器上不建议部署其他工程或内容，包括运维平台
主机	请输入节点主机 IP（内网 IP） 注：同一项目中，主机不支持重复添加多次
端口	请输入节点端口号，默认为 22
用户名	请输入有 sudo 权限的服务器用户名
验证方法	支持两种：密码、公钥
	请输入节点安装目录的服务器路径，即 2.6 节设置的目录
挂载路径	默认为~/data，~代表所使用的服务器用户的主目录 注：可在终端使用该用户账号访问服务器，输入 echo \$HOME 命令查看该用户的主目录路径
	选填项
外网 IP	如果服务器不支持内网访问，仅支持外网映射，可以填写可联通的外网 IP

2) 效果预览

若主机用户名等信息校验成功，即可添加一个组件节点。如下图所示：

- 注 1：若网络环境不满足 2.3 节要求，节点连接失败，则出现报错提醒，无法正确新增节点。
- 注 2：若服务器配置不符合 2.4 节最低要求，则出现报错提醒，无法正确新增节点。



### 4.5 开始部署

节点配置完成后，点击「开始部署」，进入部署过程。



弹窗中会显示各个组件的部署情况，若部署失败则显示失败原因。

- 注 1：部署过程中，用户可点击「隐藏至后台进行」，正常进行运维平台其他操作。
- 注 2：V1.15.0 及之后版本，支持在部署过程中点击「中止部署」按钮，停止部署并回退所有已有部署操作。

新项目部署

正在部署，请稍后...

需部署8个组件，已完成0个

mysql

部署中...

mysqld\_exporter

部署中...

elasticsearch

部署中...

elasticsearch\_expor...

部署中...

node\_exporter

部署中...

fr

部署中...

filebeat

部署中...

隐藏至后台进行

通过运维平台成功部署的新项目，自动接入运维平台的项目管理列表中，可进行相关运维操作。如下表所示：

其中的访问地址，即为该 FineReport 单机工程的访问地址，在浏览器新窗口访问该地址，即可进入工程初始化配置界面。

运维平台

运维管理

项目管理

项目总览

文件管理

备份管理

镜像管理

部署新项目

接入已有项目

超时检测

名称	类型	存活节点数	部署方式	状态	访问地址	操作
fanrua...	集群	2	非运...	可用	http://1...3/webroot/decision	🔍🔗🗑️⚙️
fanrua...	单机	1	非运维...	可用	http://1...3/webroot/decision	🔍🔗🗑️⚙️
fanru...	单机	1	运维...	可用	http://1...broot/decision	🔍🔗🗑️⚙️🔍

<<<

<

1

>

>>>

## 4.1.2 部署新 FineReport 集群项目

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.5.5	-
V1.5.6	部署过程中，不再安装冗余组件 promtail、rocketmq_namesrv、rocketmq_broker
	1) 为了确保集群日志存储的稳定性：
	容器化部署集群项目时，elasticsearch 套件不再可选安装，而是必须安装
V1.6.0	安装 elasticsearch 时，同步在每一个工程节点安装 filebeat 组件
	2) 部署项目时，工程外挂目录新增 help 文件夹
V1.9.0	部署交互界面优化
V1.10.0	部署进度界面优化，细分展示镜像拉取和组件部署进度
	1) 项目名称限制放宽，新增支持英文大写、中文
V1.11.0	2) 部署新集群项目时，支持使用外部共享目录作为文件服务器
	3) 项目配置中，访问路径支持为空
V1.14.0	部署的配置库组件，由 MySQL5 调整为 MySQL8
	1) 节点配置时，支持填写外网 IP，以适配不支持内网访问、仅可外网
V1.15.0	映射的服务器
	2) 支持在部署过程中「中止部署」，自动停止本次部署并回退已有操作

#### 1.2 应用场景

运维平台支持部署并直接接入全新 FineReport、FineBI、FineDataLink 应用。

本文将介绍，如何通过 [运维平台](#)，容器化部署全新的 **FineReport 集群**项目（本文示例三节点集群）。

注 1：本文面向服务器资源不足或端口限制较高的用户，操作步骤较为 **详细**，篇幅较长。

如能准备充足资源的**全新**服务器，可参考快速指南进行部署：[部署新帆软项目](#)

注 2：请在容器化部署前，先部署好 FineOps 运维平台。安装步骤请参见：[部署运维平台](#)。

### 2. 准备服务器

#### 2.1 确认需要安装的组件

通过运维平台容器化部署的 FineReport 集群项目，不仅仅包括 FineReport 工程，还包

括配套的外接配置库、集群组件、运维组件等等。

请先确认好需要安装的组件。本文按照安装全部组件来进行示例讲解。

服务器	组件	是否安装	组件说明
主应用服务器	FineReport	必须安装	作用：FineReport 的工程节点组件，其中不带任何的 demo 数据 说明：必须安装
	ops_agent	必须安装	作用：为服务器物理节点配套安装，用于管理组件，容器化升级也是由该组件完成 说明：FineReport 项目涉及的每个服务器都会安装一个 ops_agent
	node_exporter	必须安装	作用：服务器物理节点的 exporter，收集物理服务器的各种指标信息 说明：每一个主应用服务器，都会自动安装一个 node_exporter 组件
	filebeat	必须安装	作用：日志采集 说明：与 elasticSearch 组件配套安装，在每个 FineReport 工程节点所在服务器安装 filebeat 组件
			作用：日志存储 说明：替换 FineReport 中原有的 swift 作为日志存储
	elasticsearch	必须安装	对项目中 FineReport、外接数据库、业务库和其他集群组件的请求进行采集
			集群必须安装该组件，且请勿暂停该组件，否则会导致项目操作日志丢失
	elasticsearch_exporter	必须安装	作用：elasticSearch 的 exporter，收集 elasticSearch 的指标 说明：与 elasticSearch 组件配套安装
	nginx	必须安装	作用：提供集群的负载转发服务



	装	说明：集群其他组件服务的入口、使用，都从这个入口访问
nginx_exporter	必须安装	作用：nginx 服务的 exporter，收集 nginx 的指标信息
	装	说明：与 nginx 组件配套安装
		作用：作为集群的外接配置库
		说明：可选安装
		如果安装该组件，会安装一个 MySQL8 数据库，作为工程外接配置库
mysql	可选安装	如不安装该组件，用户必须自行准备好一个外接配置库，否则无法部署
		支持的数据库类型包括：MySQL5、MySQL8、Oracle (v1.14.0 及之前版本不支持 MySQL8)
		如需使用其他类型外接库，可在项目部署成功后，登录工程切换
mysqld_exporter	可选安装	作用：MySQL 服务的 exporter，收集 MySQL 的各种指标信息
	装	说明：与 MySQL 组件配套安装
		作用：作为集群的状态服务器
		说明：可选安装
redis	可选安装	如果安装该组件，会安装一个单机 redis，作为集群的状态服务器
		如不安装该组件，用户必须自行准备好一个 redis 组件（单机/集群均可），否则无法部署
redis_exporter	可选安装	作用：redis 服务的 exporter，收集 redis 的各种指标信息
	装	说明：与 redis 组件配套安装
		作用：作为集群的文件服务器
minio	可选安装	说明：可选安装
	装	如果安装该组件，会安装一个 minio，它是一款高性能、

		分布式的对象存储系统，作为集群的文件服务器
		如不安装该组件，用户必须自行准备好一个文件服务器，否则无法部署
		支持的文件服务器类型包括：SFTP、FTP、共享外部目录（V1.10.0 及之前版本不支持共享外部目录）
		如需使用其他类型文件服务器，可在项目部署成功后，登录工程切换
		作用：为服务器物理节点配套安装，用于管理组件，容器
ops_agent	必须安装	化升级也是由该组件完成
		说明：FineReport 项目涉及的每个服务器都会安装一个 ops_agent
		作用：服务器物理节点的 exporter，收集物理服务器的各种指标信息
node_exporter	必须安装	说明：每一个组件服务器，都会自动安装一个 node_exporter 组件

2.2 服务器数量要求

本文按照 4 个服务器进行下文的部署操作, 分为主应用服务器 ABC、组件服务器 D。

请准备好服务器的内网 IP、用户名、密码。

注：FineOps 运维平台，和 FineReport 项目，不要部署在同一服务器下。

如果 FineOps 和项目处于同一环境下, 当项目服务器发生异常时, FineOps 可能也出现故障, 从而无法起到运维的作用。

类型	要求
	建议按照「节点数量+1」准备服务器数量，这些服务器中不再安装其他工程/产品
	1) 每个工程节点独占一个服务器：
	<ul style="list-style-type: none"><li>例如本文示例的是 3 节点集群，则需要 3 个主应用服务器。</li></ul>
推荐配置	<ul style="list-style-type: none"><li>每个服务器中安装一套 FineReport、ops_agent、node_exporter、filebeat</li></ul>
	2) 其他组件独占一个服务器：即组件服务器
	<ul style="list-style-type: none"><li>该服务器中安装外接配置库（mysql）、集群组件（nginx、redis、minio）、es 组件（elasticsearch）以及运维组件（expoder 组件）</li></ul>

- 如果服务器资源充足，可以多准备一些组件服务器，在部署时会随机分配集群组件到各个服务器中，确保项目高可用

建议按照「节点数量+5」准备服务器数量，这些服务器中不再安装其他工程/产品

最优配置 1) 每个工程节点独占一个服务器，作为主应用服务器

2) redis、nginx、mysql、minio、es 组件各自独占一个服务器，作为组件服务器

2.3 服务器网络要求

请确保 FineReport 项目涉及的主应用服务器 ABC、组件服务器 D，以及 FineOps 运维平台所在的服务器 E：

这些服务器内网互相联通，如不联通则无法成功部署。

注：如自行准备了外接配置库、状态服务器、文件服务器。  
他们所在的服务器 F、G、H，也需要和 ABCDE 服务器内网互通。

2.4 服务器系统配置

请确保 FineReport 项目涉及的主应用服务器 ABC、组件服务器 B，满足此基础配置。

注意，下表的配置要求是针对每个服务器，而不是所有服务器的配置总和。

服务器	配置	要求
主应用服务器		CentOS 7.3~7.9 版本
	系统版本	Ubuntu 18.04.4 及以上版本
		redhat 7.6 及以上版本
		3.10 及以上
	系统内核	最低 8 核
	CPU 架构	仅支持 X86_64
	剩余空闲物理内存	16G 以上
	剩余空闲磁盘容量	50GB 以上
		CentOS 7.3~7.9 版本
	系统版本	Ubuntu 18.04.4 及以上版本
组件服务器		redhat 7.6 及以上版本
	系统内核	3.10 及以上

	最低 4 核
CPU 架构	仅支持 X86_64
	最低 8G
剩余空闲物理内存	若仅准备了一个 1 个组件服务器，建议 16G 以上
剩余空闲磁盘容量	50GB 以上

注：服务器之所以要满足上表所说的内存和磁盘，是由于 FineReport 项目中的各个组件有最低安装配置要求。

[点击展开更多](#)

服务器	组件	CPU	内存	xmx	磁盘
主应用服务器	默认内存为：				
	机器内存减掉 8G 后的内存大小				
	FineReport	8Core	-		50G
	最大不会超过 32G				
	ops_agent	1Core	-	1G	-
	node_exporter	0.1Core	0.1G	-	-
	filebeat	0.1Core	0.1G	-	-
	elasticsearch	1Core	-	2G	50G
	elasticsearch_exporter	0.1Core	0.1G	-	-
	nginx	1Core	-	-	-
	nginx_exporter	0.1Core	0.1G	0.1G	-
	mysql	2Core	-	-	50G
	mysqld_exporter	0.1Core	0.1G	-	-
	redis	1Core	-	4G	-
组件服务器	redis_exporter	0.1Core	0.1G	-	-
	minio	1Core	-	-	100G
	ops_agent	1Core	-	1G	-
	node_exporter	0.1Core	0.1G	-	-

## 2.5 服务器端口要求

请确保 FineReport 项目涉及的主应用服务器 ABC、组件服务器 D 中，待部署应用和组件的默认端口可用。

- 请确认默认的映射端口是否已被使用，如已被使用，请安排好其他空闲端口。

- nginx 的映射端口将作为项目访问端口，因此请确认准备使用的 nginx 映射端口是否开放了防火墙、安全组等，确保可连接。

注 1：端口占用和防火墙开放操作方法可参考：[端口占用检查与防火墙配置](#)。

注 2：端口 portMapping 为 {映射端口:容器内端口}，其中映射端口可改，容器内端口不支持修改。

注 3：如果准备了多个组件服务器，组件会随机分布部署在各个服务器下。因此请确保每一个组件服务器都满足各个组件的端口要求。

服务器	组件	默认端口
主应用服务器		端口：{8080: 8080, 7800: 7800, 7830: 7830, 7840: 7840, 7850: 7850, 7870: 7870}
	FineReport	7830, 7840: 7840, 7850: 7850, 7870: 7870}
	ops_agent	端口：{9070:9070}
	node_exporter	端口：{9100:9100}
	filebeat	不占用端口
		端口：{9200:9200}
	elasticsearch	用户名：elastic（不支持修改） 密码：elasticadmin（支持修改）
	elasticsearch_exporter	端口：{9114:9114}
		端口：{80: 8080}
	nginx	非 root 用户建议修改占用端口，因为非 root 用户默认无权限使用 1024 以下端口
组件服务器		端口：{9113: 9113}
		端口：{3306:3306}
	mysql	用户名：root（不支持修改） 密码：随机生成，届时请务必修改
	mysqld_exporter	端口：{9104:9104}
		端口：{6379:6379}
	redis	用户名：admin（不支持修改） 密码：随机生成，届时请务必修改
	redis_exporter	端口：{9121:9121}
		服务器端口：{9000:9000}
	minio	控制台端口：{9001:9001}

用户名：minioadmin（不支持修改）

密码：minioadmin（支持修改，但密码必须大于等于 8 位，否则使用默认密码）

ops\_agent                    端口：{9070:9070}

node\_exporter                端口：{9100:9100}

2.6 设置项目挂载目录

容器化部署的 FineReport 项目，重要文件均外挂。请提前在主应用服务器 ABC 和组件服务器 D 的空闲磁盘下，新建好挂载目录。

1) 查看服务器磁盘空间

命令：df -h，找出可用空间较大的挂载点

例如本示例中，服务器最合适的挂载目录为/home 目录下

```
Last login: Tue Jul 25 10:16:49 2023 from 192.168.5.2
[root@192 ~]# df -h
文件系统      容量  已用  可用 已用% 挂载点
/dev/mapper/centos-root  50G   14G   37G   28% /
devtmpfs      9.8G   0    9.8G   0% /dev
tmpfs         9.8G   0    9.8G   0% /dev/shm
tmpfs         9.8G  962M   8.8G   10% /run
tmpfs         9.8G   0    9.8G   0% /sys/fs/cgroup
/dev/mapper/centos-home 142G   12G  130G    9% /home
/dev/sda1     1014M  146M   869M   15% /boot
tmpfs         2.0G   0    2.0G   0% /run/user/0
```

2) 新建文件夹

- 使用 mkdir 命令创建文件夹（本示例在/home 目录下创建一个名叫 fanruan 的文件夹）：mkdir /home/fanruan
- 使用 chmod 命令设置文件夹的权限（本示例直接赋 777 权限）：  
chmod 777 /home/fanruan

```
[root@192 ~]#
[root@192 ~]# mkdir /home/fanruan
[root@192 ~]# chmod 777 /home/fanruan
[root@192 ~]#
[root@192 ~]#
[root@192 ~]# cd /home
[root@192 home]# ls
fanruan  finekey  finekey-operation-all.tar.gz  fr
[root@192 home]#
```

注：FineReport 集群项目部署成功后，文件服务器和项目挂载目录中的 fr 文件夹中，分别存放着工程外挂内容，其中的文件夹和存放内容如下表所示。

[点击展开更多](#)

存储位置	文件夹	说明
项目挂载目录的 fr 文件夹  (每个节点所在服务	classes	工程调用的默认和自定义 class 文件
	config	平台调用的配置库（finedb）相关文件
	customlib	工程定制/外部引入的 JAR 包

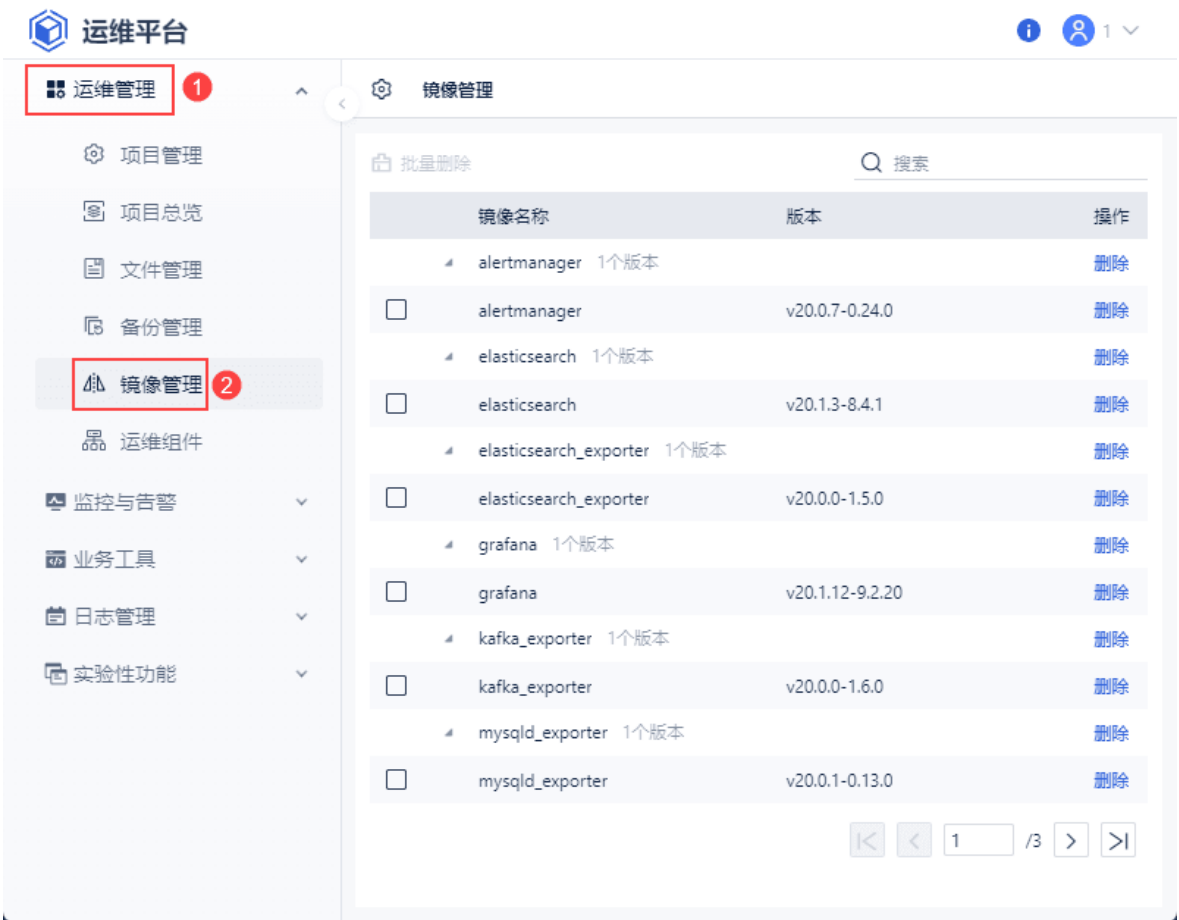
器都存了一份，完全相同)	dpworks	FineDataLink 任务相关的配置文件（如不存在该文件夹，说明工程没有使用 FDL 相关功能）
	help	工程需要调用的一些 css、js、html、图片等资源
	local	存放新引擎的编译文件
	logs	swift 日志（即 logdb）
	mount	存放 setenv.sh 文件，用于设置工程的环境变量和启动参数
	plugins	插件相关文件
	tomcat_logs	Tomcat 通用日志
temp_attach：读写缓存存储路径		
文件服务器	assets	vcs：FineReport 模板备份文件
		其他文件：通用的共享持久化目录
	backup	工程历史备份文件
	reportlets	FineReport 模板存放目录
	resources	存放工程相关的资源配置文件
	schedule	定时调度生成的文件
	treasures	云端运维生成的数据包

3. 确认镜像仓库

部署新项目中的各个组件，是需要通过镜像进行部署的。要么镜像仓库已有镜像，要么支持从云端拉取镜像。

1) 管理员登录运维平台，点击「运维管理>镜像管理」，查看是否存在相关镜像。

- 必须安装：FineReport、node\_exporter、ops\_agent、nginx、nginx\_exporter、elasticsearch、elasticsearch\_exporter、filebeat
- 可选安装：mysql、mysqld\_exporter、redis、redis\_exporter、minio



2) 如镜像仓库中不存在相关镜像，请查看镜像仓库地址，并确保地址可连通外网云端仓库。

- 查看镜像仓库地址：管理员登录运维平台，点击「运维管理>镜像管理>全局配置」，即可查看镜像仓库地址。
- 确保镜像仓库所在服务器支持访问以下外网地址：

域名：registry.cn-hangzhou.aliyuncs.com，端口：443

域名：fineops-registry.cn-hangzhou.cr.aliyuncs.com，端口：443

即可在部署项目时自动从云端拉取最新项目组件镜像。





## 4. 部署新 FineReport 集群项目

本节正式进入部署 FineReport 集群项目的操作环节。请在开始部署前务必确认上文的所有内容，否则可能会部署失败！

### 4.1 新增项目

管理员登录 FineOps 运维平台，点击「运维管理>项目管理」，点击「部署新项目」，如下图所示：



### 4.2 项目设置

项目设置是配置项目的类型、需要安装的组件、占用的端口等信息。

进入「项目设置」界面，输入相关项目信息，点击「下一步」，如下图所示：

部署新项目

1 项目设置

2 节点配置

项目名称

1

应用类型

2 ☒ FineReport ☐ FineBI ☐ FineDataLink

项目类型

3 ☒ 集群 ☐ 单机

默认部署组件

☒ 运维基础套件 ☒ Nginx ☒ Elasticsearch

自定义部署组件

4 ☒ 文件服务器(MinIO) ☒ 外接数据库(MySQL) ☒ 状态服务器(Redis)

部署配置

☐ 使用默认配置 ☒ 自定义端口、组件用户名密码 [去设置](#)

访问路径

配置后应用的访问链接为{域名/IP}/{访问路径}/decision

取消

5 下一步

各配置项如下表所示：

信息	说明
自定义项目的名称	
项目名称不允许为空	
项目名称	项目名称仅支持中文、英文大小写、数字
	项目名称长度最大为 12 位
	项目名称不允许与已有项目重名
应用类型	选择「FineReport」
应用类型	选择「集群」
	即必须安装的组件，包括：
	1) 运维基础套件（包括工程节点）：  FineReport、node_exporter、ops_agent
默认部署组件	2) Nginx（负载均衡，作为项目唯一入口）：  nginx、nginx_exporter
	3) Elasticsearch（作为日志采集和存储）：  elasticsearch、elasticsearch_exporter、filebeat
自定义部署组件	即可选安装的组件，包括：

件

1) 文件服务器:

minio

如不安装该组件，用户必须自行准备好一个文件服务器，否则无法部署

支持的文件服务器类型包括：SFTP、FTP、共享外部目录（V1.10.0 及之前版本不支持共享外部目录）

如需使用其他类型文件服务器，可在项目部署成功后，登录工程切换

安装成功后，如需访问 MinIO，可参考：[MinIO 文件服务器简介](#)

2) 外接数据库:

mysql、mysqld\_exporter

如不安装该组件，用户必须自行准备好一个外接配置库，否则无法部署

支持的数据库类型包括：MySQL5、MySQL8、Oracle（V1.14.0 及之前版本不支持 MySQL8）

如需使用其他类型外接库，可在项目部署成功后，登录工程切换

3) 状态服务器:

redis、redis\_exporter

如不安装该组件，用户必须自行填写状态服务器信息，否则无法部署

支持两种：

1) 使用默认配置

2) 自定义端口、组件用户名密码，即对 2.5 节的默认端口、密码进行调整

请务必修改并记住 MySQL、Redis、minio、elasticsearch 组件的密码，该密码默认随机生成

部署配置

部署新项目

1 项目设置

2 节点配置

项目名称

fanruan1

应用类型

FineReport

FineBI

FineDataLink

应用类型

集群

单机

默认部署组件

运维基础套件

Nginx

ElasticSearch

自定义部署组件

文件服务器(Minio)

外接数据库(MySQL)

状态服务器

部署配置

使用默认配置

自定义端口、组件用户名密码

去设置

访问路径

webroot

配置后应用的访问链接为(域名/IP)/(访问路径)/decision

部署配置自定义

主机端口

8080

容器端口

8080

主机端口

12100

容器端口

12100

redis

主机端口

6379

容器端口

6379

redis\_exporter

主机端口

9121

容器端口

9121

mysql

主机端口

3306

容器端口

3306

用户名

root

密码

\*\*\*\*\*

取消

完成

取消

下一步

说明：

应用的访问链接为 {域名/IP}/{访问路径}/decision

支持用户自定义应用的访问路径，访问路径默认为 webroot

举例：

一般应用访问链接为 http://IP:端口/webroot/decision

访问

路径 修改访问路径为 demo

则部署好的应用访问链接为 http://IP:端口/demo/decision

注意事项：

1.11.0 及之后版本，访问路径支持为空，即最终应用访问链接为 http://IP:端口/decision

但此时工程对应的 root 目录访问路径为：http://IP:端口/root

4.3 节点配置-新增主应用节点

配置完需要的组件和端口后，即可进入「节点配置」。

首先需要新增主应用节点，即为添加主应用服务器 ABC 的相关信息（也就是本章操作需要反复执行多次）。

1) 点击「新增节点」，选择节点类型为「主应用」，填写相关设置项，点击「测试连接」。



具体节点设置如下图所示：

节点设置	说明
	请选择「主应用」
节点类型	主应用节点要求最低配置 8 核 16G、磁盘剩余 50GB 以上 注：帆软项目需要独占服务器，即该主应用服务器上不建议部署其他工程或内容，包括运维平台
主机	请输入节点主机 IP（内网 IP） 注：同一项目中，主机不支持重复添加多次
端口	请输入节点端口号，默认为 22
用户名	请输入有 sudo 权限的服务器用户名
验证方法	支持两种：密码、公钥
	请输入节点安装目录的服务器路径，即 2.6 节设置的目录
挂载路径	默认为~/data，~代表所使用的服务器用户的主目录 注：可在终端使用该用户账号访问服务器，输入 echo \$HOME 命令查看该用户的主目录路径
	选填项
外网 IP	如果服务器不支持内网访问，仅支持外网映射，可以填写可联通的外网 IP

2) 效果预览

若主机用户名等信息校验成功，即可添加一个主应用节点。三个节点均添加完成后，如下图所示：

注 1：若网络环境不满足 2.3 节要求，节点连接失败，则出现报错提醒，无法正确新增节点。  
注 2：若服务器配置不符合 2.4 节最低要求，则出现报错提醒，无法正确新增节点。



### 4.4 节点配置-新增组件节点

接着要新增组件节点，即为添加组件服务器 D 的相关信息。

1) 点击「新增节点」，选择节点类型为「组件」，填写相关设置项，点击「测试连接」。



具体节点设置如下图所示：

节点设置	说明
节点类型	<p>请选择「组件」</p> <p>组件节点要求最低配置 4 核 8G、磁盘剩余 50GB 以上</p> <p>注 1：如配置了多个组件节点，则组件随机部署在各个节点</p> <p>注 2：帆软项目需要独占服务器，即该组件服务器上不建议部署其他工程或内容，包括运维平台</p>
主机	<p>请输入节点主机 IP（内网 IP）</p> <p>注：同一项目中，主机不支持重复添加多次</p>
端口	<p>请输入节点端口号，默认为 22</p>
用户名	<p>请输入有 sudo 权限的服务器用户名</p>
验证方法	<p>支持两种：密码、公钥</p>
挂载路径	<p>请输入节点安装目录的服务器路径，即 2.6 节设置的目录</p> <p>默认为~/data，~代表所使用的服务器用户的主目录</p> <p>注：可在终端使用该用户账号访问服务器，输入 echo \$HOME 命令查看该用户的主目录路径</p>
外网 IP	<p>选填项</p>

如果服务器不支持内网访问，仅支持外网映射，可以填写可联通的外网 IP

2) 效果预览

若主机用户名等信息校验成功，即可添加一个组件节点。如下图所示：

注 1：若网络环境不满足 2.3 节要求，节点连接失败，则出现报错提醒，无法正确新增节点。

注 2：若服务器配置不符合 2.4 节最低要求，则出现报错提醒，无法正确新增节点。



4.5 开始部署

节点配置完成后，点击「开始部署」，进入部署过程。



弹窗中会显示镜像拉取进度，以及各个组件的部署情况，若部署失败则显示失败原因。

注 1：部署过程中，用户可点击「隐藏至后台进行」，正常进行运维平台其他操作。

注 2：V1.15.0 及之后版本，支持在部署过程中点击「中止部署」按钮，停止部署并回退所有已有部署操作。

新项目部署

✓ 拉取镜像

2 执行部署

正在部署，请稍后。

需部署组件数量：13，已完成数量：10

redis

部署成功

-

redis\_exporter

部署成功

-

mysql

部署成功

-

mysqld\_exporter

部署成功

-

minio

部署成功

-

elasticsearch

部署成功

-

elasticsearch\_exporter

部署成功

-

...

部署成功

-

隐藏至后台进行

通过运维平台成功部署的新项目，自动接入运维平台的项目管理列表中，可进行相关运维操作。如下表所示：

其中的访问地址，即为该 FineReport 集群工程的访问地址，在浏览器新窗口访问该地址，即可进入工程初始化配置界面。

运维平台

部署新项目

接入已有项目

超时检测

项目管理

项目总览

文件管理

备份管理

镜像管理

名称	类型	存活节点数	部署方式	状态	访问地址	操作
fanruan1	集群	3	运维平...	可用	http://192.168.1.100:8080/webroot/decision	🔍 ⚙️ 🗑️ 🔄
fanruan2	集群	1	运维平...	可用	http://192.168.1.101:8080/webroot/decision	🔍 ⚙️ 🗑️ 🔄

<

<

1

/1

>

>



## 4.1.3 部署新 FineBI 单机项目

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.5.5	-
V1.5.6	部署过程中，不再安装冗余组件 promtail、rocketmq_namesrv、rocketmq_broker
V1.6.0	1) 单机部署时，若选择安装 elasticsearch，同步安装 filebeat 组件 2) 部署项目时，工程外挂目录新增 help 文件夹
V1.9.0	部署交互界面优化
V1.10.0	部署进度界面优化，细分展示镜像拉取和组件部署进度
V1.11.0	1) 项目名称限制放宽，新增支持英文大写、中文 2) 项目配置中，访问路径支持为空
V1.14.0	部署的配置库组件，由 MySQL5 调整为 MySQL8
V1.15.0	1) 节点配置时，支持填写外网 IP，以适配不支持内网访问、仅可外网映射的服务器 2) 支持在部署过程中「中止部署」，自动停止本次部署并回退已有操作 3) 部署的 FineBI 组件，默认 Xmx 限制提升，降低 oomkill 风险  优化前：默认 Xmx=机器内存减掉 8G 后的内存大小  优化后：默认 Xmx=机器内存减掉 10G 后的内存大小

#### 1.2 功能简介

运维平台支持部署并直接接入全新 FineReport、FineBI、FineDataLink 应用。

本文将介绍，如何通过 [运维平台](#)，容器化部署全新的 **FineBI 单机**项目。

注 1：本文面向服务器资源不足或端口限制较高的用户，操作步骤较为详细，篇幅较长。

如能准备充足资源的全新服务器，可参考快速指南进行部署：[部署新帆软项目](#)

注 2：请在容器化部署前，先部署好 FineOps 运维平台。安装步骤请参见：[部署运维平台](#)。

### 2. 准备服务器

#### 2.1 确认需要安装的组件

通过运维平台容器化部署的 FineBI 单机项目，不仅仅包括 FineBI 工程，还包括配套的外接配置库、运维组件等等。

请先确认好需要安装的组件。本文按照安装全部组件来进行示例讲解。

服务 器	安装的组 件	是否 安装	说明
	FineBI	必须	作用：FineBI 的工程节点组件，其中不带任何的 demo 数据
		安装	说明：必须安装
	ops_agen t	必须	作用：为服务器物理节点配套安装，用于管理组件，容器化升
		安装	级也是由该组件完成 说明：FineBI 项目涉及的每个服务器都会安装一个 ops_agent
主应 用服 务器	node_exp orter	必须	作用：服务器物理节点的 exporter，收集物理服务器的各种指
		安装	标信息 说明：FineBI 项目涉及的每个服务器都会安装一个 node_expo rter
	filebeat	可 选 安 装	作用：日志采集 说明：与 elasticSearch 组件配套安装
			如不安装 elasticsearch 组件，则不安装 filebeat 组件 如安装 elasticsearch 组件，则同步在 FineBI 工程节点所在服务 器安装 filebeat 组件 作用：作为工程的外接配置库 说明：可选安装
组件 服务 器	mysql	可 选	如果安装该组件，会安装一个 MySQL8 数据库，作为工程外
		安装	接配置库 如不安装该组件，用户必须自行准备好一个外接配置库，否则 无法部署 支持的数据库类型包括：MySQL5、MySQL8、Oracle（V1.14. 0 及之前版本不支持 MySQL8）

如需使用其他类型外接库，可在项目部署成功后，登录工程切换		
换		
mysqld_exporter	可选	作用：MySQL 服务的 exporter，收集 MySQL 的各种指标信息
	安装	说明：与 MySQL 组件配套安装
作用：日志存储		
说明：可选安装		
elasticsearch	可选	如果安装该组件，则取代 FineBI 中原有的 swift，作为日志存
	安装	储
如不安装该组件，将启用 FineBI 中原有的 swift (logdb) 进行		
日志存储		
elasticsearch_exporter	可选安装	作用：elasticSearch 的 exporter，收集 elasticSearch 的指标
		说明：与 elasticSearch 组件配套安装
ops_agent	必须	作用：为服务器物理节点配套安装，用于管理组件，容器化升
	安装	级也是由该组件完成
说明：FineBI 项目涉及的每个服务器都会安装一个 ops_agent		
node_exporter	必须	作用：服务器物理节点的 exporter，收集物理服务器的各种指
	安装	标信息
说明：		
如果准备了组件服务器，则自动安装一个 node_exporter 组件		
如果不安装 mysql 和 elasticsearch 套件，则无需准备组件服务		
器，则不会安装该组件		

2.2 服务器数量要求

本文按照 2 个服务器进行下文的部署操作，分为主应用服务器 A、组件服务器 B。

请准备好服务器的内网 IP、用户名、密码。

注：FineOps 运维平台，和 FineBI 项目，不要部署在同一服务器下。

如果 FineOps 和项目处于同一环境下，当项目服务器发生异常时，FineOps 可能也出现故障，从而无法起到运维的作用。

类型	要求
	建议准备 2 个服务器，这两个服务器中不再安装其他工程/产品
推荐配置	<div><ul style="list-style-type: none"><li>工程独占一个服务器：即主应用服务器，包括 FineBI、ops_agent</li><li>其他组件独占一个服务器：即组件服务器，如果不安装 mysql 和 elastic search 套件，则无需准备该服务器</li></ul></div>
	至少准备 1 个服务器，这个服务器中不再安装其他工程/产品
最低配置	<div><ul style="list-style-type: none"><li>整个 FineBI 项目独占一个服务器（只有 1 个服务器的情况下，所有组件均安装在该服务器中）</li></ul></div>

2.3 服务器网络要求

请确保 FineBI 项目涉及的主应用服务器 A、组件服务器 B，以及 FineOps 运维平台所在的服务器 C：

这些服务器内网互相联通，如不联通则无法成功部署。

注：如自行准备了外接配置库，外接配置库所在的服务器 D，也需要和 ABC 服务器内网互通。

2.4 服务器系统配置

请确保 FineBI 项目涉及的主应用服务器 A、组件服务器 B，满足此基础配置。

服务器	配置	要求
		CentOS 7.3~7.9 版本
	系统版本	Ubuntu 18.04.4 及以上版本
		redhat 7.6 及以上版本
主应用服务器		3.10 及以上
	系统内核	最低 8 核
	CPU 架构	仅支持 X86_64
	剩余空闲物理内存	32G 以上
	剩余空闲磁盘容量	100GB 以上
组件服务器		CentOS 7.3~7.9 版本
	系统版本	Ubuntu 18.04.4 及以上版本

	redhat 7.6 及以上版本
	3.10 及以上
系统内核	
	最低 4 核
CPU 架构	仅支持 X86_64
剩余空闲物理内存	8G 以上
剩余空闲磁盘容量	50GB 以上

注：服务器之所以要满足上表所说的内存和磁盘，是由于 FineBI 项目中的各个组件有最低安装配置要求。

点击展开更多

服务器	组件	CPU	内存	xmx	磁盘
			默认内存为：		
			机器内存减掉 1		
	FineBI	16Core	-	0G 后的内存大小	100G
主应用服务器					
				最大不会超过 3	
				2G	
	ops_agent	1Core	-	1G	-
	node_exporter	0.1Core	0.1G	-	-
	filebeat	0.1Core	0.1G	-	-
	mysql	2Core	-	-	-
	mysqld_exporter	0.1Core	0.1G	-	-
	elasticsearch	1Core	-	2G	50G
组件服务器	elasticsearch_exporter	0.1Core	0.1G	-	-
	ops_agent	1Core	-	1G	-
	node_exporter	0.1Core	0.1G	-	-

2.5 服务器端口要求

请确保 FineBI 项目涉及的主应用服务器 A、组件服务器 B 中，待部署应用和组件的默认端口可用。

- 请确认默认的映射端口是否已被使用，如已被使用，请安排好其他空闲端口。
- FineBI 的映射端口将作为项目访问端口，因此请确认准备使用的 FineBI 映射端口是否开放了防火墙、安全组等，确保可连接。

注 1：端口占用和防火墙开放操作方法可参考：[端口占用检查与防火墙配置](#)。

注 2：端口 portMapping 为 {映射端口:容器内端口}，其中映射端口可改，容器内端口不支持修改。

服务器	组件	默认端口
主应用服务器	FineBI	端口：{8080: 8080, 12100: 12100}
	ops_agent	端口：{9070:9070}
	node_exporter	端口：{9100:9100}
	filebeat	不占用端口
		端口：{3306:3306}
组件服务器	mysql	用户名：root（不支持修改）  密码：随机生成，届时请务必修改
	mysqld_exporter	端口：{9104:9104}
		端口：{9200:9200}
	elasticsearch	用户名：elastic（不支持修改）  密码：elasticadmin（支持修改）
	elasticsearch_exporter	端口：{9114:9114}
	ops_agent	端口：{9070:9070}
	node_exporter	端口：{9100:9100}

2.6 设置项目挂载目录

容器化部署的 FineBI 项目，重要文件均外挂。请提前在主应用服务器 A 和组件服务器 B 的空闲磁盘下，新建好挂载目录。

1) 查看服务器磁盘空间

命令：df -h，找出可用空间较大的挂载点

例如本示例中，服务器最合适的挂载目录为/home 目录下

```
Last login: Tue Jul 25 10:16:49 2023 from 192.168.5.2
[root@192 ~]# df -h
文件系统      容量  已用  可用 已用% 挂载点
/dev/mapper/centos-root  50G   14G   37G   28% /
devtmpfs      9.8G    0   9.8G    0% /dev
tmpfs         9.8G    0   9.8G    0% /dev/shm
tmpfs         9.8G  962M   8.8G   10% /run
tmpfs         9.8G    0   9.8G    0% /sys/fs/cgroup
/dev/mapper/centos-home 142G   12G  130G    9% /home
/dev/sda1     1014M  146M   869M   15% /boot
tmpfs         2.0G    0   2.0G    0% /run/user/0
```

2) 新建文件夹

- 使用 mkdir 命令创建文件夹（本示例在/home 目录下创建一个名叫 fanruan 的文件夹）：mkdir /home/fanruan
- 使用 chmod 命令设置文件夹的权限（本示例直接赋 777 权限）：  
chmod 777 /home/fanruan

```
[root@192 ~]#  
[root@192 ~]# mkdir /home/fanruan  
[root@192 ~]# chmod 777 /home/fanruan  
[root@192 ~]#  
[root@192 ~]#  
[root@192 ~]# cd /home  
[root@192 home]# ls  
fanruan finekey finekey-operation-all.tar.gz fr  
[root@192 home]#
```

注：FineBI 单机项目部署成功后，项目挂载目录中的 bi 文件夹，存放着工程外挂内容，其中的文件夹和存放内容如下表所示。

点击展开更多

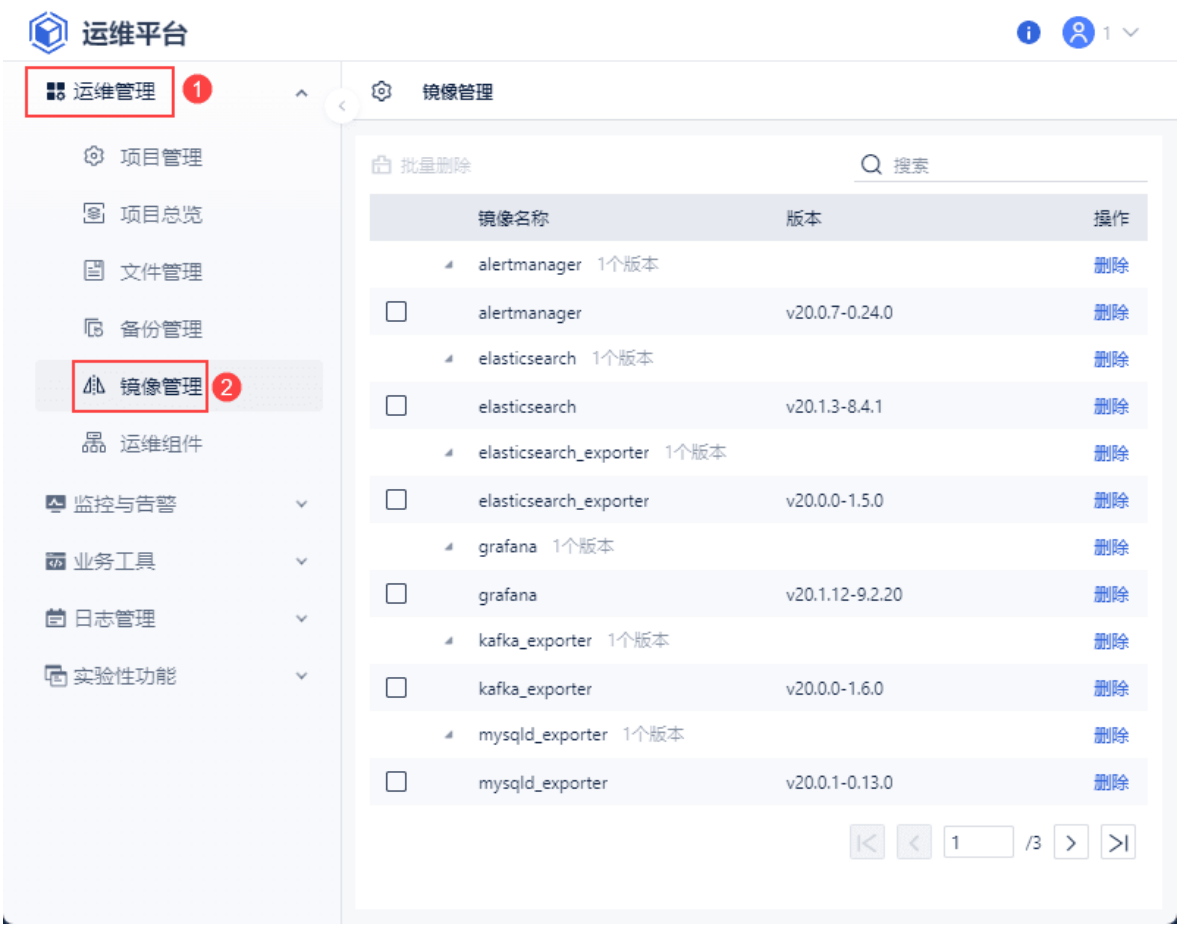
外挂文件夹	说明
	temp_attach：存放着工程 Excel 原始文件信息
assets	vcs：FineReport 模板备份文件  其他文件：通用的共享持久化目录
backup	工程历史备份文件
bi-data	FineBI 抽取数据
classes	工程调用的默认和自定义 class 文件
config	平台调用的配置库（finedb）相关文件
customlib	工程定制/外部引入的 JAR 包
dpworks	FineDataLink 任务相关的配置文件（如不存在该文件夹，说明工程没有使用 FDL 相关功能）
help	工程需要调用的一些 css、js、html、图片等资源
local	存放新引擎的编译文件
logs	swift 日志（即 logdb）
mount	存放 setenv.sh 文件，用于设置工程的环境变量和启动参数
plugins	插件相关文件
reportlets	FineReport 模板存放目录
resources	存放工程相关的资源配置文件
schedule	定时调度生成的文件
tomcat_logs	Tomcat 通用日志
treasures	云端运维生成的数据包

### 3. 确认镜像仓库

部署新项目中的各个组件，是需要通过镜像进行部署的。要么镜像仓库已有镜像，要么支持从云端拉取镜像。

1) 管理员登录运维平台，点击「运维管理>镜像管理」，查看是否存在相关镜像。

- 必须安装：FineBI、node\_exporter、ops\_agent
- 可选安装：mysql、mysqld\_exporter、elasticsearch、elasticsearch\_exporter、filebeat



2) 如镜像仓库中不存在相关镜像，请查看镜像仓库地址，并确保地址可连通外网云端仓库。

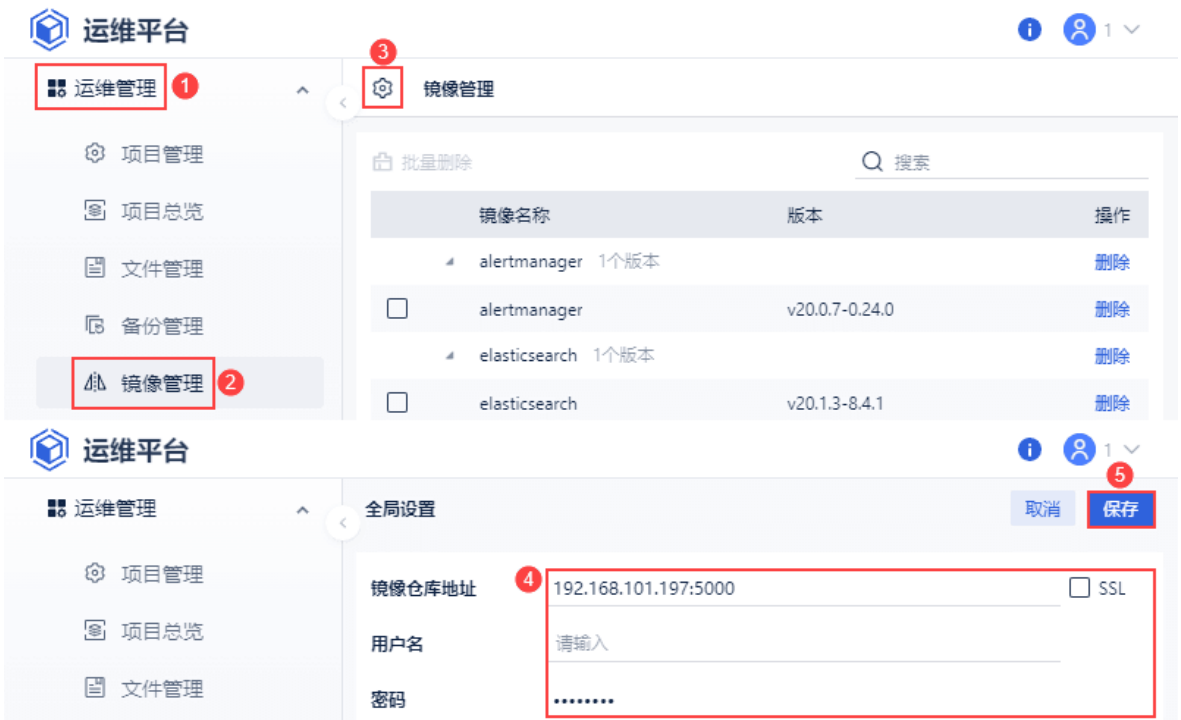
- 查看镜像仓库地址：管理员登录运维平台，点击「运维管理>镜像管理>全局配置」，即可查看镜像仓库地址。
- 确保镜像仓库所在服务器支持访问以下外网地址：

域名：registry.cn-hangzhou.aliyuncs.com，端口：443

域名：fineops-registry.cn-hangzhou.cr.aliyuncs.com，端口：443

即可在部署项目时自动从云端拉取最新项目组件镜像。





#### 4. 部署新 FineBI 单机项目

本节正式进入部署 FineBI 单机项目的操作环节。请在开始部署前务必确认上文的所有内容，否则可能会部署失败！

##### 4.1 新增项目

管理员登录 FineOps 运维平台，点击「运维管理>项目管理」，点击「部署新项目」，如下图所示：



##### 4.2 项目设置

项目设置是配置项目的类型、需要安装的组件、占用的端口等信息。

进入「项目设置」界面，输入相关项目信息，点击「下一步」，如下图所示：

部署新项目

1 项目设置

2 节点配置

项目名称

1

应用类型

☐ FineReport

2 ☒ FineBI

☐ FineDataLink

项目类型

☐ 集群

3 ☒ 单机

默认部署组件

☒ 运维基础套件

自定义部署组件

4 

☒ 外接数据库(MySQL) ☒ ElasticSearch

部署配置

☐ 使用默认配置

☒ 自定义端口、组件用户名密码 [去设置](#)

访问路径

配置后应用的访问链接为(域名/IP)/(访问路径)/decision

取消

5 下一步

各配置项如下表所示：

信息	说明
自定义项目的名称	
项目名称不允许为空	
项目名称	项目名称仅支持中文、英文大小写、数字
	项目名称长度最大为 12 位
	项目名称不允许与已有项目重名
应用类型	选择「FineBI」
应用类型	选择「单机」
默认部署组件	即必须安装的组件，包括： FineBI、node_exporter、ops_agent
	即可选安装的组件，包括：
	1) 外接数据库
自定义部署组件	mysql、mysqld_exporter
	如不安装该组件，用户必须自行准备好一个外接配置库，否则无法部署
	支持的数据库类型包括：MySQL5、MySQL8、Oracle（V1.14.0 及之前版本不支持 MySQL8）

如需使用其他类型外接库，可在项目部署成功后，登录工程切换

2) ElasticSearch（可选择，默认不安装）：

elasticsearch、elasticsearch\_exporter、filebeat

如不安装该组件，将启用 FineBI 中原有的 swift（logdb），用作日志存储

支持两种：

1) 使用默认配置

2) 自定义端口、组件用户名密码，即对 2.5 节的默认端口、密码进行调整

请务必修改并记住 MySQL、elasticsearch 组件的密码，该密码默认随机生成

部署新项目

1 项目设置

2 节点配置

部署配置

项目名称

fanruan1

应用类型

☐ FineReport

☒ FineBI

☐ FineDataLink

应用类型

☐ 集群

☒ 单机

默认部署组件

☒ 运维基础套件

自定义部署组件

☒ 外接数据库(MySQL)

☒ ElasticSearch

部署配置

☐ 使用默认配置

☒ 自定义端口、组件用户名密码

访问路径

webroot

配置后应用的访问链接为{域名/IP}/{访问路径}/decision

部署配置自定义

FineBI

主机端口

8080

容器端口

8080

主机端口

12100

容器端口

12100

mysql

主机端口

3306

容器端口

3306

用户名

root

密码

.....

mysql\_exporter

主机端口

9104

容器端口

9104

elasticsearch

主机端口

9200

容器端口

9200

取消

完成

去设置

取消

下一步

说明：

应用的访问链接为{域名/IP}/{访问路径}/decision

支持用户自定义应用的访问路径，访问路径默认为 webroot

举例：

一般应用访问链接为 http://IP:端口/webroot/decision

访问  
路径

修改访问路径为 demo

则部署好的应用访问链接为 http://IP:端口/demo/decision

注意事项：

1.11.0 及之后版本，访问路径支持为空，即最终应用访问链接为 http://IP:端口/decision

但此时工程对应的 root 目录访问路径为：http://IP:端口/root

### 4.3 节点配置-新增主应用节点

配置完需要的组件和端口后，即可进入「节点配置」。

首先需要新增主应用节点，即为添加主应用服务器 A 的相关信息。

- 1) 点击「新增节点」，选择节点类型为「主应用」，填写相关设置项，点击「测试连接」。



具体节点设置如下图所示：

节点设置	说明
节点类型	<p>请选择「主应用」</p> <p>主应用节点要求最低配置 8 核 32G、磁盘剩余 100GB 以上</p> <p>注 1：如未准备组件服务器，那么所有组件都将部署在该主应用节点下</p> <p>注 2：帆软项目需要独占服务器，即该主应用服务器上不建议部署其他工程或内容，包括运维平台</p>
主机	<p>请输入节点主机 IP（内网 IP）</p> <p>注：同一项目中，主机不支持重复添加多次</p>
端口	请输入节点端口号，默认为 22
用户名	请输入有 sudo 权限的服务器用户名
验证方法	支持两种：密码、公钥

请输入节点安装目录的服务器路径，即 2.6 节设置的目录

挂载路径 默认为~/data，~代表所使用的服务器用户的主目录  
注：可在终端使用该用户账号访问服务器，输入 echo \$HOME 命令查看该用户的主目录路径

选填项  
外网 IP 如果服务器不支持内网访问，仅支持外网映射，可以填写可联通的外网 IP

2) 效果预览

若主机用户名等信息校验成功，即可添加一个主应用节点。如下图所示：

注 1：若网络环境不满足 2.3 节要求，节点连接失败，则出现报错提醒，无法正确新增节点。

注 2：若服务器配置不符合 2.4 节最低要求，则出现报错提醒，无法正确新增节点。



4.4 节点配置-新增组件节点

接着要新增组件节点，即为添加组件服务器 B 的相关信息。如未准备组件服务器，无需执行本章操作，所有组件都将部署在上一节主应用节点下。

1) 点击「新增节点」，选择节点类型为「组件」，填写相关设置项，点击「测试连接」。



具体节点设置如下图所示：

节点设置	说明
节点类型	<p>请选择「组件」</p> <p>组件节点要求最低配置 4 核 8G、磁盘剩余 50GB 以上</p> <p>注 1：如配置了多个组件节点，则组件随机部署在各个节点</p> <p>注 2：帆软项目需要独占服务器，即该组件服务器上不建议部署其他工程或内容，包括运维平台</p>
主机	<p>请输入节点主机 IP（内网 IP）</p> <p>注：同一项目中，主机不支持重复添加多次</p>
端口	请输入节点端口号，默认为 22
用户名	请输入有 sudo 权限的服务器用户名
验证方法	<p>支持两种：密码、公钥</p> <p>请输入节点安装目录的服务器路径，即 2.6 节设置的目录</p>
挂载路径	<p>默认为~/data，~代表所使用的服务器用户的主目录</p> <p>注：可在终端使用该用户账号访问服务器，输入 echo \$HOME 命令查看该用户的主目录路径</p>
外网 IP	选填项

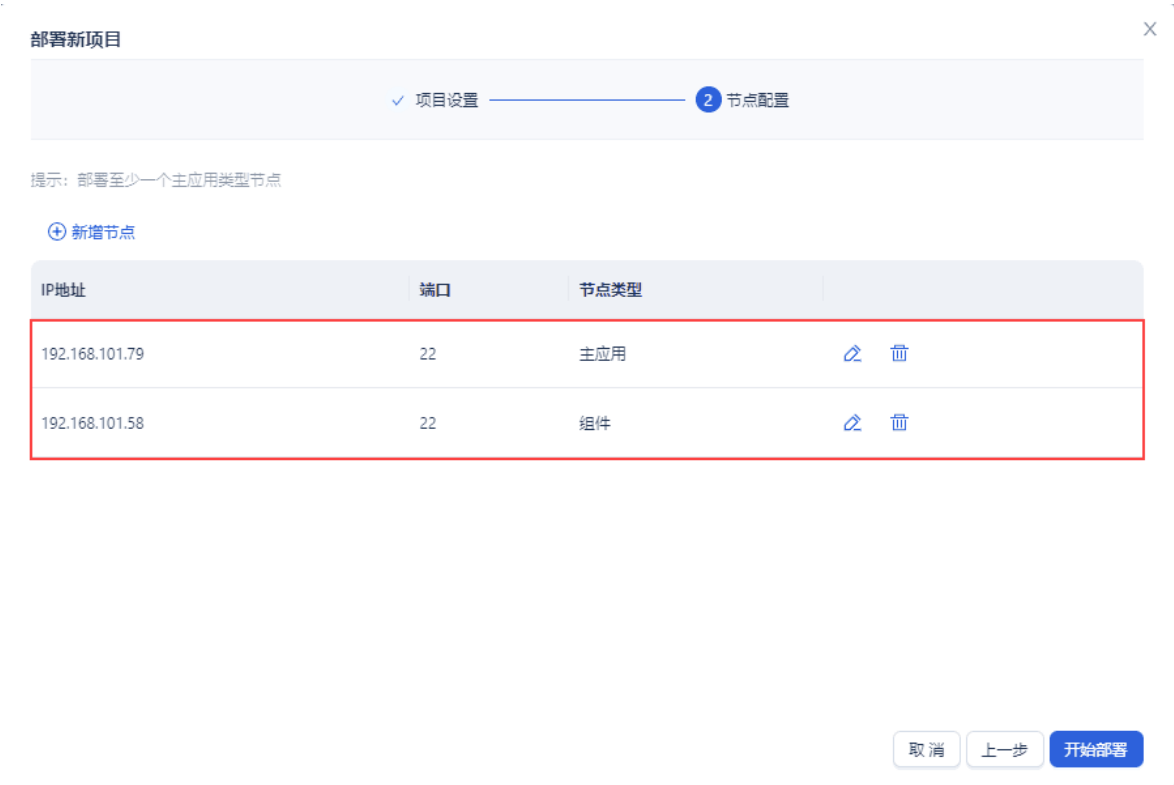
如果服务器不支持内网访问，仅支持外网映射，可以填写可联通的外网 IP

2) 效果预览

若主机用户名等信息校验成功，即可添加一个组件节点。如下图所示：

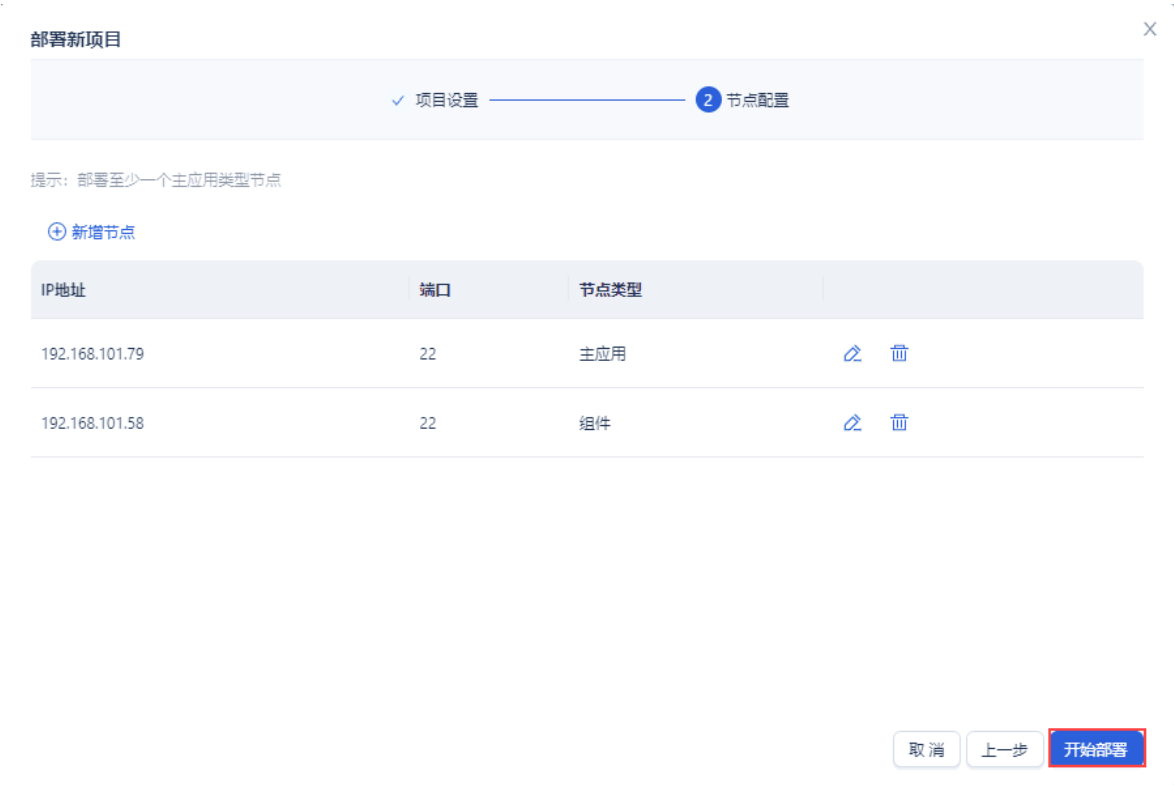
注 1：若网络环境不满足 2.3 节要求，节点连接失败，则出现报错提醒，无法正确新增节点。

注 2：若服务器配置不符合 2.4 节最低要求，则出现报错提醒，无法正确新增节点。



4.5 开始部署

节点配置完成后，点击「开始部署」，进入部署过程。



弹窗中会显示各个组件的部署情况，若部署失败则显示失败原因。

- 注 1：部署过程中，用户可点击「隐藏至后台进行」，正常进行运维平台其他操作。
- 注 2：V1.15.0 及之后版本，支持在部署过程中点击「中止部署」按钮，停止部署并回退所有已有部署操作。

新项目部署

正在部署，请稍后...

需部署8个组件，已完成0个

mysql

部署中...

mysqld\_exporter

部署中...

elasticsearch

部署中...

elasticsearch\_expor...

部署中...

node\_exporter

部署中...

fr

部署中...

filebeat

部署中...

隐藏至后台进行

通过运维平台成功部署的新项目，自动接入运维平台的项目管理列表中，可进行相关运维操作。如下表所示：

其中的访问地址，即为该 FineBI 单机工程的访问地址，在浏览器新窗口访问该地址，即可进入工程初始化配置界面。

运维平台

运维管理

项目管理

项目总览

文件管理

备份管理

镜像管理

部署新项目

接入已有项目

超时检测

名称	类型	存活节点数	部署方式	状态	访问地址	操作
fanrua...	集群	2	非运...	可用	http://10.10.10.10:30080/webroot/decision	🔍 ⚙️ 🗑️ 📄
fanrua...	单机	1	非运维...	可用	http://10.10.10.10:30080/webroot/decision	🔍 ⚙️ 🗑️ 📄
fanru...	单机	1	运维...	可用	http://10.10.10.10:30080/broot/decision	🔍 ⚙️ 🗑️ 📄 🔄

<<

<

1

/1

>

>>



## 4.1.4 部署新 FineBI 集群项目

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.5.5	-
V1.5.6	部署过程中，不再安装冗余组件 promtail、rocketmq_namesrv、rocketmq_broker
V1.6.0	1) 为了确保集群日志存储的稳定性：  容器化部署集群项目时，elasticsearch 套件不再可选安装，而是必须安装 安装 elasticsearch 时，同步在每一个工程节点安装 filebeat 组件
V1.9.0	2) 部署项目时，工程外挂目录新增 help 文件夹  部署交互界面优化
V1.10.0	部署进度界面优化，细分展示镜像拉取和组件部署进度  1) 项目名称限制放宽，新增支持英文大写、中文
V1.11.0	2) 部署新集群项目时，支持使用外部共享目录作为文件服务器  3) 项目配置中，访问路径支持为空
V1.14.0	部署的配置库组件，由 MySQL5 调整为 MySQL8  1) 节点配置时，支持填写外网 IP，以适配不支持内网访问、仅可外网映射的服务器  2) 支持在部署过程中「中止部署」，自动停止本次部署并回退已有操作
V1.15.0	3) 部署的 FineBI 组件，默认 Xmx 限制提升，降低 oomkill 风险  优化前：默认 Xmx=机器内存减掉 8G 后的内存大小  优化后：默认 Xmx=机器内存减掉 10G 后的内存大小

#### 1.2 功能简介

运维平台支持部署并直接接入全新 FineReport、FineBI、FineDataLink 应用。

本文将介绍，如何通过 [运维平台](#)，容器化部署全新的 **FineBI 抽取集群**项目（本文示例三节点集群）。

注 1：本文面向服务器资源不足或端口限制较高的用户，操作步骤较为详细，篇幅较长。

如能准备充足资源的全新服务器，可参考快速指南进行部署：[部署新帆软项目](#)

注 2：请在容器化部署前，先部署好 FineOps 运维平台。安装步骤请参见：[部署运维平台](#)。

## 2. 准备服务器

### 2.1 确认需要安装的组件

通过运维平台容器化部署的 FineBI 集群项目，不仅仅包括 FineBI 工程，还包括配套的外接配置库、集群组件、运维组件等等。

请先确认好需要安装的组件。本文按照安装全部组件来进行示例讲解。

服务器	组件	是否安装	组件说明
主应用服务器	FineBI	必须安装	作用：FineBI 的工程节点组件，其中不带任何的 demo 数据 说明：必须安装
	ops_agent	必须安装	作用：为服务器物理节点配套安装，用于管理组件，容器化升级也是由该组件完成 说明：FineBI 项目涉及的每个服务器都会安装一个 ops_agent
	node_exporter	必须安装	作用：服务器物理节点的 exporter，收集物理服务器的各种指标信息 说明：每一个主应用服务器，都会自动安装一个 node_exporter 组件
	filebeat	必须安装	作用：日志采集 说明：与 elasticSearch 组件配套安装，在每个 FineBI 工程节点所在服务器安装 filebeat 组件
			作用：日志存储
	elasticsearch	必须安装	说明：替换 FineBI 中原有的 swift 作为日志存储 对项目中 FineBI、外接数据库、业务库和其他集群组件的请求进行采集

集群必须安装该组件，且请勿暂停该组件，否则会导致项目操作日志丢失

elasticsearch_exporter	必须安装	<p>作用：elasticSearch 的 exporter，收集 elasticSearch 的指标</p> <p>说明：与 elasticSearch 组件配套安装</p>
nginx	必须安装	<p>作用：提供集群的负载转发服务</p> <p>说明：集群其他组件服务的入口、使用，都从这个入口访问</p>
nginx_exporter	必须安装	<p>作用：nginx 服务的 exporter，收集 nginx 的指标信息</p> <p>说明：与 nginx 组件配套安装</p>
		<p>作用：作为集群的外接配置库</p> <p>说明：可选安装</p> <p>如果安装该组件，会安装一个 MySQL8 数据库，作为集群的外接配置库</p>
mysql	可选安装	<p>如不安装该组件，用户必须自行准备好一个外接配置库，否则无法部署</p> <p>支持的数据库类型包括：MySQL5、MySQL8、Oracle（V 1.14.0 及之前版本不支持 MySQL8）</p> <p>如需使用其他类型外接库，可在项目部署成功后，登录工程切换</p>
mysqld_exporter	可选安装	<p>作用：MySQL 服务的 exporter，收集 MySQL 的各种指标信息</p> <p>说明：与 MySQL 组件配套安装</p>
		<p>作用：作为集群的状态服务器</p> <p>说明：可选安装</p>
redis	可选安装	<p>如果安装该组件，会安装一个单机 redis，作为集群的状态服务器</p> <p>如不安装该组件，用户必须自行准备好一个 redis 组件（单机集群均可），否则无法部署</p>

redis_exporter	可选安装	作用：redis 服务的 exporter，收集 redis 的各种指标信息 说明：与 redis 组件配套安装 作用：作为集群的文件服务器 说明：可选安装 如果安装该组件，会安装一个 minio，它是一款高性能、
minio	可选安装	分布式的对象存储系统，作为集群的文件服务器 如不安装该组件，用户必须自行准备好一个文件服务器，否则无法部署 支持的文件服务器类型包括：SFTP、FTP、共享外部目录（V1.10.0 及之前版本不支持共享外部目录） 如需使用其他类型文件服务器，可在项目部署成功后，登录工程切换
ops_agent	必须安装	作用：为服务器物理节点配套安装，用于管理组件，容器化升级也是由该组件完成 说明：FineBI 项目涉及的每个服务器都会安装一个 ops_agent 作用：服务器物理节点的 exporter，收集物理服务器的各种指标信息
node_exporter	必须安装	说明：每一个组件服务器，都会自动安装一个 node_exporter 组件

2.2 服务器数量要求

本文按照 4 个服务器进行下文的部署操作, 分为主应用服务器 ABC、组件服务器 D。

请准备好服务器的内网 IP、用户名、密码。

注：FineOps 运维平台，和 FineBI 项目，不要部署在同一服务器下。

如果 FineOps 和项目处于同一环境下, 当项目服务器发生异常时, FineOps 可能也出现故障, 从而无法起到运维的作用。

类型	要求
推荐配置	建议按照「节点数量+1」准备服务器数量，这些服务器中不再安装其他工程/产品 1) 每个工程节点独占一个服务器： <ul style="list-style-type: none"><li>例如本文示例的是 3 节点集群，则需要 3 个主应用服务器。</li></ul>

- 每个服务器中安装一套 FineBI、ops\_agent、node\_exporter、filebeat

2) 其他组件独占一个服务器：即组件服务器

- 该服务器中安装外接配置库 (mysql)、集群组件 (nginx、redis、minio)、es 组件 (elasticsearch) 以及运维组件 (expoder 组件)
- 如果服务器资源充足，可以多准备一些组件服务器，在部署时会随机分配集群组件到各个服务器中，确保项目高可用

建议按照「节点数量+5」准备服务器数量，这些服务器中不再安装其他工程/产品

高可用配置 1) 每个工程节点独占一个服务器，作为主应用服务器

2) redis、nginx、mysql、minio、es 组件各自独占一个服务器，作为组件服务器

2.3 服务器网络要求

请确保 FineBI 项目涉及的主应用服务器 ABC、组件服务器 D，以及 FineOps 运维平台所在的服务器 E：

这些服务器内网互相联通，如不联通则无法成功部署。

注：如自行准备了外接配置库、状态服务器、文件服务器。他们所在的服务器 F、G、H，也需要和 ABCDE 服务器内网互通。

2.4 服务器系统配置

请确保 FineBI 项目涉及的主应用服务器 ABC、组件服务器 B，满足此基础配置。

注意，下表的配置要求是针对每个服务器，而不是所有服务器的配置总和。

服务器	配置	要求
主应用服务器		CentOS 7.3~7.9 版本
	系统版本	Ubuntu 18.04.4 及以上版本
		redhat 7.6 及以上版本
		3.10 及以上
	系统内核	最低 8 核
	CPU 架构	仅支持 X86_64
	剩余空闲物理内存	32G 以上

组件服务器	剩余空闲磁盘容量	100GB 以上
	系统版本	CentOS 7.3~7.9 版本
	系统版本	Ubuntu 18.04.4 及以上版本
	系统版本	redhat 7.6 及以上版本
	系统内核	3.10 及以上
	CPU 架构	仅支持 X86_64
组件服务器	最低 4 核	
	最低 8G	
	剩余空闲物理内存	若仅准备了一个 1 个组件服务器，建议 16G 以上
组件服务器	剩余空闲磁盘容量	50GB 以上

注：服务器之所以要满足上表所说的内存和磁盘，是由于 FineBI 项目中的各个组件有最低安装配置要求。

点击展开更多

服务器	组件	CPU	内存	xmx	磁盘
主应用服务器	默认内存为：				
	机器内存减掉 10G 后的内存大小				
	FineBI	16Core	-		100G
	最大不会超过 32G				
	ops_agent	1Core	-	1G	-
	node_exporter	0.1Core	0.1G	-	-
	filebeat	0.1Core	0.1G	-	-
	elasticsearch	1Core	-	2G	50G
	elasticsearch_exporter	0.1Core	0.1G	-	-
	nginx	1Core	-	-	-
	nginx_exporter	0.1Core	0.1G	0.1G	-
	mysql	2Core	-	-	50G
组件服务器	mysqld_exporter	0.1Core	0.1G	-	-
	redis	1Core	-	4G	-
	redis_exporter	0.1Core	0.1G	-	-
	minio	1Core	-	-	100G
	ops_agent	1Core	-	1G	-
	node_exporter	0.1Core	0.1G	-	-

2.5 服务器端口要求

请确保 FineBI 项目涉及的主应用服务器 ABC、组件服务器 D 中，待部署应用和组件的默认端口可用。

- 请确认默认的映射端口是否已被使用，如已被使用，请安排好其他空闲端口。
- nginx 的映射端口将作为项目访问端口，因此请确认准备使用的 nginx 映射端口是否开放了防火墙、安全组等，确保可连接。

注 1：端口占用和防火墙开放操作方法可参考：[端口占用检查与防火墙配置](#)。

注 2：端口 portMapping 为 {映射端口:容器内端口}，其中映射端口可改，容器内端口不支持修改。

注 3：如果准备了多个组件服务器，组件会随机分布部署在各个服务器下。因此请确保每一个组件服务器都满足各个组件的端口要求。

服务器	组件	默认端口
主应用服务器	FineBI	端口：{8080: 8080, 7800: 7800, 7830: 7830, 7840: 7840, 7850: 7850, 7870: 7870}
	ops_agent	端口：{9070:9070}
	node_exporter	端口：{9100:9100}
	filebeat	不占用端口
		端口：{9200:9200}
组件服务器	elasticsearch	用户名：elastic（不支持修改）
		密码：elasticadmin（支持修改）
	elasticsearch_exporter	端口：{9114:9114}
		端口：{80: 8080}
	nginx	非 root 用户建议修改占用端口，因为非 root 用户默认无权限使用 1024 以下端口
	nginx_exporter	端口：{9113: 9113}
		端口：{3306:3306}
	mysql	用户名：root（不支持修改）
		密码：随机生成，届时请务必修改
	mysqld_exporter	端口：{9104:9104}
		端口：{6379:6379}
	redis	用户名：admin（不支持修改）

	密码：随机生成，届时请务必修改
redis_exporter	端口：{9121:9121}
	服务器端口：{9000:9000}
	控制台端口：{9001:9001}
minio	用户名：minioadmin（不支持修改）
	密码：minioadmin（支持修改，但密码必须大于等于 8 位， 否则使用默认密码）
ops_agent	端口：{9070:9070}
node_exporter	端口：{9100:9100}

2.6 设置项目挂载目录

容器化部署的 FineBI 项目，重要文件均外挂。请提前在主应用服务器 ABC 和组件服务器 D 的空闲磁盘下，新建好挂载目录。

1) 查看服务器磁盘空间

命令：df -h，找出可用空间较大的挂载点

例如本示例中，服务器最合适的挂载目录为/home 目录下

```
Last login: Tue Jul 25 10:16:49 2023 from 192.168.5.2
[root@192 ~]# df -h
文件系统          容量  已用  可用 已用% 挂载点
/dev/mapper/centos-root    50G   14G   37G   28% /
devtmpfs              9.8G    0 9.8G    0% /dev
tmpfs                 9.8G    0 9.8G    0% /dev/shm
tmpfs                 9.8G  962M 8.8G   10% /run
tmpfs                 9.8G    0 9.8G    0% /sys/fs/cgroup
/dev/mapper/centos-home   142G   12G 130G    9% /home
/dev/sda1              1014M  146M 869M   15% /boot
tmpfs                  2.0G    0 2.0G    0% /run/user/0
```

2) 新建文件夹

- 使用 mkdir 命令创建文件夹（本示例在/home 目录下创建一个名叫 fanruan 的文件夹）：mkdir /home/fanruan
- 使用 chmod 命令设置文件夹的权限（本示例直接赋 777 权限）：  
chmod 777 /home/fanruan

```
[root@192 ~]#
[root@192 ~]# mkdir /home/fanruan
[root@192 ~]# chmod 777 /home/fanruan
[root@192 ~]#
[root@192 ~]#
[root@192 ~]# cd /home
[root@192 home]# ls
fanruan finekey finekey-operation-all.tar.gz fr
[root@192 home]#
```

注：FineBI 集群项目部署成功后，文件服务器和项目挂载目录中的 bi 文件夹中，分别存放着工程



外挂内容，其中的文件夹和存放内容如下表所示。

点击展开更多

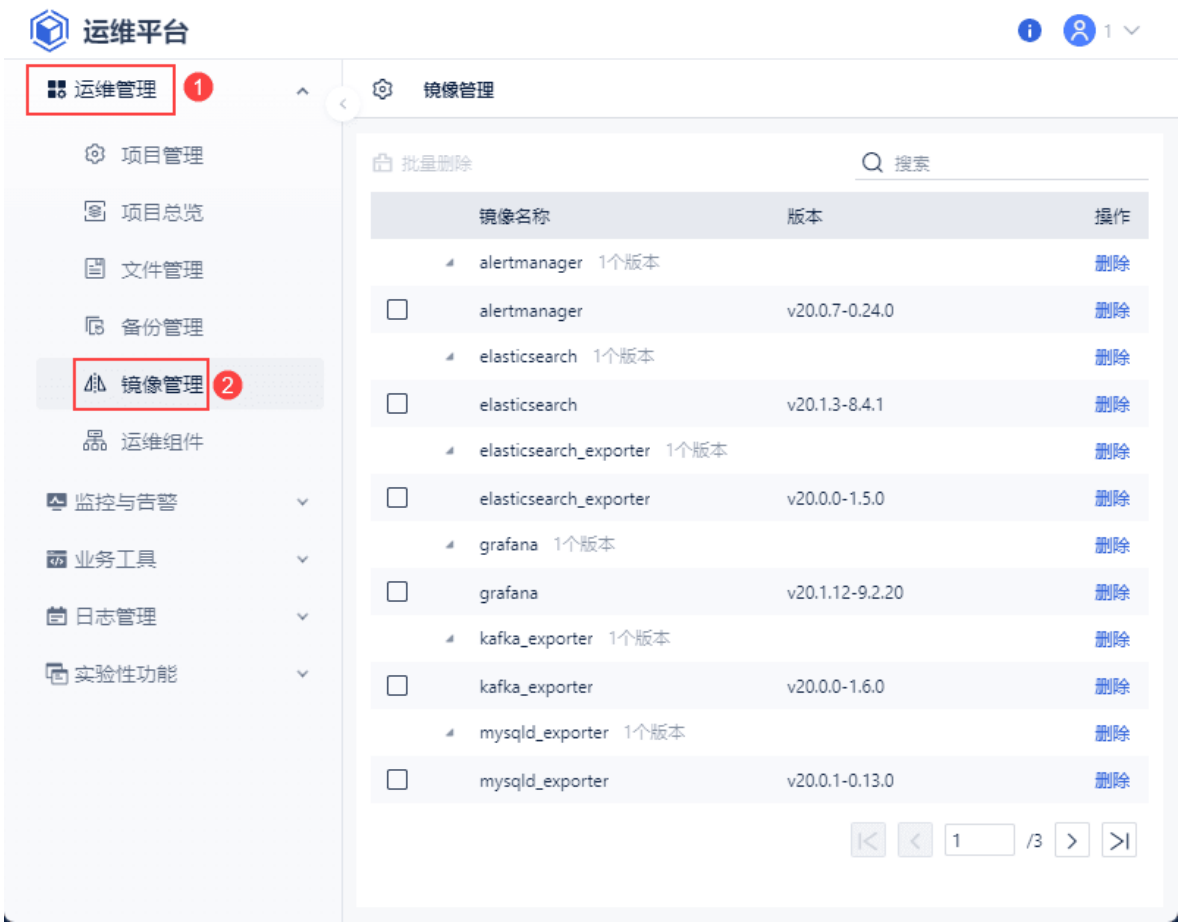
存储位置	文件夹	说明
项目挂载目录的 bi	bi-data	FineBI 抽取数据
	classes	工程调用的默认和自定义 class 文件
	config	平台调用的配置库（finedb）相关文件
	customlib	工程定制/外部引入的 JAR 包
文件夹	dpworks	FineDataLink 任务相关的配置文件 (如不存在该文件夹, 说明工程没有使用 FDL 相关功能)
(每个节点所在服	help	工程需要调用的一些 css、js、html、图片等资源
	local	存放新引擎的编译文件
	logs	swift 日志（即 logdb）
	mount	存放 setenv.sh 文件，用于设置工程的环境变量和启动参数
务器都存了一份，完	plugins	插件相关文件
	tomcat_logs	Tomcat 通用日志
		temp_attach：存放着工程 Excel 原始文件信息
		vcs：FineReport 模板备份文件
全相同)	assets	其他文件：通用的共享持久化目录
文件服务器	backup	工程历史备份文件
	reportlets	FineReport 模板存放目录
	resources	存放工程相关的资源配置文件
	schedule	定时调度生成的文件
	treasures	云端运维生成的数据包

### 3. 确认镜像仓库

部署新项目中的各个组件，是需要通过镜像进行部署的。要么镜像仓库已有镜像，要么支持从云端拉取镜像。

1) 管理员登录运维平台，点击「运维管理>镜像管理」，查看是否存在相关镜像。

- 必须安装：FineBI、node\_exporter、ops\_agent、nginx、nginx\_exporter、elasticsearch、elasticsearch\_exporter、filebeat
- 可选安装：mysql、mysqld\_exporter、redis、redis\_exporter、minio



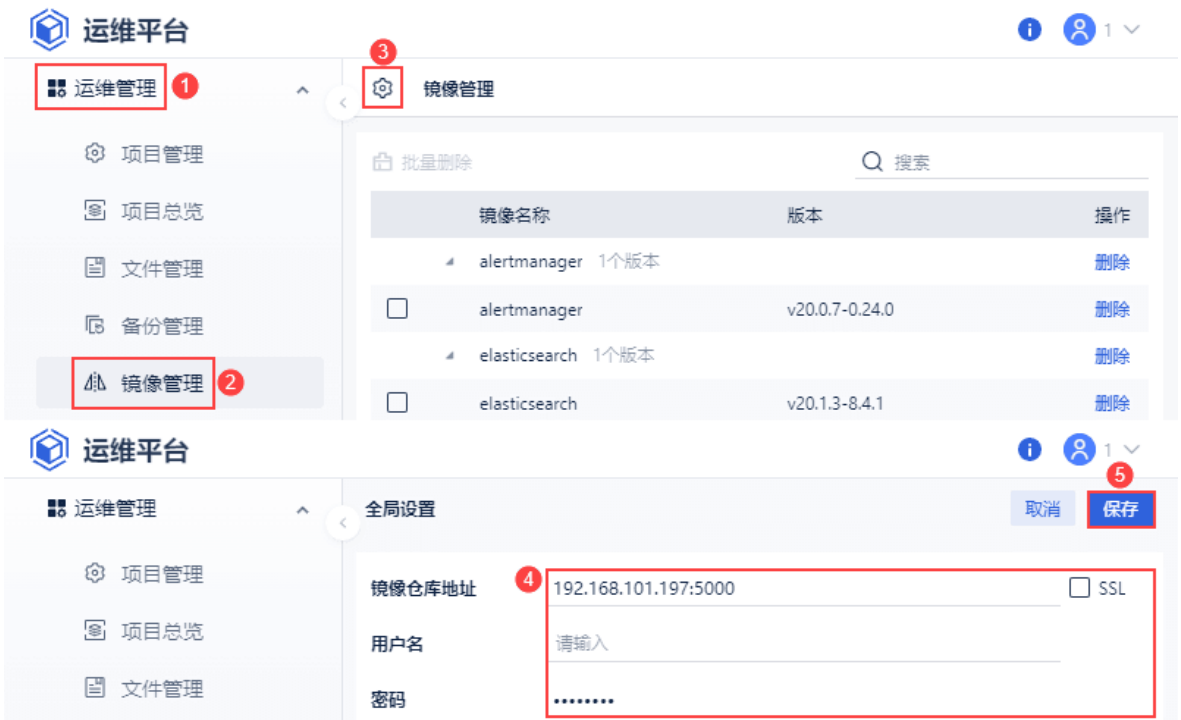
2) 如镜像仓库中不存在相关镜像，请查看镜像仓库地址，并确保地址可连通外网云端仓库。

- 查看镜像仓库地址：管理员登录运维平台，点击「运维管理>镜像管理>全局配置」，即可查看镜像仓库地址。
- 确保镜像仓库所在服务器支持访问以下外网地址：

域名：registry.cn-hangzhou.aliyuncs.com，端口：443

域名：fineops-registry.cn-hangzhou.cr.aliyuncs.com，端口：443

即可在部署项目时自动从云端拉取最新项目组件镜像。



#### 4. 部署新 FineBI 集群项目

本节正式进入部署 FineBI 集群项目的操作环节。请在开始部署前务必确认上文的所有内容，否则可能会部署失败！

##### 4.1 新增项目

管理员登录 FineOps 运维平台，点击「运维管理>项目管理」，点击「新增项目>部署新项目」，如下图所示：



##### 4.2 项目设置

项目设置是配置项目的类型、需要安装的组件、占用的端口等信息。

进入「项目设置」界面，输入相关项目信息，点击「下一步」，如下图所示：

部署新项目

1 项目设置

2 节点配置

项目名称

1

2

应用类型

☐ FineReport ☒ FineBI ☐ FineDataLink

应用类型

3 ☒ 集群 ☐ 单机

默认部署组件

☒ 运维基础套件 ☒ Nginx ☒ ElasticSearch

自定义部署组件

4 ☒ 文件服务器(MinIO) ☒ 外接数据库(MySQL) ☒ 状态服务器(Redis)

部署配置

☐ 使用默认配置 ☒ 自定义端口、组件用户名密码 [去设置](#)

访问路径

配置后应用的访问链接为{域名/IP}/{访问路径}/decision

取消

5 下一步

各配置项如下表所示：

信息	说明
	自定义项目的名称
	项目名称不允许为空
项目名称	项目名称仅支持中文、英文大小写、数字
	项目名称长度最大为 12 位
	项目名称不允许与已有项目重名
应用类型	选择「FineBI」
应用类型	选择「集群」
	即必须安装的组件，包括：
	1) 运维基础套件（包括工程节点）： FineBI、node_exporter、ops_agent
默认部署组件	2) Nginx（负载均衡，作为项目唯一入口）： nginx、nginx_exporter
	3) ElasticSearch（作为日志采集和存储）： elasticsearch、elasticsearch_exporter、filebeat
自定义部署组件	即可选安装的组件，包括：
件	1) 文件服务器：

## minio

如不安装该组件，用户必须自行准备好一个文件服务器，否则无法部署

支持的文件服务器类型包括：SFTP、FTP、共享外部目录（V1.10.0 及之前版本不支持共享外部目录）

如需使用其他类型文件服务器，可在项目部署成功后，登录工程切换

安装成功后，如需访问 MinIO，可参考：[MinIO 文件服务器简介](#)

## 2) 外接数据库:

## mysql、mysqld\_exporter

如不安装该组件，用户必须自行准备好一个外接配置库，否则无法部署

支持的数据库类型包括: MySQL5、MySQL8、Oracle (V1.14.0 及之前版本不支持 MySQL8)

如需使用其他类型外接库，可在项目部署成功后，登录工程切换

3) 状态服务器:

## redis、redis\_exporter

如不安装该组件，用户必须自行填写状态服务器信息，否则无法部署

支持两种：

### 1) 使用默认配置

2) 自定义端口、组件用户名密码，即对 2.5 节的默认端口、密码进行调整

请务必修改并记住 MySQL、Redis、minio、elasticsearch 组件的密码，该密码默认随机生成

部署新项目

1 项目设置

2 节点配置

部署配置

项目名称

fanruan1

应用类型

☐ FineReport ☒ FineBI ☐ FineDataLink

应用类型

☒ 集群 ☐ 单机

默认部署组件

☒ 运维基础套件 ☒ Nginx ☒ ElasticSearch

自定义部署组件

☒ 文件服务器(MinIO) ☒ 外接数据库(MySQL) ☒ 状态服务器

部署配置

☐ 使用默认配置 ☒ 自定义端口、组件用户名密码 [去设置](#)

访问路径

webroot

配置后应用的访问链接为[域名/IP]/[访问路径]/decision

访问路说明:

说明：  
访问路径 应用的访问链接为{域名/IP}/{访问路径}/decision

支持用户自定义应用的访问路径，访问路径默认为 webroot

举例：

一般应用访问链接为 http://IP:端口/webroot/decision

修改访问路径为 demo

则部署好的应用访问链接为 http://IP:端口/demo/decision

注意事项：

1.11.0 及之后版本，访问路径支持为空，即最终应用访问链接为 http://IP:端口/decision

但此时工程对应的 root 目录访问路径为：http://IP:端口/root

4.3 节点配置-新增主应用节点

配置完需要的组件和端口后，即可进入「节点配置」。

首先需要新增主应用节点，即为添加主应用服务器 ABC 的相关信息（也就是本章操作需要反复执行多次）。

1) 点击「新增节点」，选择节点类型为「主应用」，填写相关设置项，点击「测试连接」。



具体节点设置如下图所示：

节点设置	说明
------	----

请选择「主应用」

节点类型

主应用节点要求最低配置 8 核 32G、磁盘剩余 100GB 以上

注：帆软项目需要独占服务器，即该主应用服务器上不建议部署其他工程或内容，包括运维平台

主机

请输入节点主机 IP（内网 IP）

注：同一项目中，主机不支持重复添加多次

端口

请输入节点端口号，默认为 22

用户名

请输入有 sudo 权限的服务器用户名

验证方法

支持两种：密码、公钥

挂载路径

请输入节点安装目录的服务器路径，即 2.6 节设置的目录

默认为~/data，~代表所使用的服务器用户的主目录

注：可在终端使用该用户账号访问服务器，输入 echo \$HOME 命令查看该用户的主目录路径

选填项

外网 IP

如果服务器不支持内网访问，仅支持外网映射，可以填写可联通的外网 IP

2) 效果预览

若主机用户名等信息校验成功，即可添加一个主应用节点。三个节点均添加完成后，

如下图所示：

注 1：若网络环境不满足 2.3 节要求，节点连接失败，则出现报错提醒，无法正确新增节点。  
注 2：若服务器配置不符合 2.4 节最低要求，则出现报错提醒，无法正确新增节点。

部署新项目

✓ 项目设置

2 节点配置

提示：请部署至少两个主应用类型节点和一个组件类型节点

新增节点

IP地址	端口	节点类型	
192.168.101.79	22	主应用	<div></div> <div></div>
192.168.101.53	22	主应用	<div></div> <div></div>
192.168.101.58	22	主应用	<div></div> <div></div>

取消

上一步

开始部署

4.4 节点配置-新增组件节点

接着要新增组件节点，即为添加组件服务器 D 的相关信息。

1) 点击「新增节点」，选择节点类型为「组件」，填写相关设置项，点击「测试连接」。



具体节点设置如下图所示：

节点设置	说明
节点类型	<p>请选择「组件」</p> <p>组件节点要求最低配置 4 核 8G、磁盘剩余 50GB 以上</p> <p>注 1：如配置了多个组件节点，则组件随机部署在各个节点</p> <p>注 2：帆软项目需要独占服务器，即该组件服务器上不建议部署其他工程或内容，包括运维平台</p>
主机	<p>请输入节点主机 IP（内网 IP）</p> <p>注：同一项目中，主机不支持重复添加多次</p>
端口	<p>请输入节点端口号，默认为 22</p>
用户名	<p>请输入有 sudo 权限的服务器用户名</p>
验证方法	<p>支持两种：密码、公钥</p>
挂载路径	<p>请输入节点安装目录的服务器路径，即 2.6 节设置的目录</p> <p>默认为~/data，~代表所使用的服务器用户的主目录</p> <p>注：可在终端使用该用户账号访问服务器，输入 echo \$HOME 命令查看该用户的主目录路径</p>
外网 IP	<p>选填项</p> <p>如果服务器不支持内网访问，仅支持外网映射，可以填写可联通的外网 IP</p>

2) 效果预览



若主机用户名等信息校验成功，即可添加一个组件节点。如下图所示：

- 注 1：若网络环境不满足 2.3 节要求，节点连接失败，则出现报错提醒，无法正确新增节点。
- 注 2：若服务器配置不符合 2.4 节最低要求，则出现报错提醒，无法正确新增节点。



#### 4.5 开始部署

节点配置完成后，点击「开始部署」，进入部署过程。



弹窗中会显示各个组件的部署情况，若部署失败则显示失败原因。

- 注 1：部署过程中，用户可点击「隐藏至后台进行」，正常进行运维平台其他操作。
- 注 2：V1.15.0 及之后版本，支持在部署过程中点击「中止部署」按钮，停止部署并回退所有已有部署操作。



通过运维平台成功部署的新项目，自动接入运维平台的项目管理列表中，可进行相关运维操作。如下表所示：

其中的访问地址，即为该 FineBI 集群工程的访问地址，在浏览器新窗口访问该地址，即可进入工程初始化配置界面。



## 4.1.5 部署新 FineDataLink 单机项目

## 4.1.6 部署新 FineDataLink 集群项目

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.8.0	-
V1.9.0	部署交互界面优化
V1.10.0	部署进度界面优化，细分展示镜像拉取和组件部署进度
	1) 项目名称限制放宽，新增支持英文大写、中文
V1.11.0	2) 部署新集群项目时，支持使用外部共享目录作为文件服务器
	3) 项目配置中，访问路径支持为空
V1.14.0	部署的配置库组件，由 MySQL5 调整为 MySQL8
	1) 节点配置时，支持填写外网 IP，以适配不支持内网访问、仅可外网映射的服务器
V1.15.0	2) 支持在部署过程中「中止部署」，自动停止本次部署并回退已有操作

#### 1.2 功能简介

运维平台支持部署并直接接入全新 FineReport、FineBI、FineDataLink 应用。

本文将介绍，如何通过 [运维平台](#)，容器化部署全新的 **FineDataLink 集群**项目。

注 1：本文面向服务器资源不足或端口限制较高的用户，操作步骤较为详细，篇幅较长。

如能准备充足资源的**全新**服务器，可参考快速指南进行部署：[部署新帆软项目](#)

注 2：请在容器化部署前，先部署好 FineOps 运维平台。安装步骤请参见：[部署运维平台](#)。

### 2. 准备服务器

#### 2.1 确认需要安装的组件

通过运维平台容器化部署的 FineDataLink 集群项目，不仅仅包括 FineDataLink 工程，还包括配套的外接配置库、集群组件、运维组件等等。

请先确认好需要安装的组件。本文按照安装全部组件来进行示例讲解。

服务器	组件	是否安装	组件说明
主应用服务器	FineDataLink	必须	作用：FineDataLink 的工程节点组件，其中不带任何的 demo 数据

服务器	安装 据		说明： 必须安装
	作用：为服务器物理节点配套安装，用于管理组件，容器化升级		
	必须	也是由该组件完成	
	ops_agent	安装	说明：FineDataLink 项目涉及的每个服务器都会安装一个 ops_agent
node_exporter	作用：服务器物理节点的 exporter，收集物理服务器的各种指标		
	必须	信息	
	安装	说明： 每一个主应用服务器， 都会自动安装一个 node_exporter 组件	
	作用： 日志采集		
filebeat	必须	说明：与 elasticSearch 组件配套安装，在每个 FineDataLink 工程	
	安装	节点所在服务器安装 filebeat 组件	
	作用： 日志存储		
	说明： 替换 FineDataLink 中原有的 swift 作为日志存储		
elasticsearch	必须	对项目 FineDataLink、外接数据库、业务库和其他集群组件的	
	安装	请求进行采集	
	集群必须安装该组件，且请勿暂停该组件，否则会导致项目操作		
	日志丢失		
组件 服务器	elasticsearch_exporter	必须	作用： elasticSearch 的 exporter，收集 elasticSearch 的指标
	nginx	安装	说明： 与 elasticSearch 组件配套安装
		必须	作用： 提供集群的负载转发服务
	nginx_exporter	安装	说明： 集群其他组件服务的入口、使用，都从这个入口访问
nacos		必须	作用： nginx 服务的 exporter，收集 nginx 的指标信息
必须	安装	说明： 与 nginx 组件配套安装	
	作用： 服务注册中心		
安装	说明：保证用户调度端请求，即定时任务内节点的执行调度的高		

		可用性和高并发，能够合理分配到不同的集群节点上
nacos_mysql	必须安装	<p>作用：nacos 服务所需的 mysql 组件</p> <p>说明：与 nacos 组件配套安装</p> <p>作用：作为集群的外接配置库</p> <p>说明：可选安装</p> <p>如果安装该组件，会安装一个 MySQL8 数据库，作为集群的外接配置库</p>
mysql	可选安装	<p>如不安装该组件，用户必须自行准备好一个外接配置库，否则无法部署</p> <p>支持的数据库类型包括：MySQL5、MySQL8、Oracle（V1.14.0 及之前版本不支持 MySQL8）</p> <p>如需使用其他类型外接库，可在项目部署成功后，登录工程切换</p>
mysqld_exporter	可选安装	<p>作用：MySQL 服务的 exporter，收集 MySQL 的各种指标信息</p> <p>说明：与 MySQL 组件配套安装</p> <p>作用：作为集群的状态服务器</p> <p>说明：可选安装</p>
redis	可选安装	<p>如果安装该组件，会安装一个单机 redis，作为集群的状态服务器</p> <p>如不安装该组件，用户必须自行准备好一个 redis 组件（单机集群均可），否则无法部署</p>
redis_exporter	可选安装	<p>作用：redis 服务的 exporter，收集 redis 的各种指标信息</p> <p>说明：与 redis 组件配套安装</p> <p>作用：作为集群的文件服务器</p> <p>说明：可选安装</p>
minio	可选安装	<p>如果安装该组件，会安装一个 minio，它是一款高性能、分布式的对象存储系统，作为集群的文件服务器</p> <p>如不安装该组件，用户必须自行准备好一个文件服务器，否则无法部署</p> <p>支持的文件服务器类型包括：SFTP、FTP、共享外部目录（V1.10.0 及之前版本不支持共享外部目录）</p>

		如需使用其他类型文件服务器，可在项目部署成功后，登录工程切换
		作用：支撑管道任务相关功能
Kafka	可选安装	说明：可选安装 如不安装该组件，将无法使用 FineDataLink 中管道任务相关功能
Kafka_exporter	可选安装	作用：Kafka 的 exporter，收集 Kafka 的指标 说明：与 Kafka 组件配套安装
ops_agent	必须安装	作用：为服务器物理节点配套安装，用于管理组件，容器化升级也是由该组件完成 说明：FineDataLink 项目涉及的每个服务器都会安装一个 ops_agent 作用：服务器物理节点的 exporter，收集物理服务器的各种指标信息
node_exporter	必须安装	说明：每一个组件服务器，都会自动安装一个 node_exporter 组件

## 2.2 服务器数量要求

本文按照 4 个服务器进行下文的部署操作, 分为主应用服务器 ABC、组件服务器 D。

请准备好服务器的内网 IP、用户名、密码。

注：FineOps 运维平台，和 FineDataLink 项目，不要部署在同一服务器下。

如果 FineOps 和项目处于同一环境下, 当项目服务器发生异常时, FineOps 可能也出现故障, 从而无法起到运维的作用。

类型	要求
建议按照「节点数量+1」准备服务器数量，这些服务器中不再安装其他工程/产品	
1) 每个工程节点独占一个服务器：	
推荐配置	<ul style="list-style-type: none"><li>例如本文示例的是 3 节点集群，则需要 3 个主应用服务器。</li><li>每个服务器中安装一套 FineDataLink、ops_agent、node_exporter、filebeat</li></ul>
2) 其他组件独占一个服务器：即组件服务器	

- 该服务器中安装外接配置库（mysql）、Kafka 组件（kafka）、nacos 组件（nacos、nacos\_mysql）、集群组件（nginx、redis、minio）、es 组件（elasticsearch）以及运维组件（expoder 组件）
- 如果服务器资源充足，可以多准备一些组件服务器，在部署时会随机分配集群组件到各个服务器中，确保项目高可用

建议按照「节点数量+7」准备服务器数量，这些服务器中不再安装其他工程/产品

- 高可用配置
- 1) 每个工程节点独占一个服务器，作为主应用服务器
  - 2) redis、nginx、mysql、minio、es、kafka、nacos 组件各自独占一个服务器，作为组件服务器

2.3 服务器网络要求

请确保 FineDataLink 项目涉及的主应用服务器 ABC、组件服务器 D，以及 FineOps 运维平台所在的服务器 E：

这些服务器内网互相联通，如不联通则无法成功部署。

注：如自行准备了外接配置库、状态服务器、文件服务器。  
他们所在的服务器 F、G、H，也需要和 ABCDE 服务器内网互通。

2.4 服务器系统配置

请确保 FineDataLink 项目涉及的主应用服务器 ABC、组件服务器 B，满足此基础配置。

注意，下表的配置要求是针对每个服务器，而不是所有服务器的配置总和。

服务器	配置	要求
主应用服务器	系统版本	CentOS 7.3~7.9 版本
		Ubuntu 18.04.4 及以上版本
		redhat 7.6 及以上版本
	系统内核	3.10 及以上
		最低 4 核
	CPU 架构	仅支持 X86_64
	剩余空闲物理内存	8G 以上
	剩余空闲磁盘容量	50GB 以上



		CentOS 7.3~7.9 版本
系统版本		Ubuntu 18.04.4 及以上版本
		redhat 7.6 及以上版本
系统内核		3.10 及以上
		最低 4 核
组件服务器	CPU 架构	仅支持 X86_64
		最低 8G
	剩余空闲物理内存	若仅准备了一个 1 个组件服务器，建议 16G 以上
	剩余空闲磁盘容量	50GB 以上 如使用管道功能，建议 1T 以上

注：服务器之所以要满足上表所说的内存和磁盘，是由于 FineDataLink 项目中的各个组件有最低安装配置要求。

[点击展开更多](#)

服务器	组件	CPU	内存	xmx	磁盘
主应用服务器				默认内存为：	
	FineDataLink	4Core	1.5xmx	机器内存减掉 8G 后的内存大小	50G
				最大不会超过 30G	
组件服务器	ops_agent	1Core	-	1G	-
	node_exporter	0.1Core	0.1G	-	-
	filebeat	0.1Core	0.1G	-	-
	elasticsearch	1Core	-	2G	50G
	elasticsearch_exporter	0.1Core	0.1G	-	-
	nginx	1Core	-	-	-
	nginx_exporter	0.1Core	0.1G	0.1G	-
	nacos	2Core	2G	-	-
	nacos_mysql	2Core	2G	-	-
	mysql	2Core	-	-	50G
	mysqld_exporter	0.1Core	0.1G	-	-
	redis	1Core	-	4G	-
	redis_exporter	0.1Core	0.1G	-	-
	minio	1Core	-	-	100G
	Kafka	4Core	-	2G	-
	Kafka_exporter	0.1Core	0.1G	-	-

ops_agent	1Core	-	1G	-
node_exporter	0.1Core	0.1G	-	-

### 2.5 服务器端口要求

请确保 FineDataLink 项目涉及的主应用服务器 ABC、组件服务器 D 中，待部署应用和组件的默认端口可用。

- 请确认默认的映射端口是否已被使用，如已被使用，请安排好其他空闲端口。
- nginx 的映射端口将作为项目访问端口，因此请确认准备使用的 nginx 映射端口是否开放了防火墙、安全组等，确保可连接。

注 1：端口占用和防火墙开放操作方法可参考：[端口占用检查与防火墙配置](#)。

注 2：端口 portMapping 为 {映射端口:容器内端口}，其中映射端口可改, 容器内端口不支持修改。

注 3：如果准备了多个组件服务器，组件会随机分布部署在各个服务器下。因此请确保每一个组件服务器都满足各个组件的端口要求。

服务器	组件	默认端口
主应用服务器	FineDataLink	端口：{7800:7800,7830:7830,7840:7840,7850:7850,7870:7870,8068:8068,12100:12100,15500:15500}
	ops_agent	端口：{9070:9070}
	node_exporter	端口：{9100:9100}
	filebeat	不占用端口
		端口：{9200:9200}
组件服务器	elasticsearch	用户名：elastic（不支持修改）  密码：elasticadmin（支持修改）
	elasticsearch_exporter	端口：{9114:9114}
		端口：{80: 8080}
	nginx	非 root 用户建议修改占用端口, 因为非 root 用户默认无权限使用 1024 以下端口
	nginx_exporter	端口：{9113: 9113}
		http 端口：{8848:8848}
	nacos	grpc 端口：{9848:9848}
		用户名：nacos（不支持修改）

	密码：随机生成，届时请务必修改
	端口： {3307:3306}
nacos_mysql	用户名： root （不支持修改）
	密码：随机生成，届时请务必修改
	端口： {3306:3306}
mysql	用户名： root （不支持修改）
	密码：随机生成，届时请务必修改
mysqld_exporter	端口： {9104:9104}
	端口： {6379:6379}
redis	用户名： admin （不支持修改）
	密码：随机生成，届时请务必修改
redis_exporter	端口： {9121:9121}
	服务器端口： {9000:9000}
	控制台端口： {9001:9001}
minio	用户名： minioadmin (不支持修改)
	密码： minioadmin （支持修改，但
	密码必须大于等于 8 位，否则使用
	默认密码)
	端口： {9092:9092}
	默认无认证
Kafka	用户可调整为密码认证，自定义用
	户名密码
Kafka_exporter	端口： {9308:9308}
ops_agent	端口： {9070:9070}
node_exporter	端口： {9100:9100}

2.6 设置项目挂载目录

容器化部署的 FineDataLink 项目，重要文件均外挂。请提前在主应用服务器 ABC 和组件服务器 D 的空闲磁盘下，新建好挂载目录。

1) 查看服务器磁盘空间

命令：df -h，找出可用空间较大的挂载点

例如本示例中，服务器最合适的挂载目录为/home 目录下

```
Last login: Tue Jul 25 10:16:49 2023 from 192.168.5.2
[root@192 ~]# df -h
文件系统              容量  已用  可用 已用% 挂载点
/dev/mapper/centos-root 50G   14G   37G   28% /
devtmpfs              9.8G   0    9.8G   0% /dev
tmpfs                 9.8G   0    9.8G   0% /dev/shm
tmpfs                 9.8G  962M   8.8G   10% /run
tmpfs                 9.8G   0    9.8G   0% /sys/fs/cgroup
/dev/mapper/centos-home 142G   12G  130G    9% /home
/dev/sda1             1014M  146M   869M   15% /boot
tmpfs                 2.0G   0    2.0G   0% /run/user/0
```

2) 新建文件夹

- 使用 mkdir 命令创建文件夹（本示例在/home 目录下创建一个名叫 fanruan 的文件夹）：mkdir /home/fanruan
- 使用 chmod 命令设置文件夹的权限（本示例直接赋 777 权限）：  
chmod 777 /home/fanruan

```
[root@192 ~]#
[root@192 ~]# mkdir /home/fanruan
[root@192 ~]# chmod 777 /home/fanruan
[root@192 ~]#
[root@192 ~]#
[root@192 ~]# cd /home
[root@192 home]# ls
fanruan finekey finekey-operation-all.tar.gz fr
[root@192 home]#
```

注：FineDataLink 集群项目部署成功后，文件服务器和项目挂载目录中的 fdl 文件夹中，分别存放着工程外挂内容，其中的文件夹和存放内容如下表所示。

[点击展开更多](#)

存储位置	文件夹	说明
项目挂载目录的 fdl 文件夹	config	平台调用的配置库（finedb）相关文件
	customlib	工程定制/外部引入的 JAR 包
	dpworks	FineDataLink 任务相关的配置文件（如不存在该文件夹，说明工程没有使用 FDL 相关功能）
	logs	swift 日志（即 logdb）
	mount	存放 setenv.sh 文件，用于设置工程的环境变量和启动参数
（每个节点所在服务器都存了一份，完全相同）	plugins	插件相关文件
	tomcat_logs	Tomcat 通用日志
	assets	通用的共享持久化目录
	backup	工程历史备份文件
	resources	存放工程相关的资源配置文件
文件服务器	schedule	定时调度生成的文件
	treasures	云端运维生成的数据包

3. 确认镜像仓库

部署新项目中的各个组件，是需要通过镜像进行部署的。要么镜像仓库已有镜像，要么支持从云端拉取镜像。

运维平台默认不提供 FineDataLink、Kafka 的镜像包。

付费用户请联系帆软技术支持，获取包含 FineDataLink、Kafka 相关镜像包的 finekey 工具，并参考文档更新运维平台：[内网升级运维平台](#)。

技术支持联系方式：前往「[服务](#)」，选择「在线支持」、电话「400-811-8890」。

## 4. 部署新 FineDataLink 集群项目

本节正式进入部署 FineDataLink 集群项目的操作环节。请在开始部署前务必确认上文的所有内容，否则可能会部署失败！

### 4.1 新增项目

管理员登录 FineOps 运维平台，点击「运维管理>项目管理」，点击「部署新项目」，如下图所示：



### 4.2 项目设置

项目设置是配置项目的类型、需要安装的组件、占用的端口等信息。

进入「项目设置」界面，输入相关项目信息，点击「下一步」，如下图所示：

部署新项目

1 项目设置

2 节点配置

项目名称

1

应用类型

☐ FineReport

☐ FineBI

☒ FineDataLink 2

项目类型

3 

☒ 集群

☐ 单机

默认部署组件

☒ 运维基础套件

☒ Nginx

☒ ElasticSearch

☒ Nacos

自定义部署组件

4 

☒ 文件服务器(MinIO)

☒ 外接数据库(MySQL)

☒ 状态服务器(Redis)

☒ Kafka(管道任务)

部署配置

☐ 使用默认配置

☒ 自定义端口、组件用户名密码 [去设置](#)

访问路径

配置后应用的访问链接为(域名/IP)/(访问路径)/decision

取消

5 下一步

各配置项如下表所示：

信息	说明
	自定义项目的名称
	项目名称不允许为空
项目名称	项目名称仅支持中文、英文大小写、数字
	项目名称长度最大为 12 位
	项目名称不允许与已有项目重名
应用类型	选择「FineDataLink」
项目类型	选择「集群」
	即必须安装的组件，包括：
	1) 运维基础套件（包括工程节点）：
	FineDataLink、node_exporter、ops_agent
	2) Nginx（负载均衡，作为项目唯一入口）：
默认部署组件	nginx、nginx_exporter
	3) ElasticSearch（作为日志采集和存储）：
	elasticsearch、elasticsearch_exporter、filebeat
	4) Nacos（作为服务注册）：
	nacos、nacos_mysql

集群包括多种组件：

1) 文件服务器：

minio

如不安装该组件，用户必须自行准备好一个文件服务器，否则无法部署  
支持的文件服务器类型包括：SFTP、FTP、共享外部目录（V1.10.0 及之前版本不支持共享外部目录）  
如需使用其他类型文件服务器，可在项目部署成功后，登录工程切换

安装成功后，如需访问 MinIO，可参考：[MinIO 文件服务器简介](#)

2) 外接数据库：

mysql、mysqld\_exporter

需要部署的套件  
如不安装该组件，用户必须自行准备好一个外接配置库，否则无法部署  
支持的数据库类型包括：MySQL5、MySQL8、Oracle（V1.14.0 及之前版本不支持 MySQL8）  
如需使用其他类型外接库，可在项目部署成功后，登录工程切换

3) 状态服务器：

redis、redis\_exporter

如不安装该组件，用户必须自行准备好一个 redis 状态服务器（单机集群均可），否则无法部署

4) Kafka：

Kafka、Kafka\_exporter

如不安装该组件，将无法使用 FineDataLink 中管道任务相关功能  
支持两种：

1) 使用默认配置

2) 自定义端口、组件用户名密码，即对 2.5 节的默认端口、密码进行调整

请务必修改并记住 MySQL、Redis、minio、elasticsearch、Nacos 组件的密码，该密码默认随机生成

部署新项目

1 项目设置

2 节点配置

项目名称

fanruan1

应用类型

☐ FineReport

☐ FineBI

☒ FineDataLink

应用类型

☒ 集群

☐ 单机

默认部署组件

☒ 运维基础套件

☒ Nginx

☒ ElasticSearch

☒ Nacos

自定义部署组件

☒ 文件服务器(MinIO)

☒ 外接数据库(MySQL)

☒ 状态服务器

部署配置

☐ 使用默认配置

☒ 自定义端口、组件用户名密码

访问路径

webroot

配置后应用的访问链接为(域名/IP)/访问路径/decision

部署配置自定义

主机端口 12100

容器端口 12100

主机端口 15500

容器端口 15500

redis

主机端口 6379

容器端口 6379

密码 .....

redis\_exporter

主机端口 9121

容器端口 9121

mysql

主机端口 3306

容器端口 3306

用户名 root

密码 .....

取消

完成

去设置

部署配置

取消

下一步

说明：

应用的访问链接为 {域名/IP}/{访问路径}/decision

支持用户自定义应用的访问路径，访问路径默认为 webroot

举例：

一般应用访问链接为 http://IP:端口/webroot/decision

访问路径

修改访问路径为 demo

则部署好的应用访问链接为 http://IP:端口/demo/decision

注意事项：

1.11.0 及之后版本，访问路径支持为空，即最终应用访问链接为 http://IP:端口/decision

但此时工程对应的 root 目录访问路径为：http://IP:端口/root

4.3 节点配置-新增主应用节点

配置完需要的组件和端口后，即可进入「节点配置」。

首先需要新增主应用节点，即为添加主应用服务器 ABC 的相关信息（也就是本章操作需要反复执行多次）。

1) 点击「新增节点」，选择节点类型为「主应用」，填写相关设置项，点击「测试连接」。



具体节点设置如下图所示：

节点设	说明
-----	----



置

请选择「主应用」

节点类型

主应用节点要求最低配置 4 核 8G、磁盘剩余 50GB 以上

注：帆软项目需要独占服务器，即该主应用服务器上不建议部署其他工程或内容，包括运维平台

主机

请输入节点主机 IP（内网 IP）

注：同一项目中，主机不支持重复添加多次

端口

请输入节点端口号，默认为 22

用户名

请输入有 sudo 权限的服务器用户名

验证方法

支持两种：密码、公钥

请输入节点安装目录的服务器路径，即 2.6 节设置的目录

挂载路径

默认为~/data，~代表所使用的服务器用户的主目录

注：可在终端使用该用户账号访问服务器，输入 echo \$HOME 命令查看该用户的主目录路径

选填项

外网 IP

如果服务器不支持内网访问，仅支持外网映射，可以填写可联通的外网 IP

2) 效果预览

若主机用户名等信息校验成功，即可添加一个主应用节点。三个节点均添加完成后，

如下图所示：

注 1：若网络环境不满足 2.3 节要求，节点连接失败，则出现报错提醒，无法正确新增节点。  
注 2：若服务器配置不符合 2.4 节最低要求，则出现报错提醒，无法正确新增节点。



4.4 节点配置-新增组件节点

接着要新增组件节点，即为添加组件服务器 D 的相关信息。

1) 点击「新增节点」，选择节点类型为「组件」，填写相关设置项，点击「测试连接」。



具体节点设置如下图所示：

节点设置	说明
节点类型	<p>请选择「组件」</p> <p>组件节点要求最低配置 4 核 8G、磁盘剩余 50GB 以上</p> <p>注 1：如配置了多个组件节点，则组件随机部署在各个节点</p> <p>注 2：如需要启用管道功能，安装 Kafka 组件，则组件服务器可用磁盘空间建议 1T 以上</p> <p>注 3：帆软项目需要独占服务器，即该组件服务器上不建议部署其他工程或内容，包括运维平台</p>
主机	<p>请输入节点主机 IP（内网 IP）</p> <p>注：同一项目中，主机不支持重复添加多次</p>
端口	请输入节点端口号，默认为 22
用户名	请输入有 sudo 权限的服务器用户名
验证方法	支持两种：密码、公钥
挂载路径	<p>请输入节点安装目录的服务器路径，即 2.6 节设置的目录</p> <p>默认为~/data，~代表所使用的服务器用户的主目录</p> <p>注：可在终端使用该用户账号访问服务器，输入 echo \$HOME 命令查看该用户的主目录路径</p>
外网 IP	<p>选填项</p> <p>如果服务器不支持内网访问，仅支持外网映射，可以填写可联通的外网 IP</p>

2) 效果预览

若主机用户名等信息校验成功，即可添加一个组件节点。如下图所示：

- 注 1：若网络环境不满足 2.3 节要求，节点连接失败，则出现报错提醒，无法正确新增节点。
- 注 2：若服务器配置不符合 2.4 节最低要求，则出现报错提醒，无法正确新增节点。



#### 4.5 开始部署

节点配置完成后，点击「开始部署」，进入部署过程。



弹窗中会显示各个组件的部署情况，若部署失败则显示失败原因。

- 注 1：部署过程中，用户可点击「隐藏至后台进行」，正常进行运维平台其他操作。
- 注 2：V1.15.0 及之后版本，支持在部署过程中点击「中止部署」按钮，停止部署并回退所有已有部署操作。



通过运维平台成功部署的新项目，自动接入运维平台的项目管理列表中，可进行相关运维操作。如下表所示：

其中的访问地址，即为该 FineDataLink 集群工程的访问地址，在浏览器新窗口访问该地址，即可进入工程初始化配置界面。



## 4.2 接入已有项目

## 4.2.1 接入运维平台已部署的项目

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.5.5	-
	帆软应用只允许被一个运维平台接入
V1.5.6	若已被接入，应用的运维密钥处显示「已接入」
	1) 接入项目交互界面优化
V1.9.0	2) 删除冗余操作步骤：组件信息设置、服务器信息设置
V1.11.0	项目名称限制放宽，新增支持英文大写、中文

#### 1.2 功能简介

FineOps 运维平台是帮助帆软应用稳定运行和全方位运维管理的解决方案。

因此，运维平台支持接入已部署好的 FineReport、FineBI、FineDataLink 等项目。

本文将带用户体验如何将容器化部署的项目接入运维平台。

#### 1.3 步骤简介

步骤	说明
	1) 运维平台仅支持接入以下版本工程
	<ul style="list-style-type: none"><li>FineReport11.0.6 及以上版本</li><li>FineBI5.1.22 及以上版本</li><li>FineDataLink4.0.26 及以上版本</li></ul>
1 检查是否支持接入	2) 运维平台和需要被管理的 FR/BI/FDL 应用，所在的服务器必须互相内网可联通
2 准备材料	1) 获取 FR/BI/FDL 应用的访问地址 2) 获取 FR/BI/FDL 应用的运维接口密钥
3 接入已有项目	登录运维平台，在「项目管理」界面选择「接入已有项目」 自定义项目名称，填入第二步准备的信息
4 接入成功	检测以上步骤是否有问题，没问题的情况下会成功接入

在「项目管理」界面会出现新接入的 FR/BI/FDL 项目

[helpvideo]6562[/helpvideo]

2. 使用前提

2.1 检查应用版本

要求说明：

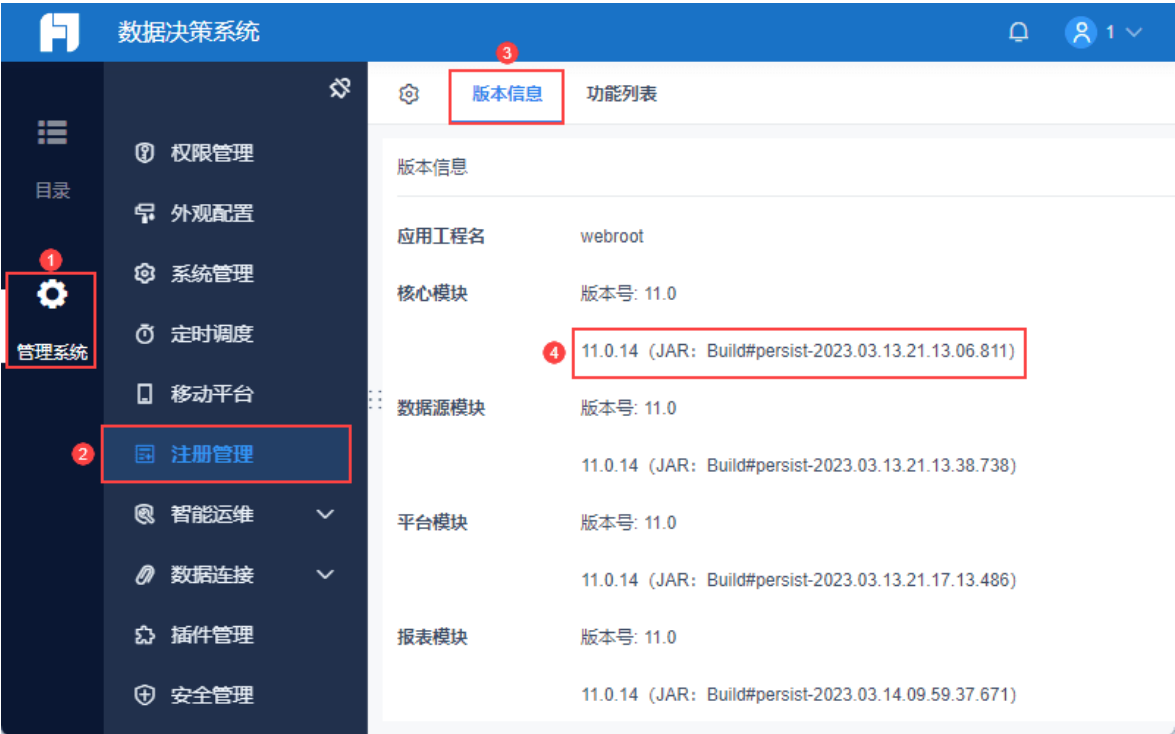
如需对接运维平台，请确保应用版本在 FineReport11.0.6/FineBI5.1.22/FineDataLink4.0.26 及以上。

检查方法：

管理员登录 FineReport/FineBI/FineDataLink 应用，点击「管理系统>注册管理>版本信息」，即可查看工程各个模块的小版本号。

解决方案：

请参考 [FineReport 小版本升级指南](#)、[FineBI 小版本升级指南](#)、[FineDataLink 一键升级](#)，升级工程到最新版本。



2.2 检查网络是否连通

请确保运维平台和需要被管理的应用，所在的服务器互相内网可联通。若网络不互通，则无法对接/影响后续功能使用。

如需精确到指定端口互通，请确保以下端口可以互相访问，若网络不互通，则无法对接/影响后续功能使用。

互通端口	开放原因
------	------

FR/BI/FDL 应用的外接数据库 MySQL 组件端口（默认端口 3306）	采集组件状态
FR/BI/FDL 应用的 mysqld_exporter 组件端口（默认端口 9104）	
FR/BI/FDL 应用的状态服务器 redis 组件端口（默认端口 6379）	采集组件状态
FR/BI/FDL 应用的 redis_exporter 组件端口（默认端口 9121）	
FR/BI/FDL 应用的负载均衡 nginx 组件端口（默认端口 80）	采集组件状态
FR/BI/FDL 应用的 nginx_exporter 组件端口（默认端口 9113）	
运维平台的 prometheus 组件端口（默认端口 9090）	组件指标传递到监
FR/BI/FDL 应用的 mysqld_exporter 组件端口（默认端口 9104）	控
运维平台的 prometheus 组件端口（默认端口 9090）	组件指标传递到监
FR/BI/FDL 应用的 redis_exporter 组件端口（默认端口 9121）	控
运维平台的 prometheus 组件端口（默认端口 9090）	组件指标传递到监
FR/BI/FDL 应用的 nginx_exporter 组件端口（默认端口 9113）	控
运维平台的 prometheus 组件端口（默认端口 9090）	组件指标传递到监
FR/BI/FDL 应用的 node_exporter 组件端口（默认端口 9100）	控
运维平台的负载均衡 nginx 组件端口（默认端口 80）	
FR/BI/FDL 应用的工程端口	确保应用管理和监 控
FR/BI/FDL 应用的 ops-agent 组件端口（默认端口 9070）	

3. 准备清单

3.1 准备运维接口密钥

要求说明：

项目管理的前提是，运维平台获得应用的认可。因此首先要从应用中获取一个运维接口密钥。

获取方法：

管理员登录 FineReport/FineBI 应用, 点击「管理系统>智能运维>健康巡检>运维工具」。

设置一个运维接口密钥，点击「保存」即可。

运维接口密钥请设置为大小写字母或数字，长度在 8~14 之间。请记住此次设置的密钥，下文需要用到。



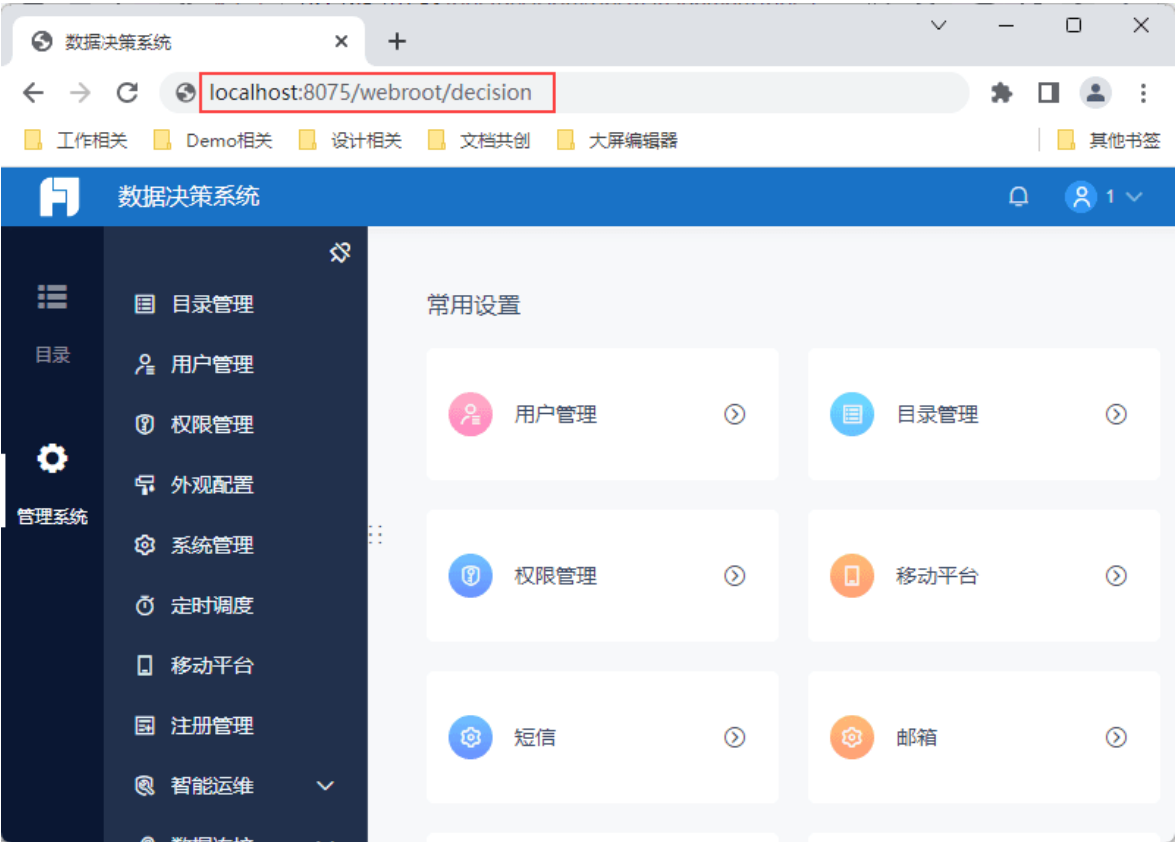
注：帆软应用只允许被一个运维平台接入。若已被接入，应用的运维密钥处显示「已接入」。



### 3.2 准备应用地址

应用地址，即为 FineReport/FineBI 应用的访问地址，形如 `http://IP:端口号/工程名/decision`

注：请勿准备短域名地址作为应用地址，可能会造成接入失败。



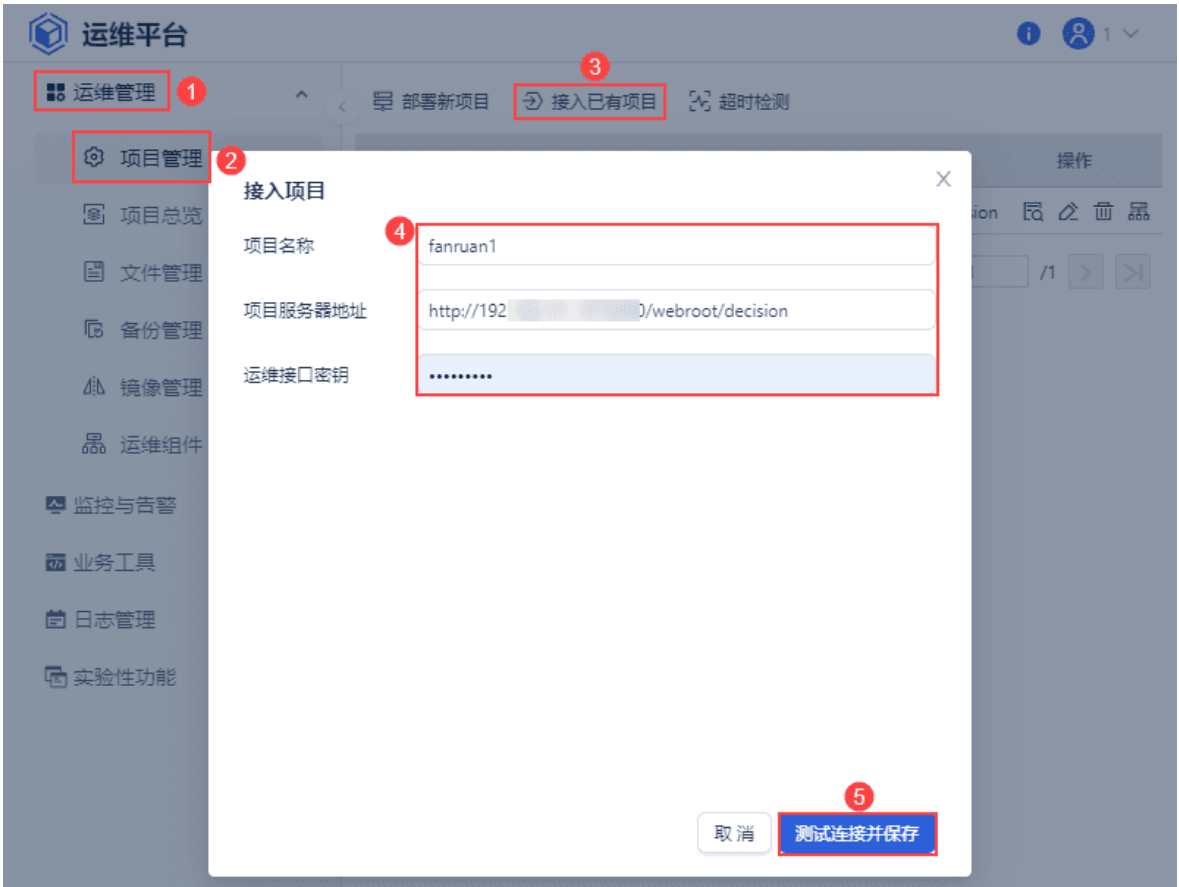
## 4. 接入已有项目

- 1) 管理员登录运维平台，点击「运维管理>项目管理」，点击「接入已有项目」按钮。
- 2) 设置相关项目信息，需要设置的应项目信息如下表所示：

信息	说明
----	----

	自定义项目的名称
项目名称	项目名称不允许为空 项目名称仅支持中文、英文大小写、数字 项目名称长度最大为 12 位
项目服务器地址	FineReport、FineBI 等项目的地址(3.2 节准备的信息) 项目服务器地址形如 http://IP:port/webroot/decision 项目服务器地址不允许为空 请勿准备短域名地址作为应用地址，可能会造成接入失败
运维接口密钥	运维接口密钥（3.1 节准备的信息） 运维接口密钥不允许为空

3) 点击「测试连接并保存」。如下图所示：



4) 连接成功后提示「接入项目成功」，在项目管理列表中出现新接入的项目，即代表接入成功，如下图所示：

运维平台

项目管理

项目总览

文件管理

备份管理

镜像管理

部署新项目

接入已有项目

超时检测

名称

类型

存活节点数

部署方式

状态

访问地址

操作

bijiqun

集群

3

运维平...

可用

http://192.168.1.100/decision

fr11jiqun

集群

2

非运维...

可用

http://192.168.1.101/root/decision

81dan

单机

1

运维...

可用

http://192.168.1.102/webroot/decision

<<

<

1

/1

>

>>

## 4.2.2 接入非运维平台部署的项目

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.5.5	-
V1.5.6	1) 组件信息设置时，允许修改组件地址和端口
	2) 服务器信息设置时，不再安装冗余组件 promtail
	3) 帆软应用只允许被一个运维平台接入
	若已被接入，应用的运维密钥处显示「已接入」
V1.9.0	接入项目交互界面优化
V1.10.0	项目名称限制放宽，新增支持英文大写、中文

#### 1.2 功能简介

FineOps 运维平台是帮助帆软应用稳定运行和全方位运维管理的解决方案。

因此，运维平台支持接入已部署好的 FineReport、FineBI 等项目。

本文将带用户体验如何将非容器化部署的项目接入运维平台。

[helpvideo]6563[/helpvideo]

#### 1.3 步骤简介

步骤		说明
1	检查是否支持接入	1) 运维平台仅支持接入 FineReport11.0.6/FineBI5.1.22 及以上版本工程
		2) 运维平台和需要被管理的 FR/BI 应用，所在的服务器必须互相内网可联通
2	接入已有项目	登录运维平台，在「项目管理」界面选择「接入已有项目」
3	是否必要	必要步骤
	帮助	帮助运维平台与 FR/BI 应用连接
	项目信息设置	1) 获取 FR/BI 应用的访问地址 2) 获取 FR/BI 应用的运维接口密钥
	前提准备	
	操作步骤	自定义项目名称，填入上一步准备的信息

		非必要步骤
		在运维平台所在服务器安装 exporter 组件，为该工程的外接数据库、负载均衡、状态服务器进行日志采集
4	组件信息设置	是否必要 可安装的 exporter 组件包括：mysqld_exporter、redis_exporter、nginx_exporter  如不安装，可直接点击「跳过」进入下一步，不影响运维平台与工程的对接，但后续运维平台将无法监控这些组件的状态并进行日志采集
		前提准备 1) 确认 FR/BI 应用中是否有符合要求的外接数据库、负载均衡、状态服务器组件  如果没有这些组件，就完全可以跳过这个步骤
		操作步骤 2) 确认 MySQL、Redis、Nginx 组件的端口可与运维平台内网联通  3) 准备 Nginx 组件的服务器地址、端口、SSL 信息  4) 为 FR/BI 应用安装 Nginx 状态监控模块
		非必要步骤 1) 自动设置 MySQL 和 Redis 组件信息 2) 手动填写 Nginx 组件信息
5	服务器信息设置	是否必要 在 FR/BI 应用所在服务器安装 node_exporter 组件，为该工程进行服务器信息采集  可以为应用的每一个节点、每一个组件所在服务器都安装一份 node_exporter 组件  如不安装上述组件，可直接点击「跳过」进入下一步，不影响运维平台与工程的对接，但后续运维平台将无法监控服务器状态并进行日志采集
		前提准备 1) 准备所有需要安装组件的节点服务器的主机 IP、端口、root 用户密码

- 2) 在所有需要安装组件的节点服务器，都准备好一个目录，用于存放 exporter 日志
- 2) 准备 node\_exporter 需要占用的端口可用性
- 3) 确保 node\_exporter 端口可与运维平台内网联通

操作步骤  
将上一步准备的所有服务器的相关信息填入

6 接入成功

检测以上步骤是否有问题，没问题的情况下会成功接入  
在「项目管理」界面会出现新接入的 FR/BI 项目

2. 基础使用前提

2.1 检查应用部署方式和版本

要求说明：

如需对接运维平台：

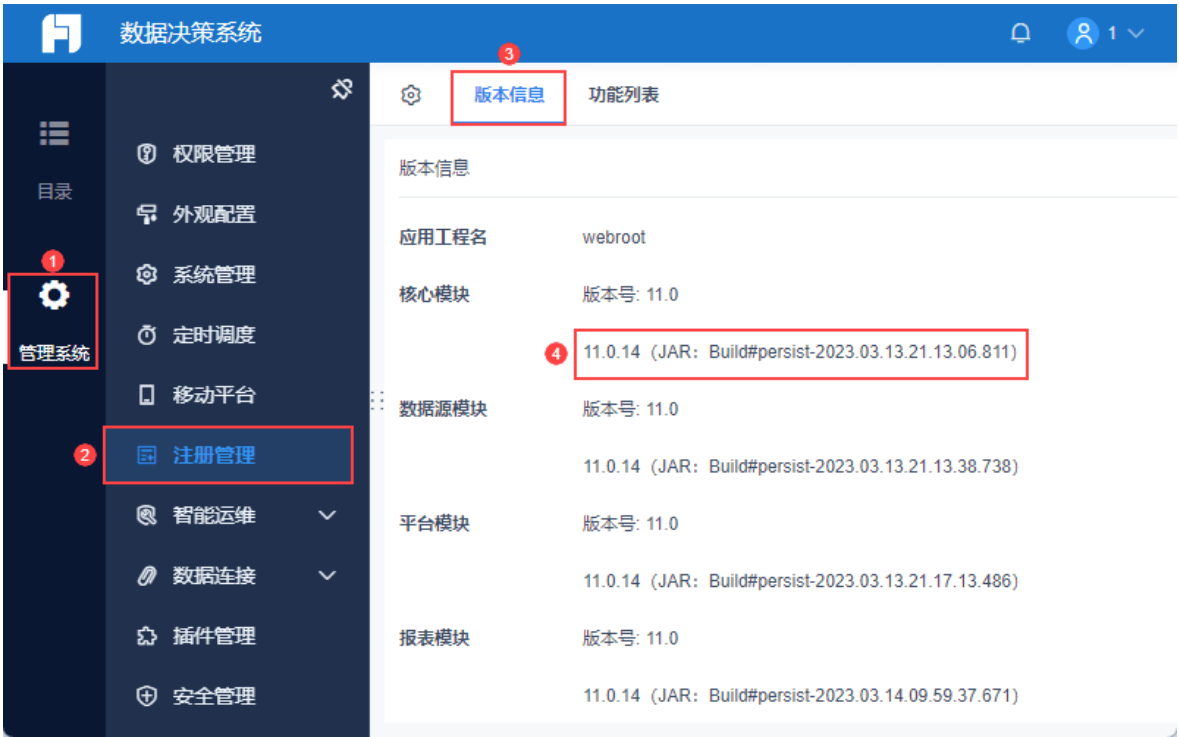
- 请确保应用部署在 Linux/Windows 系统的 Tomcat 中间件中。（其他环境部署的工程，尽管可以接入，但无法正常使用运维平台功能进行监控）
- 请确保应用版本在 FineReport11.0.6/FineBI5.1.22/FineDataLink4.0.26 及以上。

版本检查方法：

管理员登录 FineReport/FineBI/FineDataLink 应用，点击「管理系统>注册管理>版本信息」，即可查看工程各个模块的小版本号。

解决方案：

请参考 [FineReport 小版本升级指南](#)、[FineBI 小版本升级指南](#)、[FineDataLink 一键升级](#)，升级工程到最新版本。



## 2.2 检查网络是否连通

请确保运维平台和需要被管理的应用, 所在的服务器互相内网可联通。若网络不互通, 则无法对接/影响后续功能使用。

如需精确到指定端口互通, 请至少确保运维平台的工程端口和 FineReport/FineBI/FineDataLink 工程端口可以互相访问。

## 2.3 检查系统运维插件

请确保 FineReport/FineBI/FineDataLink 应用安装了「系统运维」插件。建议将插件升级到最新版本。

注: 如果应用配置了 CAS 单点登录, 安装「系统运维」插件后, 需要重启应用, 方可正常与 FineOps 运维平台对接。

## 2.4 准备运维接口密钥

### 要求说明:

项目管理的前提是, 运维平台获得应用的认可。因此首先要从应用中获取一个运维接口密钥。

### 获取方法:

管理员登录 FineReport/FineBI/FineDataLink 应用, 点击「管理系统>智能运维>健康巡检>运维工具」。设置一个运维接口密钥, 点击「保存」即可。

运维接口密钥请设置为大小写字母或数字, 长度在 8~14 之间。请记住此次设置的密

钥，下文需要用到。

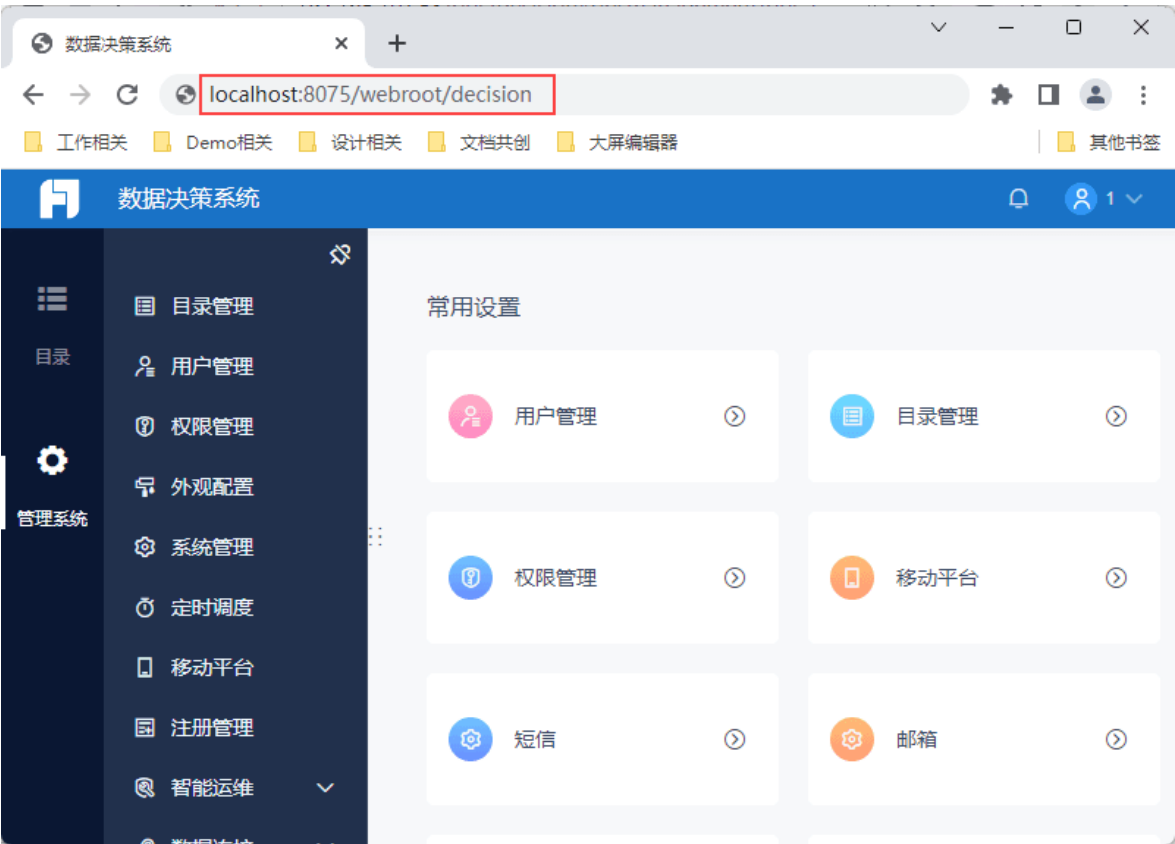
注：帆软应用只允许被一个运维平台接入。若已被接入，应用的运维密钥处显示「已接入」。



2.5 准备应用地址

应用地址，即为 FineReport/FineBI/FineDataLink 应用的访问地址，形如 `http://IP:端口号/工程名/decision`

注：请勿准备短域名地址作为应用地址，可能会造成接入失败。



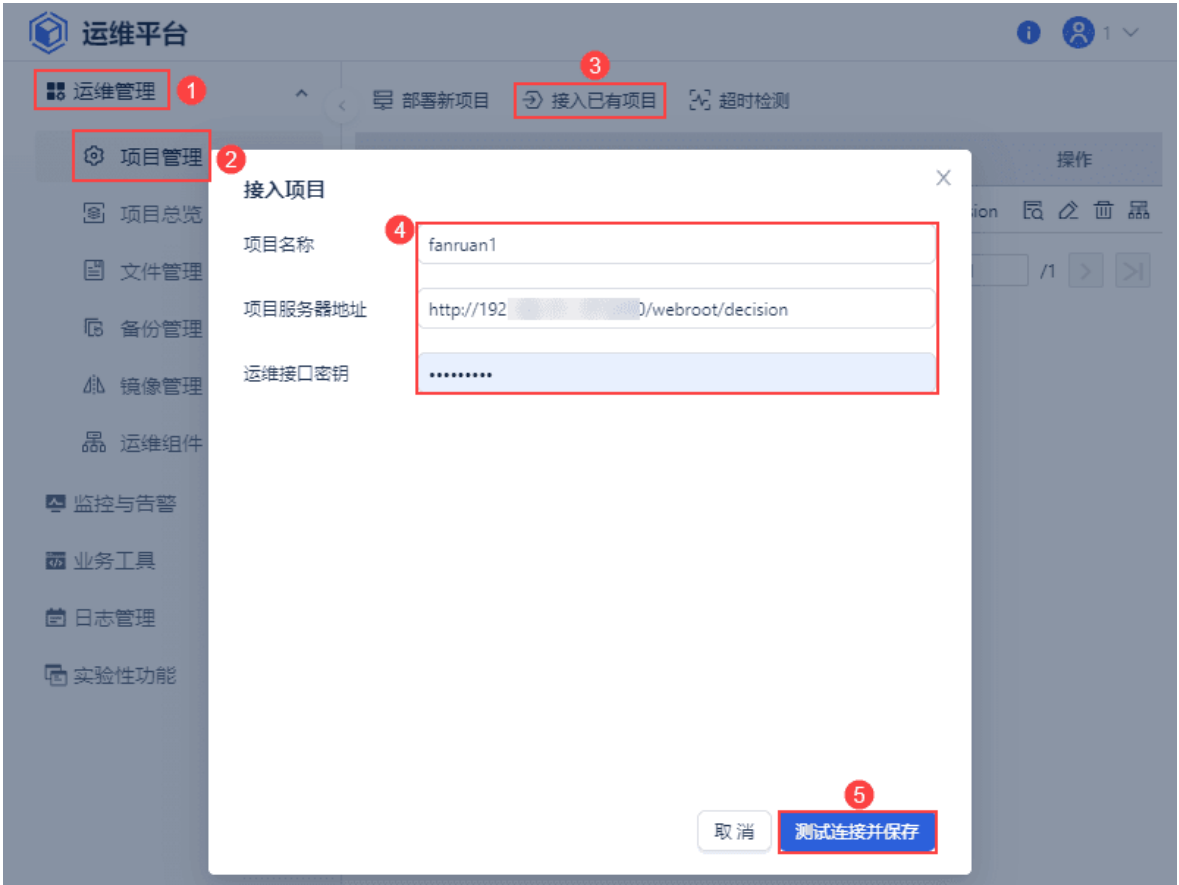
3. 项目设置

- 1) 管理员登录运维平台，点击「运维管理>项目管理」，点击「接入已有项目」按钮。
- 2) 设置相关项目信息，需要设置的应用信息如下表所示：



信息	说明
项目名称	自定义项目的名称
	项目名称不允许为空
	项目名称仅支持中文、英文大小写、数字
	项目名称长度最大为 12 位
项目服务器地址	FineReport、FineBI 等项目的地址(2.5 节准备的信息)
	项目服务器地址形如 http://IP:port/webroot/decision
	项目服务器地址不允许为空
	请勿准备短域名地址作为应用地址，可能会造成接入失败
运维接口密钥	运维接口密钥（2.4 节准备的信息）
	运维接口密钥不允许为空

3) 点击「测试连接并保存」，提示「接入项目成功」，进入「组件设置」环节。



#### 4. 组件设置（可选）

对接非容器化部署的 FineReport/FineBI/FineDataLink 应用时，可在运维平台所在服务器安装为该工程的外接数据库、负载均衡、状态服务器进行日志采集类型的 exporter

组件。

可安装的 exporter 组件包括：mysql\_exporter、redis\_exporter、nginx\_exporter

如不安装上述 exporter 组件，可直接点击「跳过」进入下一步，不影响运维平台与工程的对接，但后续运维平台也将无法监控组件状态并进行日志采集。

成功接入工程后，若又希望安装 exporter 组件，在项目管理界面点击对应项目的「编辑」按钮，重新设置即可。

4.1 安装前提

4.1.1 确认是否支持安装组件

如需安装此类组件，请确保 FineReport/FineBI 应用是否符合要求，如不符合要求，则无法安装对应的 exporter 组件。

工程类型	建议
单机-使用内置 finedb 数据库	无需安装任何 exporter 组件（没有相关被采集信息组件）  1) 无需安装 redis_exporter、nginx_exporter（集群独有）  2) 建议使用 MySQL 类型外接数据库，方可安装 mysql_exporter
单机-使用外接 finedb 数据库	3) 若外接数据库类型不满足要求，建议更换，详情请参见： <a href="#">配置 MySQL5 外接数据库</a>  1) 建议安装 mysql_exporter、redis_exporter、nginx_exporter  2) 建议使用以下类型的组件，方可安装以上三种组件
集群	<ul style="list-style-type: none"><li>• 外接数据库：MySQL</li><li>• 负载均衡：Nginx</li><li>• 状态服务器：Redis</li></ul> 3) 若组件类型不满足条件，可自行更换应用组件。

4.1.2 检查端口状态

为了 exporter 组件能正常进行日志采集和服务器信息采集，需要确保网络互通，请确认：

- MySQL、Redis 组件所在服务器和占用端口会自动读取，但用户需要自行提供 Nginx 组件的服务器地址、端口、SSL 信息

- 请确认 MySQL、Redis、Nginx 组件所在服务器和占用端口，可与运维平台内网联通。

#### 4.1.3 开启 Nginx 配置

Nginx 状态监控模块 `ngx_http_stub_status_module` 是统计 Nginx 服务所接收和处理的请求数量的模块。

若需要被管理的应用，并非使用 FineKey 容器化部署的应用，且需要安装 `nginx_exporter`，则需要额外安装该模块。

##### 1) 检查是否安装模块

- 进入 Nginx 的安装目录（本示例 `/usr/nginx` 是一个示例目录，请根据实际情况，确保使用正确的目录名）

```
cd /usr/nginx
```

- 检查 Nginx 启用了哪些模块

```
./nginx -V
```

- 结果

如果已启用，将在命令输出中看到 `http_stub_status_module` 的相关信息「`with-http_stub_status_module`」。

如果没有启用 `http_stub_status_module` 模块，则需要重新编译 nginx。

##### 2) 安装模块

- 进入 Nginx 的安装目录（本示例 `/usr/nginx` 是一个示例目录，请根据实际情况，确保使用正确的目录名）

```
cd /usr/nginx
```

- 在 `configure` 命令中，添加 `http_stub_status_module` 的支持

```
./configure --with-http_stub_status_module
```

注：请将上一步中「`nginx -V`」发现的已安装的模块都一一带上，例如 `./configure --with-http_stub_status_module --prefix=/usr/nginx --with-`

```

pcre=/usr/nginx/pcre-8.42 --with-zlib=/usr/nginx/zlib-1.2.11 --with-
http_ssl_module --with-openssl=/usr/nginx/openssl-1.1.1a --with-stream --
add-module=/usr/nginx/nginx_healthcheck_module-master

```

- 编译和安装 Nginx

```
make && make install
```

### 3) 启用模块

- 打开 nginx.conf 文件并修改

在默认情况下，nginx.conf 文件通常位于 Nginx 的安装目录下的 conf 目录中。

```
cd /usr/nginx/conf
```

- 在 nginx.conf 文件中，找到 server {} 块，并添加以下配置，以启用

ngx\_http\_stub\_status\_module 模块查询 Nginx 状态。

```

location /basic_status {
    stub_status on;
    #allow 127.0.0.1;    #限制访问 ip,可自行修改 127.0.0.1 为所需开放的 ip
    #deny all;         #deny all other hosts
}

```

注意事项:

- Location 地址请严格命名为 basic\_status。
- allow 127.0.0.1 和 deny all 表示仅允许本地访问。

若需允许 Nginx Exporter 访问，则可将这两行代码注释，或者将 127.0.0.1 设置为 Nginx Exporter 的 IP 地址。

### 4) 重启 Nginx

- 进入 Nginx 的安装目录（本示例/usr/nginx 是一个示例目录，请根据实际情况，确保使用正确的目录名）

```
cd /usr/nginx
```

- 检查 Nginx 配置文件的语法和结构是否正确

./nginx -t

- 关闭 Nginx 服务器

./nginx -s stop

- 重新加载 Nginx 的配置文件并启动服务器

nginx

5) 验证模块是否启用

验证 ngx\_http\_stub\_status\_module 模块是否已成功启动

使用语句查询，若出现提示「server accepts handled requests」则表示 ngx\_http\_stub\_status\_module 模块已成功启动。

curl http://127.0.0.1/basic\_status

4.2 操作步骤

如需安装 exporter 组件，请确保已执行并准备好 4.1 节相关 exporter 组件信息。

在「组件设置」步骤，会自动设置 MySQL 和 Redis 组件信息，需手动填写 Nginx 组件信息（4.1.2 节）。

注：为了方便内网用户对接，V1.5.6 及之后版本，MySQL 和 Redis 组件的地址和端口，也允许被修改。

点击「测试连接并保存」，连接成功后提示「已保存，连接成功」，进入下一步。

接入项目

✓ 项目设置

2 组件设置

提示：非运维平台部署的项目可选填组件、服务器设置

Redis组件

Redis组件名称redis

Redis地址192.168.5.15

端口8379

▼ Nginx组件 注：需要Nginx开启ngx\_http\_stub\_status\_module 查看详情

SSL☐

Nginx组件名称nginx

Nginx地址192.168.101.69

端口8088

取消

测试连接并保存

5. 服务器信息设置（可选）

对接非容器化部署的 FineReport/FineBI 应用时, 可在应用组件所在服务器安装为该工程进行服务器信息采集的组件。

可安装的组件包括: node\_exporter 组件, 服务器物理节点的 exporter, 收集物理服务器的各种指标信息, 每个物理节点安装一个 node\_exporter

如不安装上述组件, 可直接点击「跳过并保存」进入下一步, 不影响运维平台与工程的对接, 但后续运维平台也将无法监控服务器状态并进行日志采集。

成功接入工程后, 若又希望安装 node\_exporter 组件, 在项目管理界面点击对应项目的「编辑」按钮, 重新设置即可。

## 5.1 安装前提

### 5.1.1 准备应用节点信息

为了有权限安装组件, 需要准备:

- FineReport/FineBI 应用所在服务器的主机 IP、端口、具有 root 权限的用户密码
- 集群需要准备好所有节点所在服务器的主机 IP、端口、具有 root 权限的用户密码
- 若使用私钥, 请在本地准备好私钥文件, 并确保用户有权限上传密钥文件到运维平台挂载目录的 /root/data/fanruanxxxxx/ops/resources/certificates 文件夹中

注 1: 挂载目录寻找方式

- 1) 先使用「docker ps -a」命令, 找到 NAMES 中包含 ops 的 CONTAINER ID (容器 ID)。
- 2) 再使用「docker inspect 容器 ID」命令, 在返回的 HostConfig 中, 即可得到上面所指挂载目录的位置。

注 2: V1.11.0 及之后版本, 支持使用非 root 用户进行安装, V1.10.0 及之前版本, 仅支持使用 root 用户进行安装。

请确保准备的用户具有 **root 权限**, 有些指标数据需要访问系统资源或敏感信息, 这些操作通常需要特权 (root) 访问权限。

例如, 获取网络接口统计信息、文件系统使用情况等。这些指标数据只有在使用 root 权限运行 node\_exporter 时才能够收集和暴露。

### 5.1.2 检查端口状态

为了确保组件可以正常启动并与运维平台通信，请确保：

- node\_exporter 默认占用端口 9100，请确保 9100-9104 中至少有一个端口可用，安装时会从 9100 开始尝试五次，五次均失败便不安装该组件。
- 请确保 node\_exporter 组件占用端口，可与运维平台所在服务器内网联通。

### 5.1.3 准备日志存放目录

为了有地方存放 exporter 日志，需要准备：

- 在 FineReport/FineBI 应用所在服务器，新建一个目录，该目录用于存放 exporter 日志。
- 集群需要在所有节点所在服务器都准备好一个目录。
- 准备的用户具有该目录的读写权限。

## 5.2 操作步骤

如需安装组件，请确保已执行并准备好 5.1 节相关信息。

请将 5.1.1 节准备的 FRBI 应用服务器信息和 5.1.3 节准备的安装路径（即日志存放目录）填入。

注 1：若工程有多个节点，请点击「新增服务器」新增节点进行配置。

注 2：此处填写的安装路径，必须为绝对路径。

接入项目

✓ 项目设置

✓ 组件设置

3 服务器设置

新增服务器

服务器1

主机

192.168.101.69

端口

22

用户名

root

验证方法

密码

密码

安装路径

/root

注：服务器信息配置用于安装日志采集和服务器信息采集组件，不希望安装可以选择跳过，此时将无法监控服务器状态并进行项目日志采集

取消

上一步

跳过并保存

测试连接并保存

6. 效果预览

配置完成后，点击「测试连接并保存」，连接成功后提示「已保存，连接成功」。

在项目管理列表中出现新接入的项目，即代表接入成功，如下图所示：

运维平台

运维管理

部署新项目接入已有项目超时检测

项目管理

项目总览

文件管理

备份管理

名称	类型	存活节点数	部署方式	状态	访问地址	操作
fr11jiqun	集群	2	非运维平台部署	可用	http://[redacted]webroot/decision	民 2 前 显

1

/1



## 4.3 注册新项目

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.15.0	-

#### 1.2 功能简介

使用运维平台部署的运维项目，推荐使用容器私有云认证方式，对工程进行注册授权。

运维平台提供前端界面化快捷注册途径，帮助用户快速进行容器私有云注册。

#### 1.3 注意事项

1) 本文仅适用于使用运维平台部署的运维项目，非运维平台部署的帆软应用，注册方法请参见：[工程注册方案选择](#)

2) 授权应用不可关闭超过 24 小时。

- 若关闭时长超过 24 小时，重启后需要重新认证。无需重新上传镜像文件，只需重新扫描二维码填写验证码。
- 若关闭时长小于 24 小时，帆软应用显示注册失败，在 2 小时内将自动重新连接上授权应用，注册不失效。

3) 帆软应用会每 2 小时向授权应用发送一次请求，授权应用收到请求后以后会反馈给帆软应用。

若帆软应用连续 12 次（即 24 小时内）收不到反馈，或授权应用连续 24 小时没有收到请求，则帆软应用注册失败，需重新进行注册操作。

### 2. 注册步骤

#### 2.1 获取授权镜像

请使用合同中指定的邮箱发送注册信息给 FineReport 商务 business@fanruan.com，邮件格式如下：

商务收到邮件，核实无误后会为用户生成认证服务器的镜像文件 fanruan\_license\_server.tar，以邮件方式发送给用户。

注：若有任何特殊情况，请在邮件中注明。

清单	要求	注意事项
优先联系	请提前联系帆软销售，确认相关事项并签署合同	-
发件人	请使用合同中指定的邮箱	如发送注册信息的邮箱，非合同中指定的邮箱，则不予注册
收件人	business@fanruan.com	-
标题	XXX 公司注册 FineReport 正式授权文件	如发送注册信息时，未注明公司名称，则不予注册
	公司名称：XXX 公司	
基础信息	项目名称：XXX 项目	-
	合同签定日期：XXXX 年 XX 月 XX 日	
注册方式	容器私有云认证	-
正文	最大可注册量 集群工程节点个数	若不提供该选项，则默认为 1
		单机情况下无需提供该项
	请精确到工程小版本和 JAR 包信息	
工程版本	在「管理系统-注册管理-版本信息」中可查看	-

2.2 准备注册端口

容器私有云认证，需要在运维项目中上传启动授权镜像。

- 单节点运维项目，授权组件可部署到主应用服务器中。
- 多节点集群项目，授权组件需部署到组件服务器中。

请在对应服务器中，准备一个空闲端口，作为授权组件占用端口（默认为 18081）。

请确认准备的端口，和工程所在服务器内网连通。

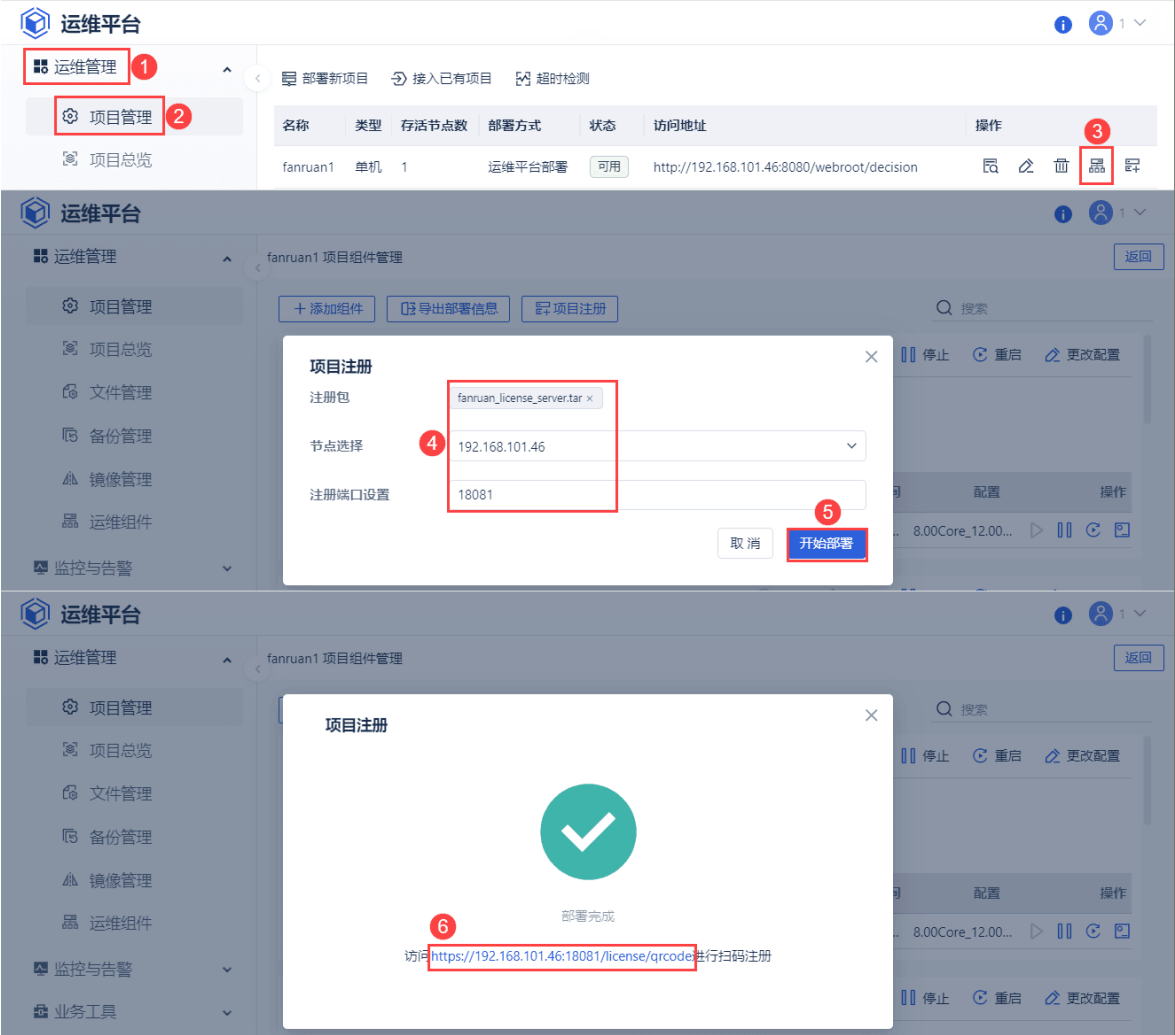
端口占用和防火墙开放操作方法可参考：[端口占用检查与防火墙配置](#)。

2.3 部署授权应用

- 1) 管理员登录运维平台，点击「运维管理>项目管理」，点击目标项目对应的「组件管理」。
- 2) 点击「项目注册」：上传 2.1 节获取的注册包镜像文件；选择 2.2 节选定的节点和端口。
- 3) 点击「开始部署」，部署完成后，提示扫码认证地址。

请记住该地址，形如 `https://IP:port/license/qrcode`（IP 和端口即为 2.2 节选定的服务器

IP 和端口)。



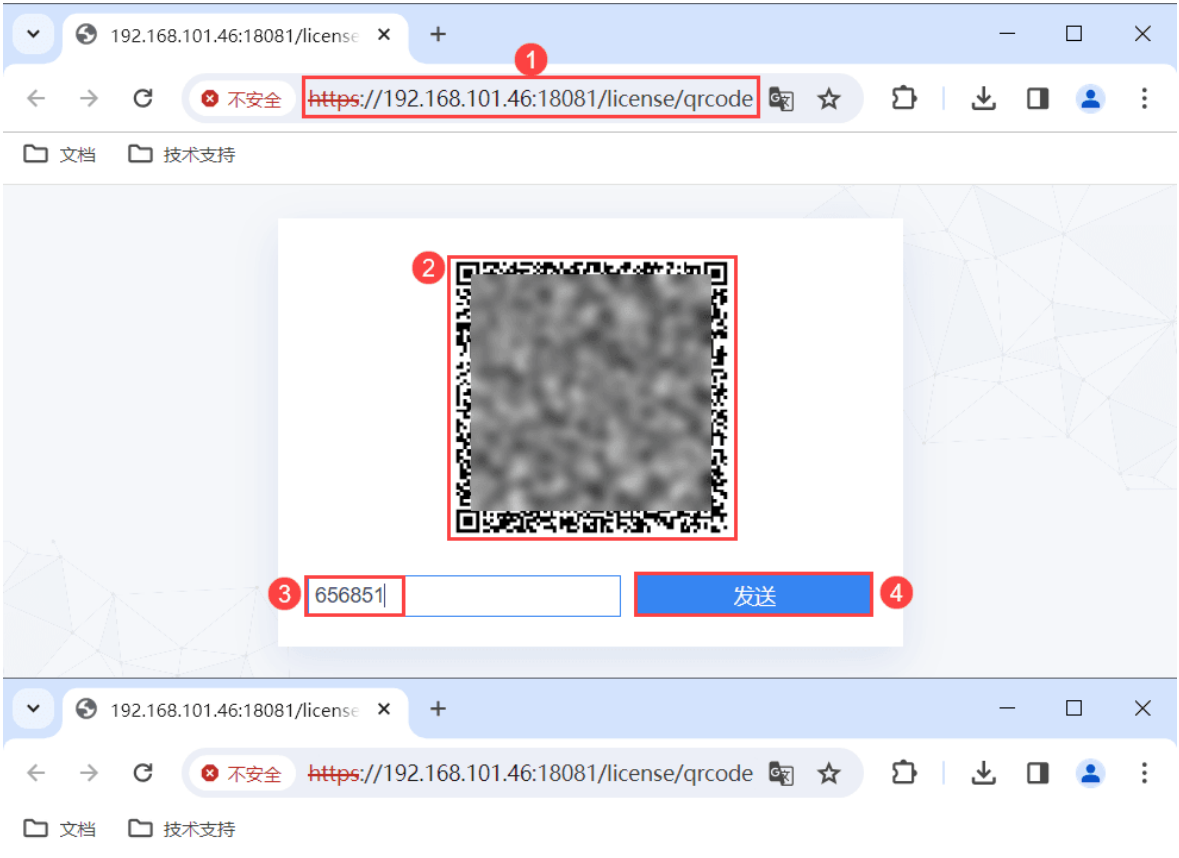
## 2.4 扫码提交认证

- 1) 访问上一节获取的扫码认证地址。
- 2) 使用手机浏览器/微信扫描生成的二维码，会返回一个 6 位的验证码，将该验证码填写到系统中，点击「提交认证」。
- 3) 认证成功后，提示「验证成功」。

注 1：二维码有效时间为 5 分钟，请于规定时间内完成注册流程。超过 5 分钟后，二维码会提示需要手动刷新。

注 2：为了防止验证码暴力破解和接口高频访问，验证码输入错误次数若超过 5 次，则锁定 15 分钟。

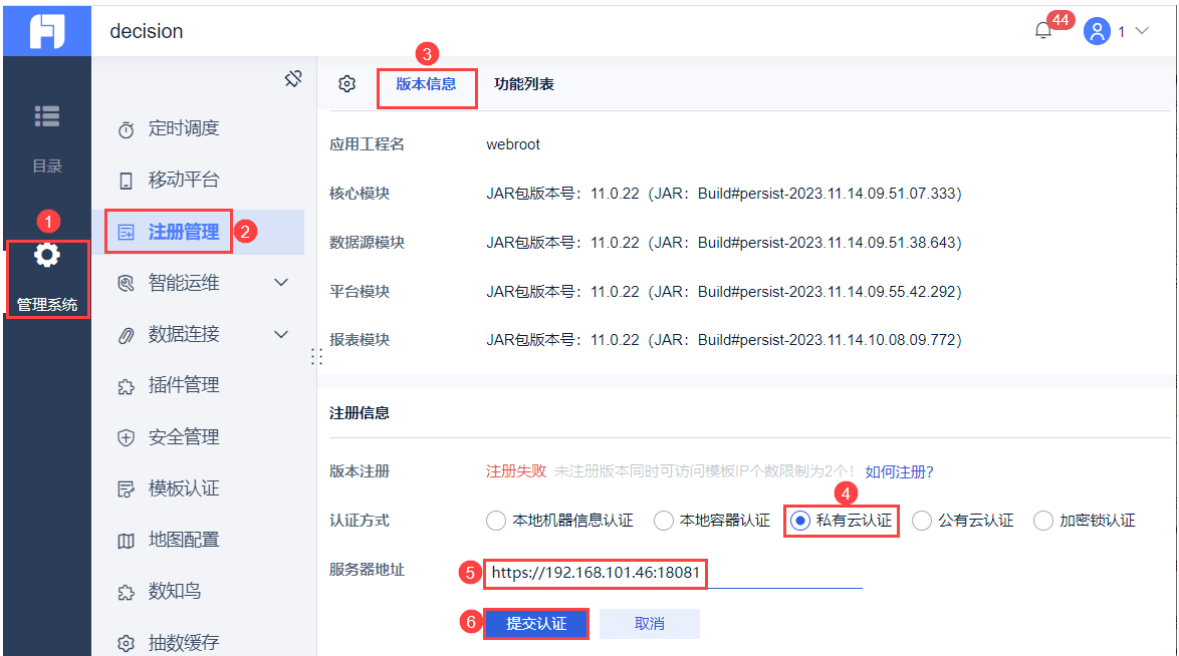
注 3：手机扫码返回验证码之后，帆软应用最好立刻注册，未注册成功前不要重启帆软应用。



**验证成功**

## 2.5 工程进行注册

- 1) 管理员登录帆软应用，点击「管理系统>注册管理>立即注册」。
- 2) 认证方式选择「私有云认证」。
- 3) 服务器地址填写 2.3 节授权应用的 IP 和端口，例如 https://192.168.101.46:18081/，点击「提交认证」。



4) 认证成功后，如下图所示：

注：若认证失败，检查防火墙设置，开放对应端口或者关闭防火墙。



### 3. 认证迁移

进行了容器私有云认证的工程，如需迁移授权服务器，请按照以下步骤进行。

#### 3.1 销毁原工程授权

请使用合同中指定的邮箱发送邮件给 FineReport 商务 business@fanruan.com。

请在邮箱正文描述清楚待迁移的授权服务器地址，例如 2.5 节填写的 https://192.168.101.46:18081/

商务收到邮件，核实无误后会为用户销毁对应授权，并以邮件方式通知用户。

注：在得到商务明确答复前，请勿重启待迁移的授权应用。

#### 3.2 拷贝 privateConfig 文件

- 1) 在「项目管理-组件管理」中，关闭待迁移的授权组件。
- 2) 在该容器的/app/config 路径下获取自动生成的 privateConfig 文件。

#### 3.3 获取新授权文件

请使用合同中指定的邮箱发送邮件给 FineReport 商务 business@fanruan.com。

邮件格式请参考本文 2.1 节，额外附上上一步获取的 privateConfig 文件。

商务收到邮件，核实无误后会为用户生成认证服务器的镜像文件 fanruan\_license\_server.tar，以邮件方式发送给用户。

用户再按照第二章的步骤启动新授权应用进行注册即可。

## 4.4 更新项目

## 4.4.1 外网升级运维项目

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.5.5	-
V1.12.0	降低运维项目升级前备份要求，存在 24 小时内备份即可升级

#### 1.2 应用场景

为满足客户的新需求以及完善之前版本某些功能的不足之处，FineReport/FineBI 在不断地更新迭代。

小明部署好容器化 FR/BI 工程后，需要定期对工程进行更新升级，但是他发现传统的升级方式存在着一些问题。

- 1) FineReport 内网/FineBI 的小版本升级需要**手动**更换 JAR 包，操作门槛高。
- 2) 由于部分版本的兼容问题，用户需要升级**指定版本**而非最新版本，只能手动更换 JAR 包，操作门槛高。
- 3) FineReport 外网升级，需要**手动重启**工程/集群所有节点，操作麻烦。
- 4) 升级失败后工程**无法自动回退**，存在风险。

#### 1.3 功能简介

FineOps 运维平台提供面向 FineReport/FineBI 项目组件的「更新」功能。

- 1) 支持升级最新/指定版本。
- 2) 通过推送镜像方式，支持内网安全更新。
- 3) 任何更新失败均可自动回退。

#### 1.4 适用范围

外网升级最新版本：支持外网联通的运维平台，通过在线获取最新镜像，即可升级指定组件。

外网升级指定版本：支持外网联通的运维平台，通过联系技术支持获取指定版本镜像号，在线获取最新镜像，即可升级指定组件。

注：内网升级项目，请参考 [内网升级项目](#)。



## 2. 外网升级最新版本项目

### 2.1 使用前提

#### 2.1.1 容器化部署

使用运维平台更新帆软项目，仅面向**容器化部署**的项目，非容器化部署工程不支持本文相关操作。

#### 2.1.2 配置镜像仓库

如需使用运维平台更新项目组件，需要有新版本的组件镜像包用于替换安装。因此要支持从云端拉取镜像。

请查看镜像仓库地址，并确保地址可连通外网云端仓库。

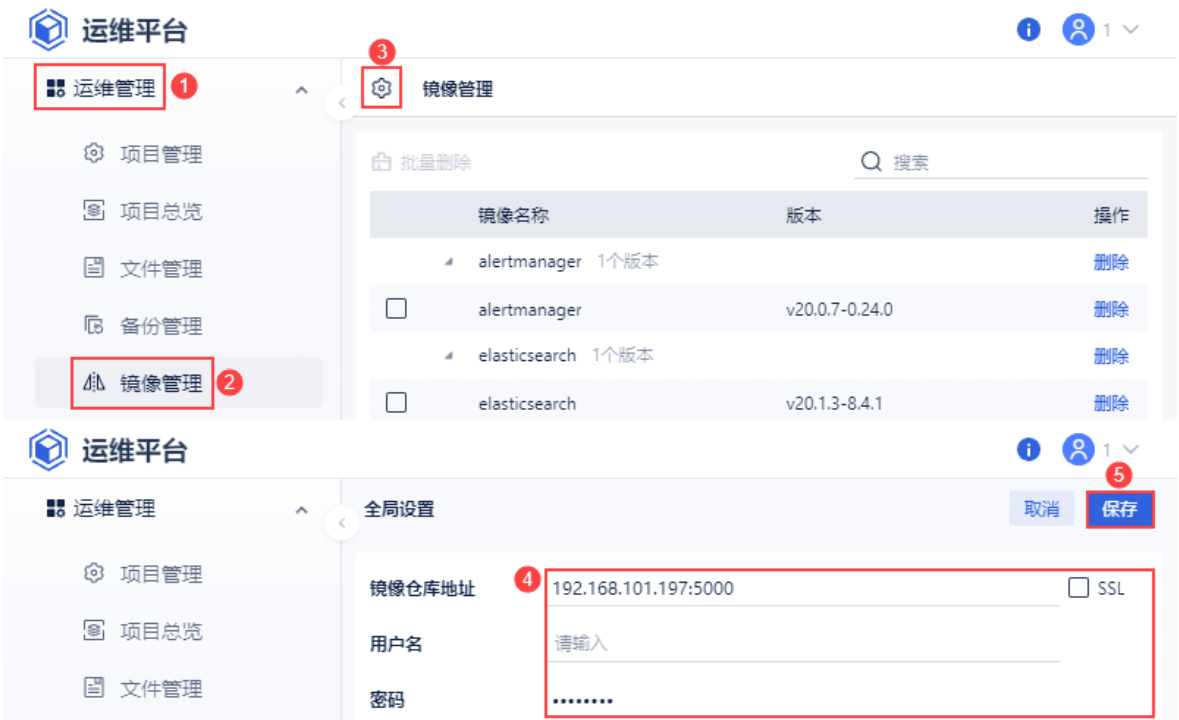
- 查看镜像仓库地址：管理员登录运维平台， 点击「运维管理>镜像管理>全局配置」，即可查看镜像仓库地址。
- 确保镜像仓库所在服务器支持访问以下外网地址：

域名：registry.cn-hangzhou.aliyuncs.com， 端口：443

域名：fineops-registry.cn-hangzhou.cr.aliyuncs.com， 端口：443

即可在从云端拉取最新项目组件镜像到镜像仓库中。



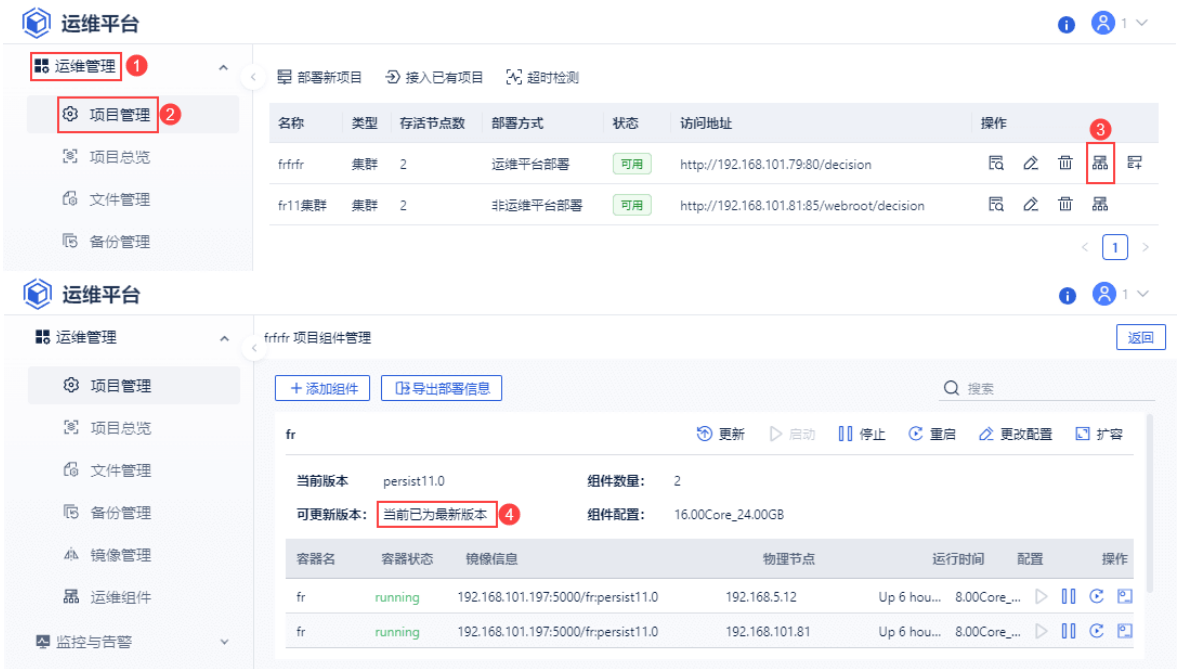


2.1.3 检查工程版本

管理员登录运维平台, 点击「运维管理>项目管理」, 点击目标项目对应的「组件管理」。

找到 FR/BI 组件，检查是否存在可更新版本。

若显示「当前已为最新版本」，则说明暂无更高版本发布，无需执行更新操作。



2.2 备份工程

建议在更新组件前，对项目进行整体备份，防止更新过程出现不可控意外。

管理员登录运维平台, 点击「运维管理>备份管理」, 选择目标项目，建议勾选「备份 jar 包」, 点击「开始备份」, 运维平台即会对该项目进行整体备份。

注：如不存在



2.3 更新版本

1) 管理员登录运维平台，点击「运维管理>项目管理」，点击目标项目对应的「组件管理」。



2) 找到 fr/bi6/fdl 组件，点击「更新」，注意是上方的整体更新，而不是单个工程节点的更新。

3) 升级最新版本，无需输入更新版本号，直接点击「开始更新>确定」，即可进行工程更新。

- 更新过程将重启组件，可能影响项目正常运行，请勿在使用期间操作。
- 如存在 24 小时内的备份，则无需再次备份，直接更新。
- 如不存在 24 小时内的备份，且运维平台磁盘空间充足，则先进行备份，再进行更新。
- 如不存在 24 小时内的备份，且运维平台磁盘空间不足，则更新终止，请对磁盘进行清理/扩容后再进行更新。



### 3. 外网升级指定版本

#### 3.1 获取镜像版本号

付费用户请联系帆软技术支持，告知所需产品和小版本号，或大概的 jar 包日期。  
技术支持会告知您具体的镜像版本号。

技术支持联系方式：前往「[服务](#)」，选择「在线支持」、电话「400-811-8890」。

#### 3.2 使用前提

##### 3.2.1 容器化部署

使用运维平台更新帆软项目，仅面向**容器化部署**的项目，非容器化部署工程不支持本文相关操作。

##### 3.2.2 配置镜像仓库

如需使用运维平台更新项目组件，需要有新版本的组件镜像包用于替换安装。因此要支持从云端拉取镜像。

请查看镜像仓库地址，并确保地址可连通外网云端仓库。

- 查看镜像仓库地址：管理员登录运维平台，点击「运维管理>镜像管理>全局配置」，即可查看镜像仓库地址。
- 确保镜像仓库所在服务器支持访问以下外网地址：

域名：registry.cn-hangzhou.aliyuncs.com，端口：443

域名：fineops-registry.cn-hangzhou.cr.aliyuncs.com，端口：443

即可在从云端拉取项目组件镜像到镜像仓库中。



### 3.3 备份工程

建议在更新组件前，对项目进行整体备份，防止更新过程出现不可控意外。

管理员登录运维平台，点击「运维管理>备份管理」，选择目标项目，建议勾选「备份jar包」，点击「开始备份」，运维平台即会对该项目进行整体备份。



### 3.4 更新版本

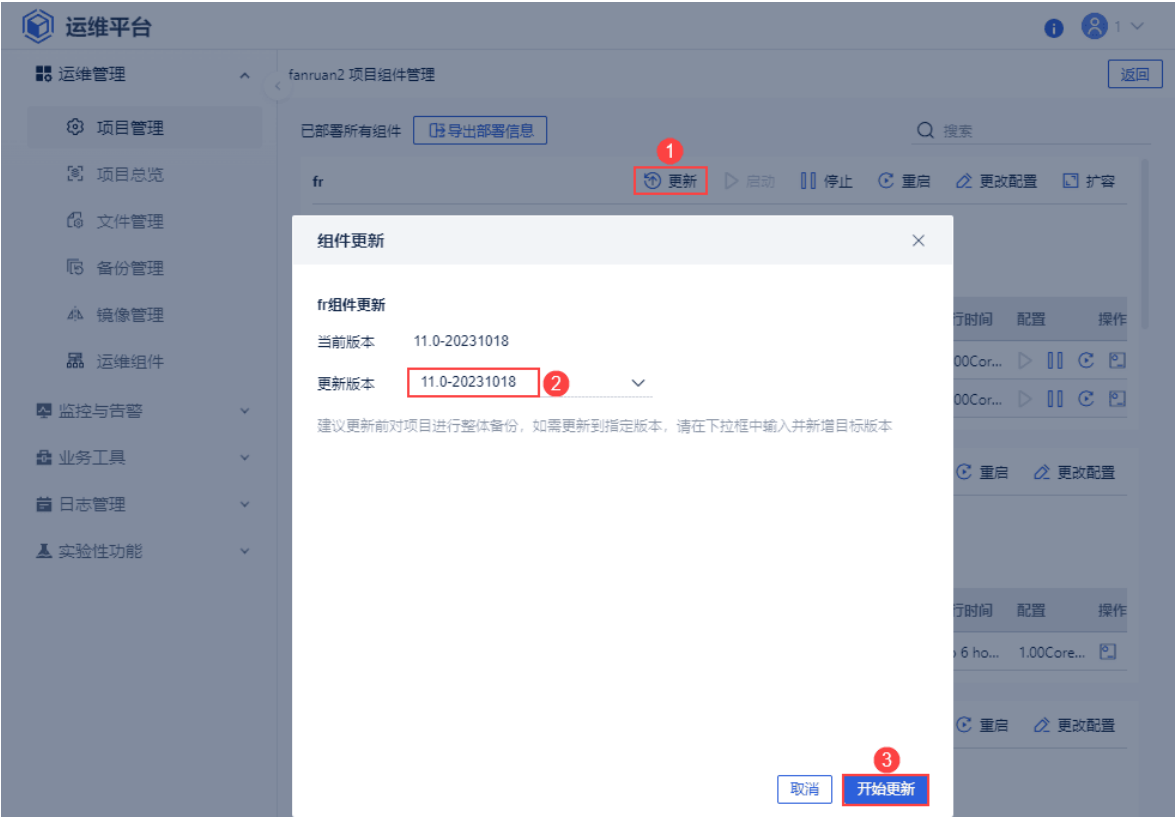
1) 管理员登录运维平台，点击「运维管理>项目管理」，点击目标项目对应的「组件管理」。



2) 找到 fr/bi6/fdl 组件，点击「更新」，注意是上方的整体更新，而不是单个工程节点的更新。

3) 在「更新版本」处手动输入 3.1 节获取的镜像版本号，点击「开始更新>确定」，即可进行工程更新。

- 更新过程将重启组件，可能影响项目正常运行，请勿在使用期间操作。
- 如存在 24 小时内的备份，则无需再次备份，直接更新。
- 如不存在 24 小时内的备份，且运维平台磁盘空间充足，则先进行备份，再进行更新。
- 如不存在 24 小时内的备份，且运维平台磁盘空间不足，则更新终止，请对磁盘进行清理/扩容后再进行更新。



## 4.4.2 内网升级运维项目

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.5.5	-
V1.12.0	降低运维项目升级前备份要求，存在 24 小时内备份即可升级

#### 1.2 应用场景

为满足客户的新需求以及完善之前版本某些功能的不足之处，FineReport/FineBI 在不断地更新迭代。

小明部署好容器化 FR/BI 工程后，需要定期对工程进行更新升级，但是他发现传统的升级方式存在着一些问题。

- 1) FineReport 内网/FineBI 的小版本升级需要**手动**更换 JAR 包，操作门槛高。
- 2) 由于部分版本的兼容问题，用户需要升级**指定版本**而非最新版本，只能手动更换 JAR 包，操作门槛高。
- 3) FineReport 外网升级，需要**手动重启**工程/集群所有节点，操作麻烦。
- 4) 升级失败后工程**无法自动回退**，存在风险。

#### 1.3 功能简介

FineOps 运维平台提供面向组件的「更新」功能。

- 1) 支持升级最新/指定版本。
- 2) 通过推送镜像方式，支持内网安全更新。
- 3) 任何更新失败均可自动回退。

#### 1.4 适用范围

内网升级版本：仅内网使用的运维平台，通过 finekey 推送镜像，将项目组件升级到指定/最新版本时的解决方案。

注：外网升级最新版本，请参考：[外网升级最新版本项目](#)



2. 使用前提

2.1 容器化部署

使用运维平台更新帆软项目，仅面向**容器化部署**的项目，非容器化部署工程不支持本文相关操作。

2.2 检查工程版本

请先检查当前工程版本。

管理员登录 FR/BI 工程，点击「管理系统>注册管理>版本信息」，查看当前版本号。

检查 [FineReport 更新日志](#)、[FineBI 更新日志](#) 即可查看官方发布的最新版本号，确认自己需要升级到的版本。



3. 推送镜像

3.1 保存原 yaml 文件

从原容器化部署工具 finekey/conf 目录下获取 finekey.yaml 文件，下文需要用到。

3.2 获取指定版本 FineKey

1) 最新版本

点击可跳转至下载页下载安装包：[FineOps 运维平台下载](#)

请下载全量版离线安装包。

2) 指定版本

付费用户联系技术支持，索要指定版本的镜像文件。请提供所需的 FineReport/FineBI 小版本号。

技术支持联系方式：在线支持：前往「[服务](#)」，选择「在线支持」；电话：400-811-8890。

技术支持会提供该版本对应的 FineKey 容器化部署工具，工具中有相关镜像包。

3.3 上传 FineKey 安装包

获得工具包后，需要将它上传到 Linux，上传文件的方法请自行查阅相关资料。

本文示例是将工具包上传到了 CentOS 7.6.1810 的 /home 路径下，如下图所示：

```
[root@localhost home]# cd /home
[root@localhost home]# ls
finekey-operation.tar.gz
[root@localhost home]# ls
```

3.4 解压 FineKey 安装包

1) 先使用 cd 命令进入工具包所在文件夹下。

2) 键入如下命令进行解压：

```
tar zxvf finekey-operation.tar.gz
```

```
[root@localhost home]# cd /home
[root@localhost home]# ls
finekey-operation.tar.gz
[root@localhost home]# cd /home
[root@localhost home]# tar zxvf finekey-operation.tar.gz
finekey/
finekey/bin/
finekey/bin/finekey
finekey/bin/fine_net_test
finekey/conf/
finekey/conf/finekey.yaml
finekey/conf/finekey.push.yaml
finekey/resources/
finekey/resources/images/
```

解压后的文件如下表所示：

文件夹/文件		文件说明
finekey/bin		运行 finekey 的可执行文件
finekey/conf	finekey.yaml	新部署运维平台配置文件
	finekey.push.yaml	拉取镜像配置文件
finekey/resources		资源文件，镜像包放在其中 images 文件夹中

3.5 配置 yaml 文件

3.5.1 替换 finekey.yaml 文件

取出从原 finekey 部署工具 finekey/conf 目录下获取 finekey.yaml 文件，将文件中的配



置值，一项一项的复制替换到新 FineKey 部署工具 finekey/conf 目录下的 finekey.yaml 文件。

注：请勿直接替换 yaml 文件，或一键复制粘贴文件内容，因为新旧版本中，yaml 文件设置可能存在微小差异，建议逐项拷贝配置值即可。

3.5.2 修改 finekey.push.yaml 文件

根据 finekey.yaml 中的信息，修改新 finekey 部署工具 finekey/conf 目录下的 finekey.push.yaml 文件。

```
node:
  ip: localhost                #机器 ip
  port: 22                    #机器 ssh 端口
  user:                       #节点 ssh 用户
  password:                   #节点 ssh 密码
repo:
  url:                        #已有 repo 的 url
  username:                   #已有 repo 的用户名
  password:                   #已有 repo 的密码
  ssl: true                   #已有 repo 是否安全仓库
```

3.6 执行推送

- 1) 先使用 cd 命令进入工具包 bin 目录。
- 2) 使用语句启动升级，此时便会自动向仓库中推送镜像。  
./finekey push --config ../conf/finekey.push.yaml

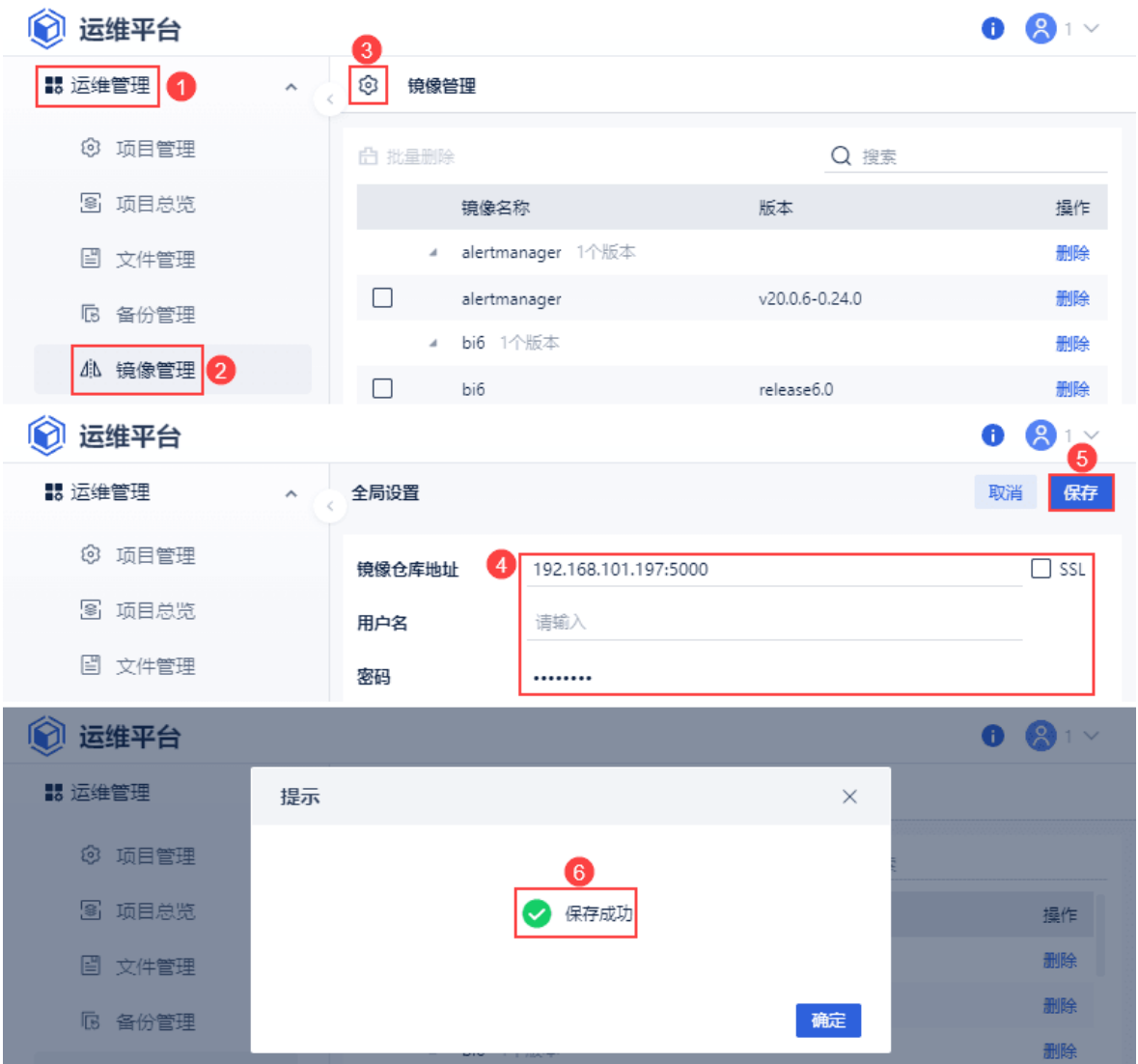
4. 检查镜像仓库

如需使用运维平台更新项目组件，需要有新版本的组件镜像包用于替换安装。

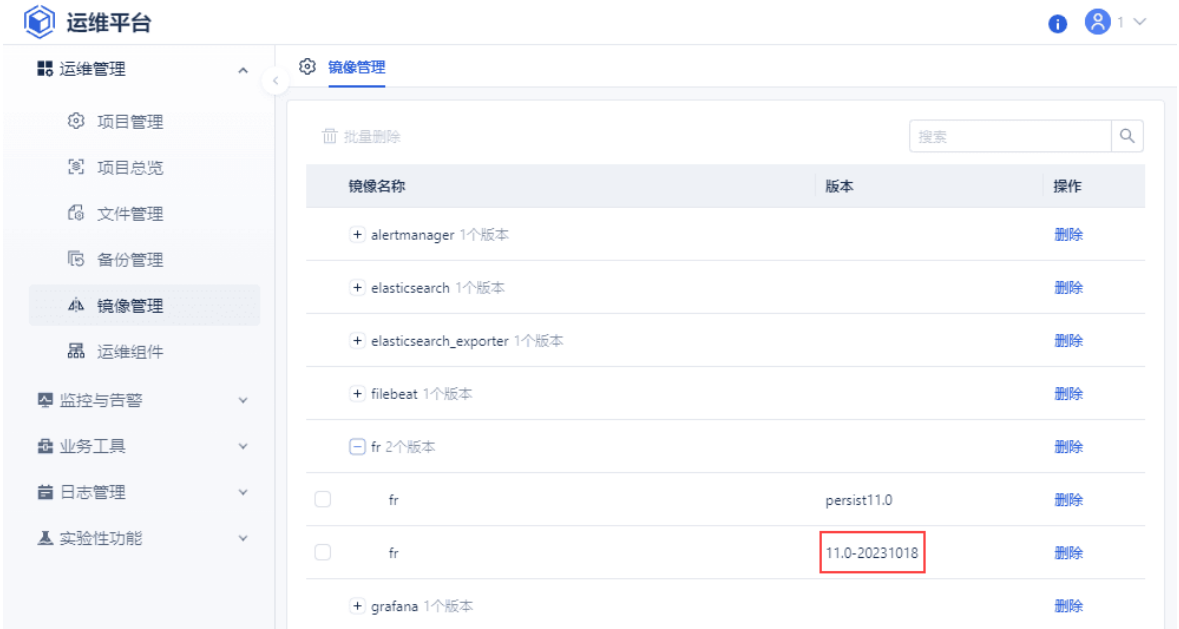
因此需要先为运维平台配置一个镜像仓库，用于推送/存放镜像包。

- 1) 使用自带镜像仓库：在部署运维平台时，会默认安装一个仓库组件 registry。镜像仓库地址为运维平台 IP:5000，无需填写用户名、密码、SSL，点击「保存」即可。
- 2) 使用用户自有仓库：用户需要自行提供仓库地址、用户名、密码、SSL，点击「保存」即可。

该镜像仓库是在上节 finekey.yaml 文件中配置了的，请再在运维平台中确认下已连接。



当仓库配置完成后，会自动显示推送到仓库的新镜像。记下需要更新到的版本号，下文需要用到。



## 5. 备份工程

建议在更新组件前，对项目进行整体备份，防止更新过程出现不可控意外。

管理员登录运维平台，点击「运维管理>备份管理」，选择目标项目，建议勾选「备份镜像」，点击「开始备份」，运维平台即会对该项目进行整体备份。



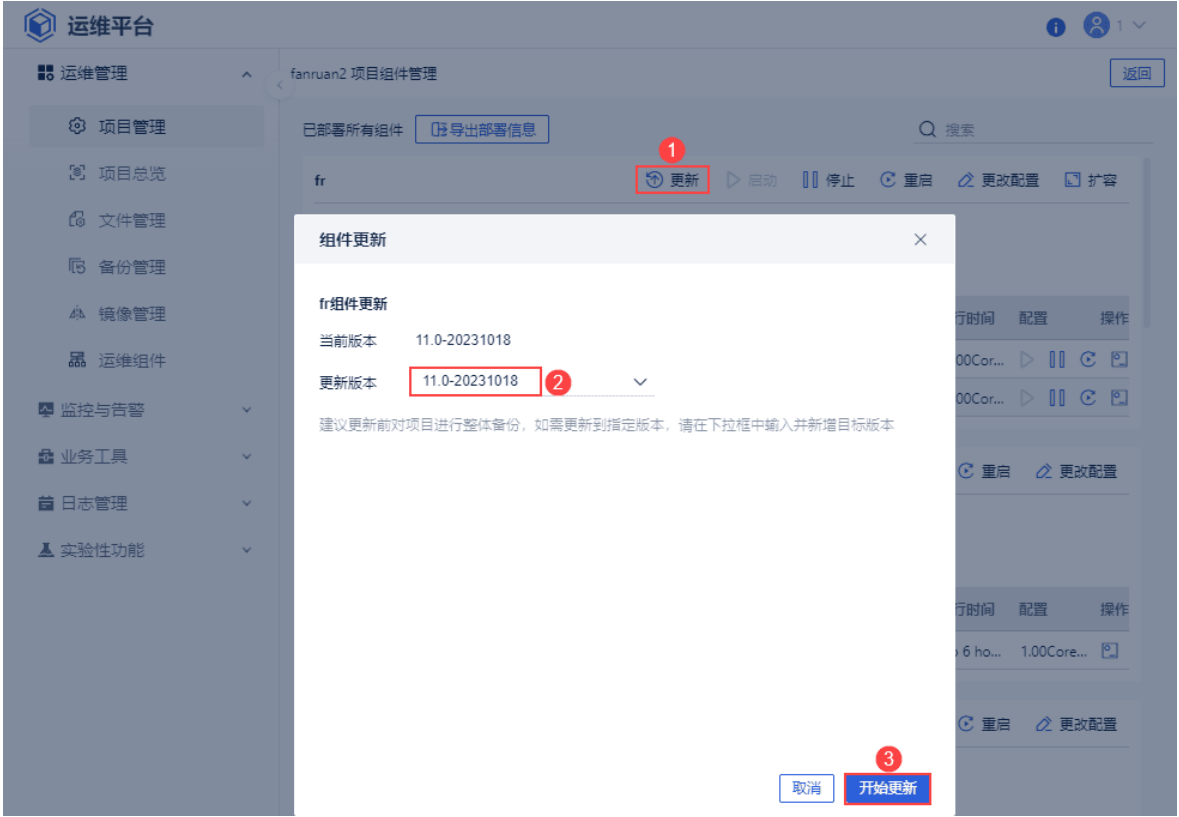
## 6. 更新版本

1) 管理员登录运维平台，点击「运维管理>项目管理」，点击目标项目对应的「组件管理」。



2) 找到待更新组件，点击「更新」，输入仓库中的版本号，点击「开始更新」，即可进行工程更新。

- 如存在 24 小时内的备份，则无需再次备份，直接更新。
- 如不存在 24 小时内的备份，且运维平台磁盘空间充足，则先进行备份，再进行更新。
- 如不存在 24 小时内的备份，且运维平台磁盘空间不足，则更新终止，请对磁盘进行清理/扩容后再进行更新。



## 4.5 迁移项目

## 4.5.1 非容器化单机迁移容器化单机

### 1. 概述

#### 1.1 版本

工程版本	功能变更
FineReport11.0	-
FineBI6.0	

#### 1.2 应用场景

相比于普通单机工程，容器化部署的工程，升级和维护成本较低。

本文方案，帮助用户将原有非容器化部署的单机工程，迁移为容器化部署的单机工程。

注：全文以 FineBI 作为演示。FineReport 迁移步骤基本一致，仅需要两个点：

- 1) 拷贝文件时，FineReport 和 FineBI 需要拷贝的文件不完全相同，注意文档中的表格即可。
- 2) 使用运维平台启停容器时，FineReport 关闭的是 fr 组件，FineBI 关闭的是 bi6 组件。

#### 1.3 步骤简介

序号	说明
1 了解迁移步骤	请务必在执行迁移前，完整阅读本节步骤简介，了解每个步骤的作用
2 新容器化工程准备	容器化部署 FineReport/FineBI，需要通过 FineOps 运维平台前端实现
	因此用户需要先部署一个 FineOps 运维平台
	迁移项目，是从原有非容器化工程，迁移到新的容器化工程
	因此用户需要先通过 FineOps 运维平台部署一个新的容器化单机工程
2.3 确认新工程版本	由于容器化部署的工程，一般为帆软发布的较新版本
	因此用户需要先确认新工程版本，确保后续升级旧工程到同一版本

		迁移时，需要将老工程的一些文件拷贝到新工程中，
		这个操作需要在新工程停止运行的状态下进行
2.4	关闭新工程	因此用户要通过 FineOps 运维平台前端快速关闭新工程
		容器化部署新工程时，会自动生成工程部署所用语句，后续运营维护时可能会需要使用/查找，在迁移
2.5	备份新工程 config 文件夹	过程中该语句所在文件会被覆盖
		因此用户需要提前异地备份该文件所在 config 文件夹
		在进行工程迁移前，为避免迁移失败，导致工程文件
3.1	备份原工程	丢失无法回退等问题，请务必对原工程进行备份后再进行后续操作
		工程迁移必须基于同版本的工程进行
3.2	升级原工程	因此需要将原非容器化部署的工程升级到新工程同一版本，方可进行迁移
3		为了保证新老工程的平台配置完全一致，需要同步
原非容器化工数据库		两个工程的配置库
3.3	确认原工程配置外接数据库	因此用户需要先确认好原工程所用的 finedb 数据库
程准备		信息
		迁移时，需要将原工程的插件覆盖新工程，而运维平
3.4	确认原工程已安装插件	台对接项目时，部分功能对项目的插件有要求
		因此需要先确认原工程已安装符合版本要求的插件
		迁移时，需要将原工程的一些文件拷贝到新工程中，
3.5	关闭原工程	这个操作需要在原工程停止运行的状态下进行
		因此用户需要手动关闭原工程
4	4.1 拷贝文件	本节即为正式迁移操作

迁 移 工 程	将原非容器化工程的必要文件，拷贝到新容器化工 程中  接下来的操作需要在新工程平台上执行
4.2 启动新工程	因此用户要通过 FineOps 运维平台前端快速启动新 工程  由于此时新老工程均使用了同一个 finedb 数据库， 而每个 finedb 数据库只能对接一个工程，否则可能
4.3 迁移新工程外接数据 库	导致使用异常  因此用户需要将新工程的外接数据库迁移到其他数 据库  仅 FineBI 工程需要执行本节操作  工程迁移后，FineBI 一般需要重新进行数据抽取，业
4.4 抽取数据	务用户方可正常使用  因此用户需要在抽取前确认并修改数据存储路径， 再执行数据抽取
5 启动原非容器化工程	如对原非容器化工程仍有使用诉求，可参考本节手动启动原 工程  若老工程进行了注册，一般需要将老工程的授权迁
6 迁移授权	移到新工程  容器化工程推荐使用「容器私有云认证」方式

2. 新容器化工程准备

注：建议部署新工程、老工程、运维平台所用服务器内网互通，会大幅降低迁移难度。

2.1 部署运维平台

详情请参见：[部署运维平台](#)

2.2 部署新工程

详情请参见：[部署新 FineReport 单机项目](#)、[部署新 FineBI 单机项目](#)

注意事项 1：迁移时，需要将老工程的一些必要文件、资源拷贝到新工程中，因此必须  
要确保新工程磁盘空间足够大。

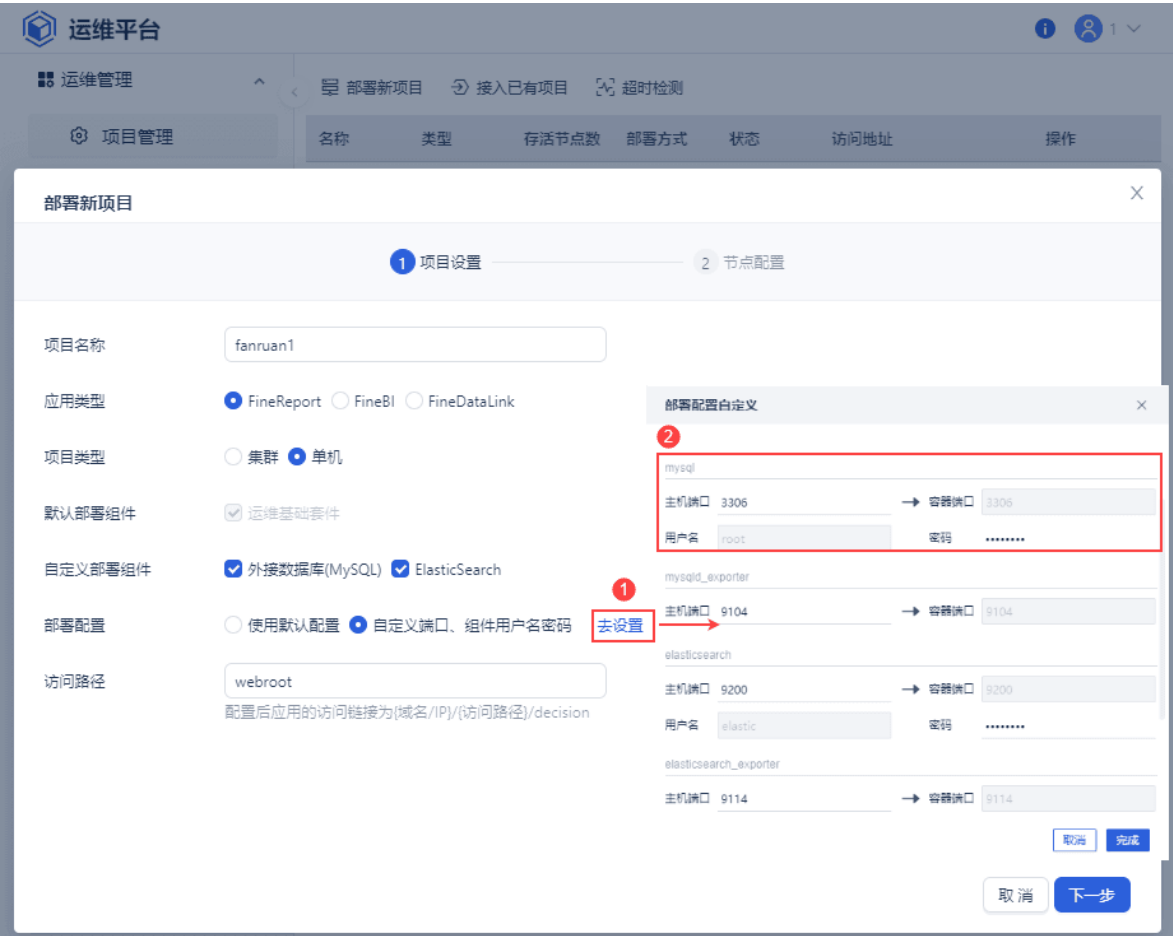


- 1) 检查原非容器化工程%Tomcat\_home%\webapps\webroot 文件夹的大小。假设为 xG。
- 2) 在准备新容器化单机工程所在服务器时，请根据原工程大小，适度扩大新容器化工程所在服务器节点剩余磁盘空间。

建议新工程最小剩余磁盘空间大小为 **max(2x , 500) G**。

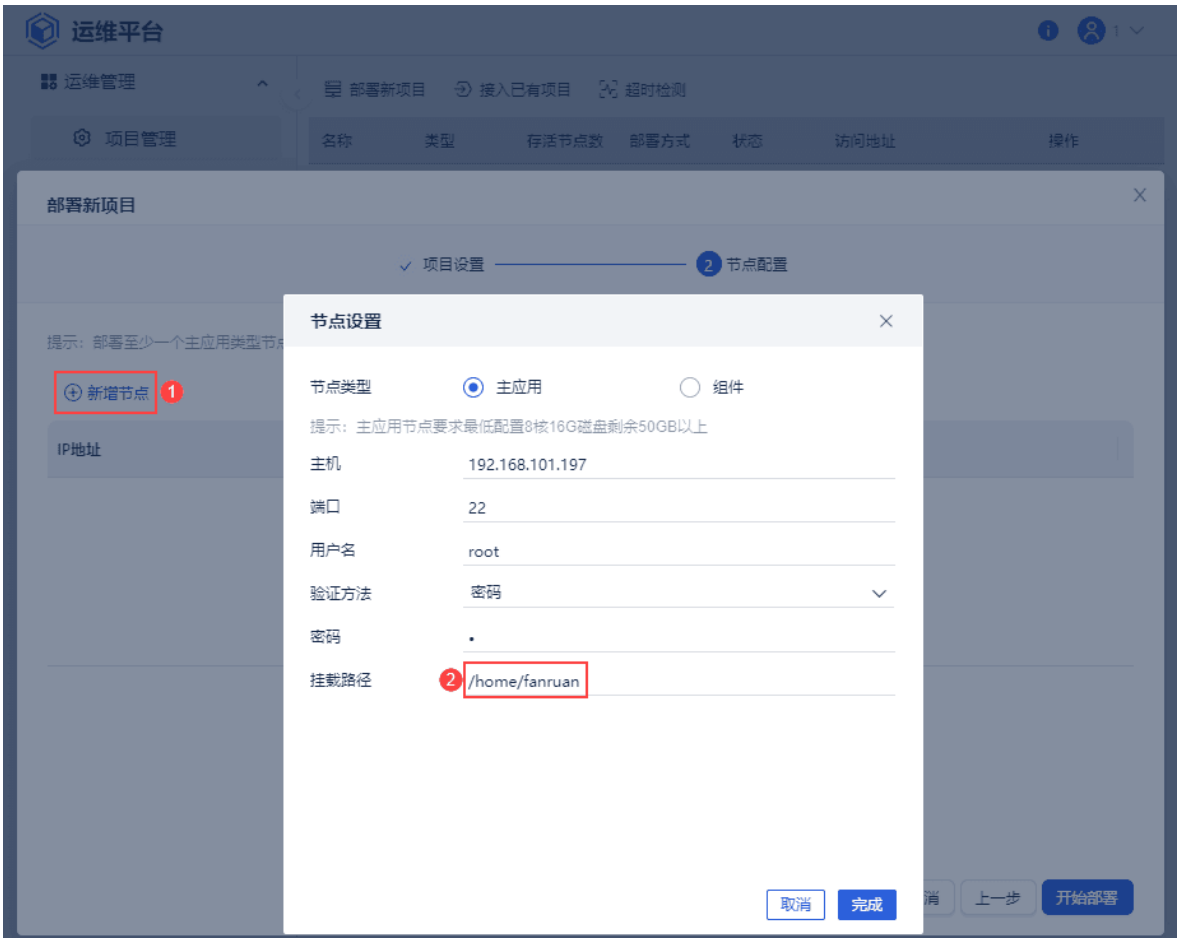
**注意事项 2：部署新工程时，会同时部署一些必要组件，建议记录下组件的信息，方便在后续维护过程中查看。**

请在「部署新项目-项目设置」环节，记录下 MySQL、elasticsearch 组件的相关信息。由于组件密码随机生成，请务必修改组件的密码，方便后续使用。



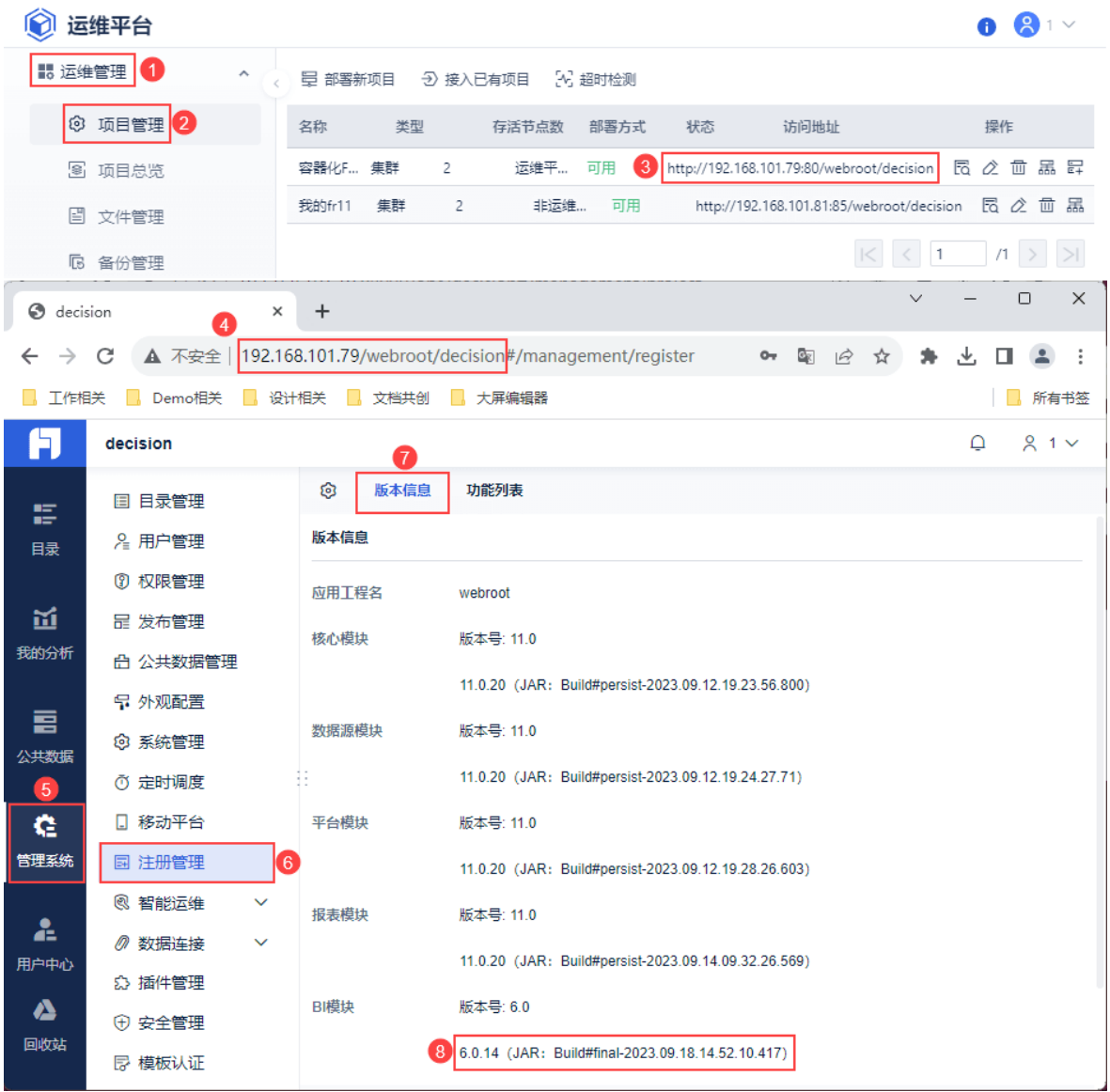
**注意事项 3：迁移时，需要将老工程的一些文件拷贝到新工程中，因此需要记录下新工程存放文件的外挂目录位置。**

请在「部署新项目-节点配置」环节，记录下工程的挂载路径。



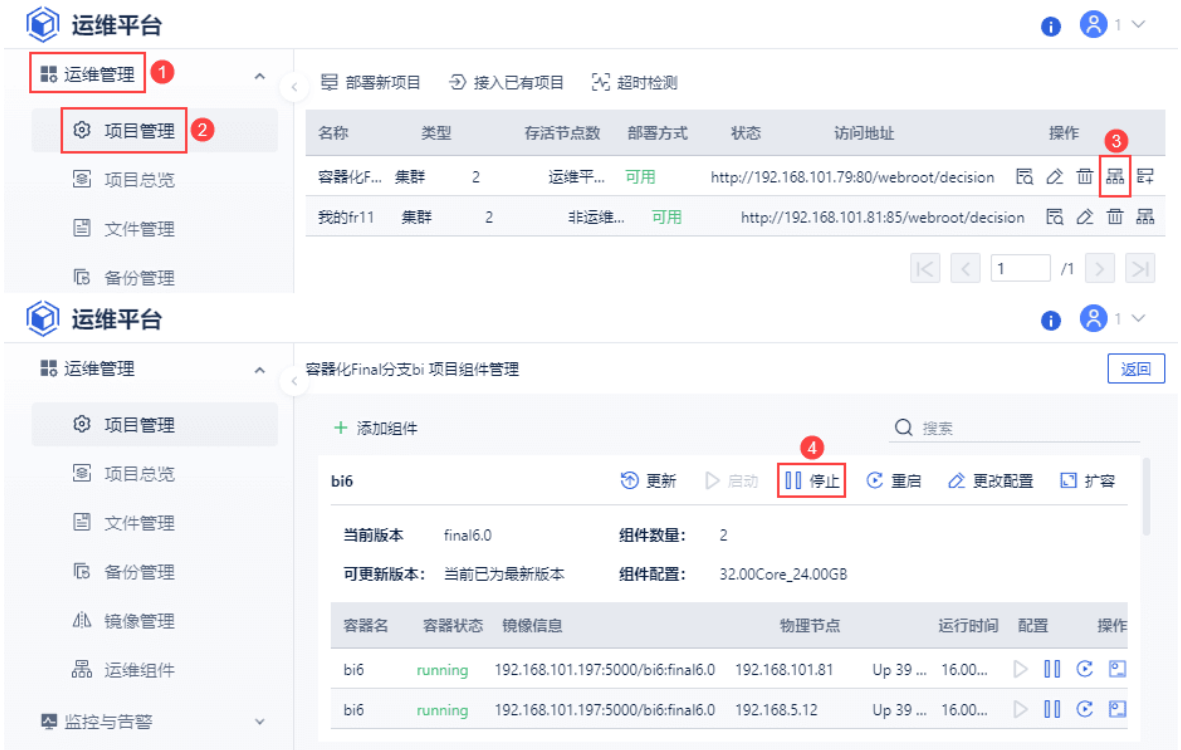
### 2.3 确认新工程版本

- 1) 管理员登录 FineOps 运维平台，在「运维管理>项目管理」中，可查看新工程的访问地址。
- 2) 管理员登录新容器化工程，点击「管理系统>注册管理>版本信息」，可查看新容器化工程的小版本号（精确到 JAR 包日期）。



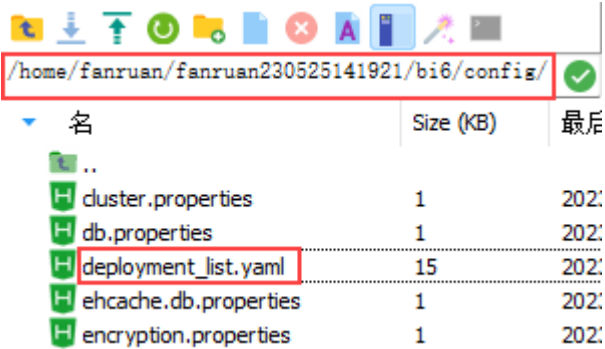
## 2.4 关闭新工程

- 1) 管理员登录 FineOps 运维平台，在「运维管理>项目管理」中，点击新容器化部署的项目后的「组件管理」按钮。
- 2) 找到 bi6/fr 组件，点击「停止」按钮，即可关闭新容器化工程。



2.5 备份新工程 config 文件夹

- 1) 使用终端设备访问新容器化工程所在服务器，进入 2.2 节记录的挂载路径 /fanruanxxx/bi 目录下。
- 2) 找到 config 文件夹，文件夹中存放的 yaml 文件有着工程部署所用语句，请务必异地备份。



3. 原非容器化工程准备

3.1 备份原工程

详情请参见 [工程备份还原方案](#) 3.1 节，对工程进行备份。

3.2 升级原工程

1) 获取 JAR 包

付费用户请联系帆软技术支持，获取 2.3 节新工程同一小版本的工程 JAR 包（精确到 JAR 包日期）。

技术支持联系方式：前往「[服务](#)」，选择「在线支持」、电话「400-811-8890」。

2) 升级原非容器化工程

请参考「[FineBI 小版本升级](#) / [FineReport 小版本升级](#)」文档升级工程至指定版本。

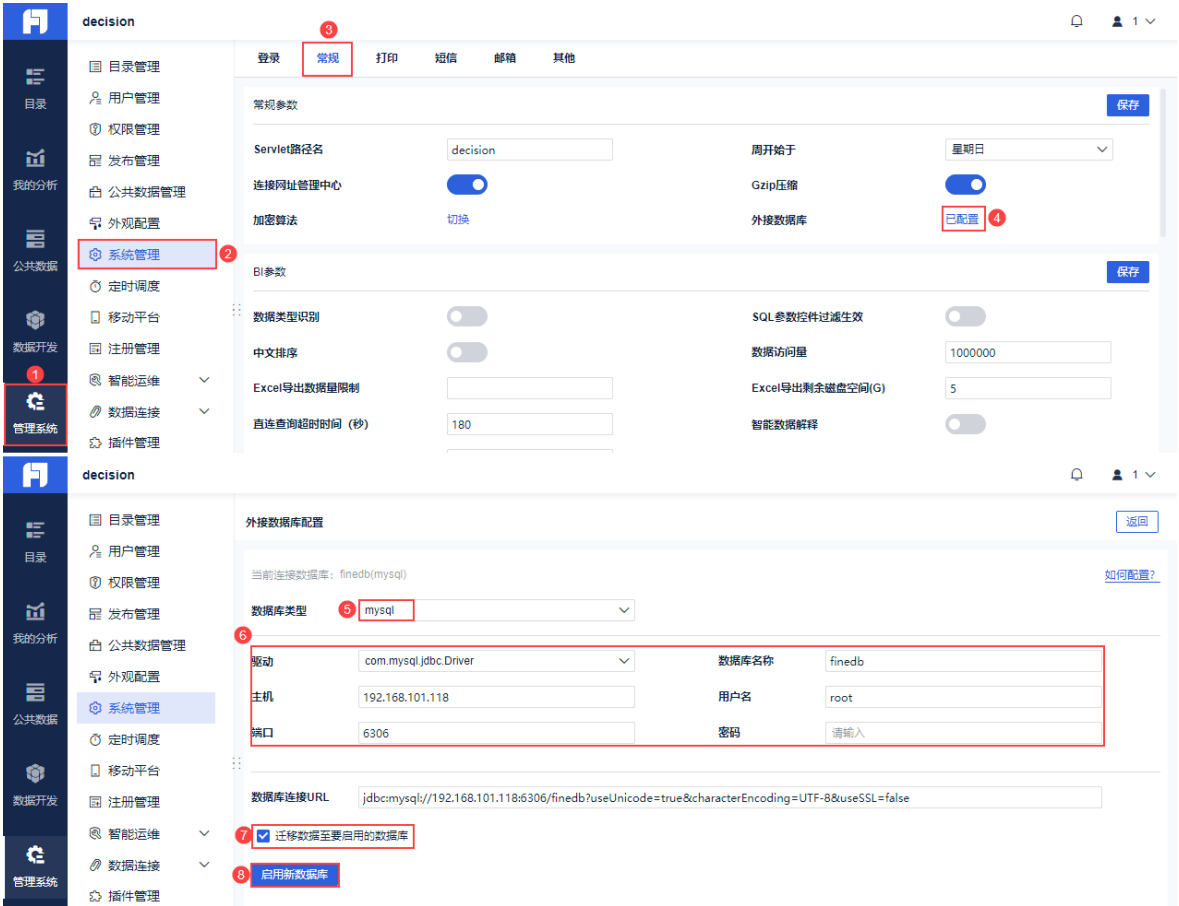
3.3 确认原工程配置外接数据库

管理员登录原非容器化工程，点击「管理系统>系统管理>常规」，检查外接数据库是否已配置。

情况	说明
已配置	说明工程已配置外接数据库，点击「已配置」，查看并记录下该外接数据库位
	置信息
未配置	用户必须确保新容器化工程所在服务器与该外接数据库所在服务器互通
	说明工程未配置外接数据库，请先为该非容器化工程配置一个外接数据库，可

使用 2.2 节的 MySQL 组件

请确保新容器化工程所在服务器与该外接数据库所在服务器互通, 请在配置时勾选「迁移数据至要启用的数据库」



3.4 确认原工程已安装插件

管理员登录原非容器化工程，点击「管理系统>插件管理」。

- 1) 必须确认安装以下插件，并升级至最新：系统运维、资源调度熔断插件。
- 2) 建议升级其他已安装插件至最新版本。



### 3.4 关闭原工程

请参考 [关闭或重启 FineBI 工程](#) / [关闭或重启 FineReport 工程](#) 文档, 关闭原非容器化部署的工程。

## 4. 迁移工程

### 4.1 拷贝文件

- 1) 请提前检查原非容器化部署的工程，是否配置了外挂目录、是否配置共享文件服务器，确保拷贝到正确的文件。
- 2) 新容器化工程外挂文件所在位置，为 2.2 节记录的挂载路径。
- 3) 将以下文件，从原非容器化部署的工程，拷贝到新容器化部署的工程中：

注 1：请勿直接 `docker cp` 整个 `webroot` 文件夹到容器，会导致功能异常。

注 2：请勿直接拷贝整个 `webroot` 文件夹到容器化工程的外挂目录，部分文件不可以被覆盖。

注 2：请勿使用 3.1 节备份的文件，而是使用工程最终的文件进行拷贝操作。

- **FineBI 需要拷贝的文件**

原非容器化工程目录	新容器化工程外挂目录	说明
%Tomcat_HOME%/logs	%BI_HOME%/fanruanxxx/bi6/tomcat_logs	作用：Tomcat 通用日志位置 是否必须拷贝：用户自行决定, Tomcat 历史访问日志，可不拷贝
%Tomcat_HOME%/webapps/webroot/backup	%BI_HOME%/fanruanxxx/bi6/backup	作用：工程历史备份文件 是否必须拷贝：用户自行决定，可存储在原始的目录下备份，可不拷贝
%Tomcat_HOME%/webapps/webroot/bi-data	%BI_HOME%/fanruanxxx/bi6/bi-data	作用：FineBI 抽取数据 是否必须拷贝：用户自行决定 如果不拷贝，在新工程启动后需要全局更新，重新抽取数据 如果拷贝，该文件夹中文件数量很多，拷

		贝耗时很久，请耐心等待
%Tomcat_HOME%/webapps/webroot/logs	%BI_HOME%/fanruanxxx/bi6/logs	<p>作用：swift 日志</p> <p>是否必须拷贝：<span>用户自行决定</span></p> <p>如果不拷贝，会丢失工程历史操作日志（即 logdb），「管理系统&gt;平台日志」功能无数据</p> <p>如对历史使用情况无要求，无需拷贝</p>
%Tomcat_HOME%/webapps/webroot/WEB-INF/assets/temp_attach	%BI_HOME%/fanruanxxx/bi6/assets/temp_attach	<p>作用：FineBI 数据表相关信息</p> <p>是否必须拷贝：必须拷贝</p> <p>该文件夹中存放着工程 Excel 原始文件信息，如不拷贝该文件夹会导致原始 Excel 丢失</p>
%Tomcat_HOME%/webapps/webroot/WEB-INF/assets/vcs	%BI_HOME%/fanruanxxx/bi6/assets/vcs	<p>作用：FineReport 模板备份文件</p> <p>是否必须拷贝：用户自行决定</p> <p>如果工程未使用 FineReport 模板，或不需要回退历史开发的 FineReport 模板，则无需拷贝</p>
%Tomcat_HOME%/webapps/webroot/WEB-INF/assets/其他文件	%BI_HOME%/fanruanxxx/bi6/assets/其他文件	<p>作用：通用的共享持久化目录</p> <p>是否必须拷贝：必须拷贝</p> <p>工程正常运行所需要的文件，如不拷贝会影响工程正常使用</p>
%Tomcat_HOME%/webapps/webroot/WEB-INF/classes	%BI_HOME%/fanruanxxx/bi6/classes	<p>作用：工程调用的默认和自定义 class 文件</p> <p>是否必须拷贝：必须拷贝</p> <p>有可能存在自定义的 class 文件，如不拷贝会影响工程正常使用</p>
%Tomcat_HOME%/webapps/webroot/WEB-INF/config	%BI_HOME%/fanruanxxx/bi6/config	作用：配置库相关文件

	9	<p>是否必须拷贝：必须拷贝</p> <p>平台调用的配置库 (finedb)，如不拷贝会影响工程正常使用</p>
<p>找到以下两个目录：</p> <p>%Tomcat_HOME%/webapps/webroot/WEB-INF/lib</p> <p>%Tomcat_HOME%/webapps/webroot/WEB-INF/customLib</p> <p>以下为工程自带 JAR，<b>无需拷贝</b>，除此之外的用户自定义 JAR 均需拷贝：</p> <p>点击展开更多</p> <p>fdl-bi-extension-4.0.jar</p> <p>fdl-boot-4.0.jar</p> <p>fdl-core-4.0.jar</p> <p>fdl-cron-4.0.jar</p> <p>fdl-datasource-4.0.jar</p> <p>fdl-third-4.0.jar</p> <p>fine-accumulator-11.0.jar</p> <p>fine-activator-11.0.jar</p> <p>fine-autuator-formula-11.0.jar</p> <p>fine-bi-adapter-6.0.jar</p> <p>fine-bi-datamining-6.0.jar</p> <p>fine-bi-datamining-third-6.0.jar</p> <p>fine-bi-engine-spider-6.0.jar</p> <p>fine-bi-engine-third-6.0.jar</p> <p>fine-bi-foundation-6.0.jar</p> <p>fine-bi-middle-6.0.jar</p> <p>fine-bi-query-6.0.jar</p> <p>fine-bi-query-excel-6.0.jar</p> <p>fine-bi-query-third-6.0.jar</p> <p>fine-bi-scheduler-6.0.jar</p> <p>fine-bi-spider-cluster-6.0.jar</p> <p>fine-cbb-11.0.jar</p> <p>fine-core-11.0.jar</p> <p>fine-datasource-11.0.jar</p> <p>fine-decision-11.0.jar</p> <p>fine-decision-bi-11.0.jar</p> <p>fine-decision-report-11.0.jar</p> <p>fine-report-engine-11.0.jar</p> <p>fine-schedule-11.0.jar</p> <p>fine-schedule-bi-11.0.jar</p> <p>fine-schedule-report-11.0.jar</p> <p>fine-swift-log-adaptor-11.0.jar</p>	<p>%BI_HOME%/fanruanxxx/bi6/customlib</p>	<p>作用：工程定制/外部引入的 JAR 包</p> <p>是否必须拷贝：必须拷贝，如不拷贝会影响模板访问</p>



ar fine-third-11.0.jar fine-webui-11.0.jar h2-2.1.214.jar jtds-1.3.1.jar mssql-jdbc-9.4.1.jre8.jar mysql-connector-java-5.1.4 9-bin.jar ojdbc8.jar orai18n.jar sqlite-jdbc-3.35.4.jar jarfile-checksum.txt readme.txt version-bi.txt		
%Tomcat_HOME%/webapps/webroot/WEB-INF/dpworks	%BI_HOME%/fanruanxxx/bi6/dpworks	作用：FineDataLink 任务相关的配置文件  是否必须拷贝：必须拷贝，如不拷贝会影响 FineDataLink 任务使用  （如不存在该文件夹，说明工程没有使用 FDL 相关功能，略过）
%Tomcat_HOME%/webapps/webroot/WEB-INF/plugins	%BI_HOME%/fanruanxxx/bi6/plugins	作用：插件相关文件  是否必须拷贝：必须拷贝，如不拷贝会影响插件相关功能的实现。  请先删除文件夹中原有文件，再将老工程插件包拷贝进来，否则同一个插件可能共存多个版本，导致异常。
%Tomcat_HOME%/webapps/webroot/WEB-INF/reportlets	%BI_HOME%/fanruanxxx/bi6/reportlets	作用：FineReport 模板存放目录  是否必须拷贝：用户自行决定  如果工程未使用 FineReport 模板，则无需拷贝
%Tomcat_HOME%/webapps/webroot/WEB-INF/schedule	%BI_HOME%/fanruanxxx/bi6/schedule	作用：定时调度生成的文件  是否必须拷贝：用户自行决定  如果不拷贝，定时任务挂载到决策平台的结果报表无法访问

- **FineReport 需要拷贝的文件**

原非容器化工程目录	新容器化工程外挂目录	说明
%Tomcat_HOME%/logs	%FR_HOME%/fanruanxxx/fr/tomcat_logs	作用：Tomcat 通用日志位置 是否必须拷贝：用户自行决定，Tomcat 历史访问日志，可不拷贝
%Tomcat_HOME%/webapps/webroot/backup	%FR_HOME%/fanruanxxx/fr/backup	作用：工程历史备份文件 是否必须拷贝：用户自行决定，可存储在原始的目录下备份，可不拷贝
%Tomcat_HOME%/webapps/webroot/logs	%FR_HOME%/fanruanxxx/fr/logs	作用：swift 日志 是否必须拷贝：用户自行决定 如果不拷贝，会丢失工程历史操作日志（即 logdb），「管理系统>平台日志」功能无数据 如对历史使用情况无要求，无需拷贝
%Tomcat_HOME%/webapps/webroot/WEB-INF/assets/temp_attach	%FR_HOME%/fanruanxxx/fr/assets/temp_attach	作用：读写缓存存储路径 是否必须拷贝：必须拷贝 该文件夹中存放着工程读写（图片）缓存，不拷贝可能会导致模板设置的背景图片预览为空
%Tomcat_HOME%/webapps/webroot/WEB-INF/assets/vcs	%FR_HOME%/fanruanxxx/fr/assets/vcs	作用：FineReport 模板备份文件 是否必须拷贝：用户自行决定 如果工程不需要回退历史开发的 FineReport 模板，则无需拷贝
%Tomcat_HOME%/webapps/webroot/WEB-INF/assets/其他文件	%FR_HOME%/fanruanxxx/fr/assets/其他文件	作用：通用的共享持久化目录 是否必须拷贝：必须拷贝 工程正常运行所需要的文件，如不拷贝会影响工程正常使用

%Tomcat_HOME%/webapps/webroot/WEB-INF/classes	%FR_HOME%/fanruanxxx/fr/classes	<p>作用：工程调用的默认和自定义 class 文件</p> <p>是否必须拷贝：必须拷贝</p> <p>有可能存在自定义的 class 文件，如不拷贝会影响工程正常使用</p>
%Tomcat_HOME%/webapps/webroot/WEB-INF/config	%FR_HOME%/fanruanxxx/fr/config	<p>作用：配置库相关文件</p> <p>是否必须拷贝：必须拷贝</p> <p>平台调用的配置库 (finedb)，如不拷贝会影响工程正常使用</p>
<p>找到以下两个目录：</p> <p>%Tomcat_HOME%/webapps/webroot/WEB-INF/lib</p> <p>%Tomcat_HOME%/webapps/webroot/WEB-INF/customLib</p> <p>以下为工程自带 JAR，无需拷贝，除此之外的用户自定义 JAR 均需拷贝：</p> <p>点击展开更多</p> <p>fine-accumulator-11.0.jar</p> <p>fine-activator-11.0.jar</p> <p>fine-cbb-11.0.jar</p> <p>fine-core-11.0.jar</p> <p>fine-datasource-11.0.jar</p> <p>fine-decision-11.0.jar</p> <p>fine-decision-report-11.0.jar</p> <p>fine-report-engine-11.0.jar</p> <p>fine-schedule-11.0.jar</p> <p>fine-schedule-report-11.0.jar</p> <p>fine-service-management-2.0.7.jar</p> <p>fine-swift-log-adaptor-11.0.jar</p> <p>fine-third-11.0.jar</p> <p>fine-webui-11.0.jar</p> <p>ifxjdbc_informix.jar</p> <p>jtds-1.3.1.jar</p> <p>mysql-connector-java-5.1.49-bin.jar</p> <p>ojdbc8.jar</p> <p>orai18n.jar</p>	%FR_HOME%/fanruanxxx/fr/customlib	<p>作用：工程定制/外部引入的 JAR 包</p> <p>是否必须拷贝：必须拷贝，如不拷贝会影响模板访问</p>

sqlite-jdbc-3.39.4.0.jar sqljdbc.jar sybase.jar jarfile-checksum.txt readme.txt		
%Tomcat_HOME%/webapps/webroot/WEB-INF/dpworks	%FR_HOME%/fanruanxxx/fr/dpworks	作用: FineDataLink 任务相关的配置文件  是否必须拷贝: 必须拷贝, 如不拷贝会影响 FineDataLink 任务使用  (如不存在该文件夹, 说明工程没有使用 FDL 相关功能, 略过)
%Tomcat_HOME%/webapps/webroot/WEB-INF/plugins	%FR_HOME%/fanruanxxx/fr/plugins	作用: 插件相关文件  是否必须拷贝: 必须拷贝, 如不拷贝会影响插件相关功能的实现。  请先删除文件夹中原有文件, 再将老工程插件包拷贝进来, 否则同一个插件可能共存多个版本, 导致异常。
%Tomcat_HOME%/webapps/webroot/WEB-INF/reportlets	%FR_HOME%/fanruanxxx/fr/reportlets	作用: FineReport 模板存放目录  是否必须拷贝: 必须拷贝, 如不拷贝会导致工程所有模板丢失
%Tomcat_HOME%/webapps/webroot/WEB-INF/schedule	%FR_HOME%/fanruanxxx/fr/schedule	作用: 定时调度生成的文件  是否必须拷贝: 用户自行决定  如果不拷贝, 定时任务挂载到决策平台的结果报表无法访问

4.2 启动新工程

用户登录运维平台, 点击「运维管理>项目管理」, 点击新容器化部署的工程项目后的「组件管理」按钮。

找到 bi6/fr 组件, 点击「启动」按钮, 即可启动新容器化工程。如下图所示:



4.3 迁移新工程外接数据库

1) 新建空数据库

- 如果老工程外接数据库使用的非 MySQL8 类型数据库：

新工程外接数据库仍然部署在该数据库下的新库里，2.2 节的 MySQL 数据库无需使用

请在老工程原外接数据库地址新建一个库名，作为待迁移的空库

- 如果老工程外接数据库使用的 MySQL8 类型数据库：

新工程外接数据库建议部署在 2.2 节的 MySQL 数据库里

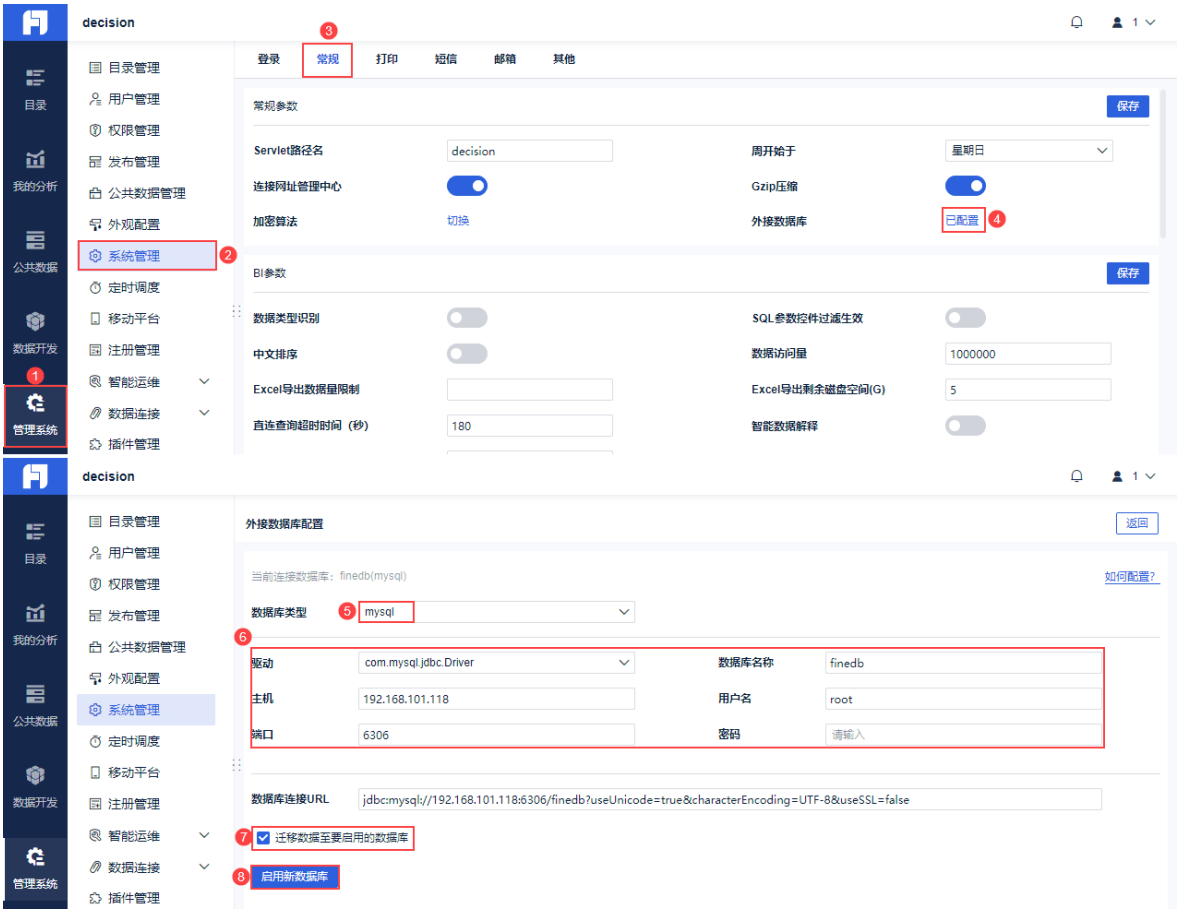
请在 2.2 节容器化部署的 MySQL8 组件新建一个库名，作为待迁移的空库

2) 迁移数据库

管理员登录新容器化工程， 点击「管理系统>系统管理>常规>外接数据库」。

- 配置外接数据库为上一步的空库
- 勾选「迁移数据至要启用的数据库」
- 点击「启用新数据库」

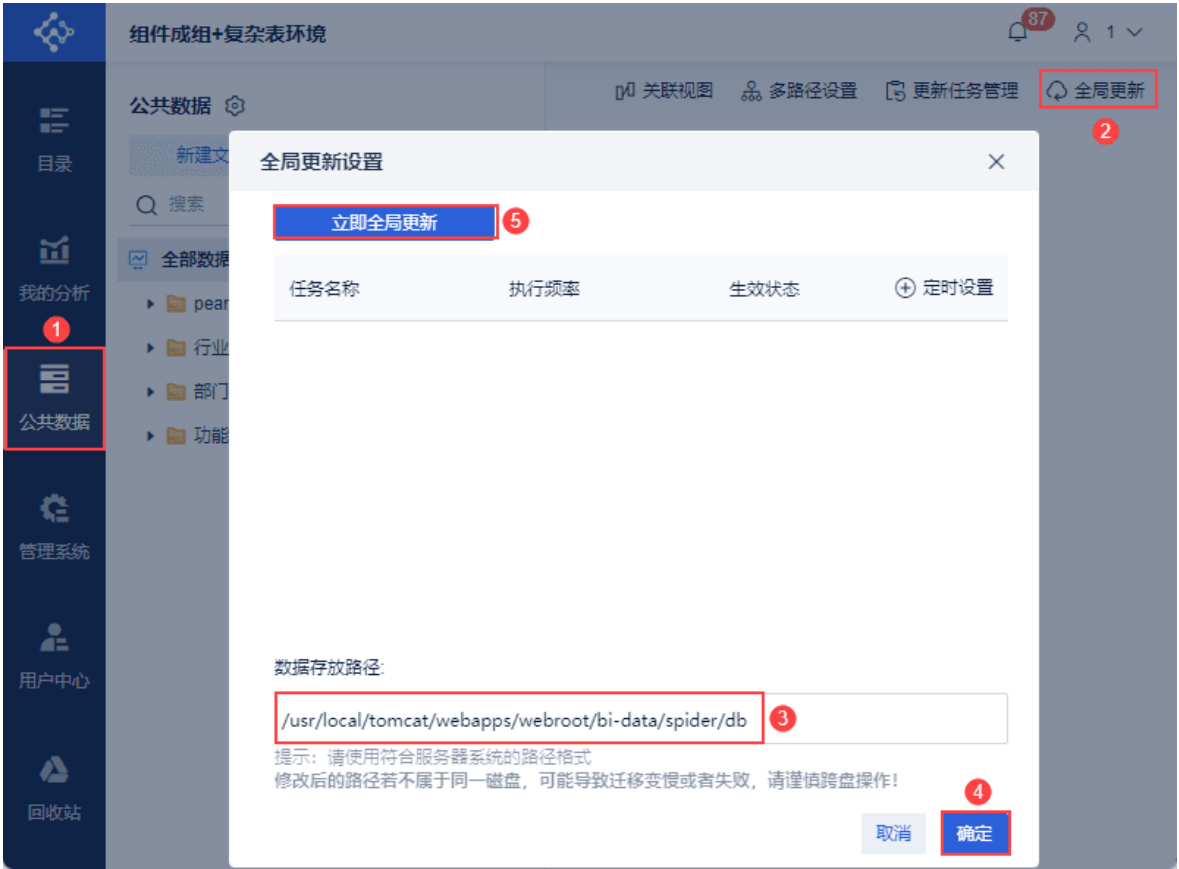
当出现提示「已成功切换至目标数据库!」，则代表外接数据库切换成功。



#### 4.4 抽取数据

注：仅 FineBI 工程需要执行本节操作。

- 1) 管理员登录新工程，点击「公共数据>全局更新」，即可看到数据的存储路径。
- 2) 请确修改为新工程容器内数据存放路径： /usr/local/tomcat/webapps/webroot/bi-data/spider/db，点击「确定」，更改成功。
- 3) 再次访问「公共数据>全局更新」，执行「立即全局更新」，即可抽取最新数据，并存放到正确的存储路径中。



5. 启动原非容器化工程

参考 [关闭或重启 FineBI 工程](#) / [关闭或重启 FineReport 工程](#) 文档，启动原非容器化部署的工程。

6. 迁移授权

若老工程进行了注册，一般需要将授权迁移到新工程。容器化工程推荐使用 [容器私有云认证](#) 方式。

- 1) 参考 [授权迁移插件](#) 的使用步骤，向商务索要一个授权迁移插件，销毁原非容器化工程的旧授权。
- 2) 找到原非容器化工程%BI\_HOME%/webroot/WEB-INF/resources 文件夹下的销毁凭证。
- 4) 参考 [容器私有云认证](#) 的使用步骤，向商务索要注册镜像文件 fanruan\_license\_server.tar，并参考文档为**新容器化工程**进行授权。

请在邮件中附上第三步获取的销毁凭证。并注明是「非容器化工程迁移容器化工程」

## 4.5.2 非容器化单机迁移容器化集群

### 1. 概述

#### 1.1 版本

工程版本	功能变更
FineReport11.0	-
FineBI6.0	-

#### 1.2 应用场景

相比于普通单机工程，容器化部署的工程，升级和维护成本较低。

相比于单机工程，集群具有高可用性、高性能、易于管理、可伸缩性和安全保障等优点，适用于企业级的报表生成和管理需求。

本文方案，帮助用户将原有非容器化部署的单机工程，迁移为容器化部署的集群工程。

注：全文以 FineBI 作为演示。FineReport 迁移步骤基本一致，仅需要两个点：

- 1) 拷贝文件时，FineReport 和 FineBI 需要拷贝的文件不完全相同，注意文档中的表格即可。
- 2) 使用运维平台启停容器时，FineReport 关闭的是 fr 组件，FineBI 关闭的是 bi6 组件。

### 2. 新容器化工程准备

#### 2.1 部署运维平台

详情请参见：[部署运维平台](#)

#### 2.2 部署新工程

详情请参见：[部署新 FineBI 集群项目](#)、[部署新 FineReport 集群项目](#)

**注意事项 1：**迁移时，需要将老工程的一些必要文件、资源拷贝到新工程中，因此必须确保新工程磁盘空间足够大。

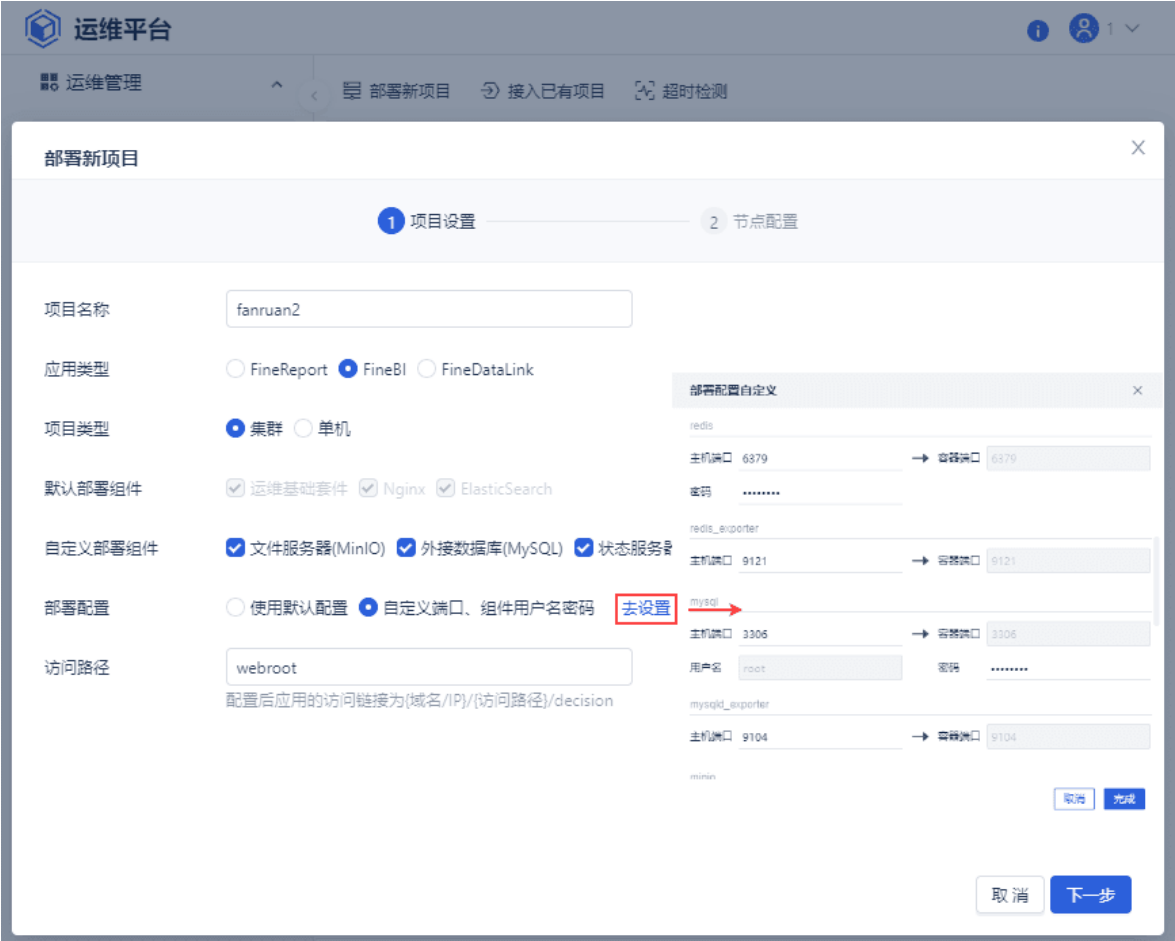
- 1) 检查原非容器化工程%Tomcat\_home%\webapps\webroot 文件夹的大小。假设为 xG。
- 2) 在准备新容器化集群文件服务器时，请根据原工程大小，适度扩大新工程文件服务器所在节点剩余磁盘空间。

建议最小剩余磁盘空间大小为 **max(2x , 500) G**。

**注意事项 2：**部署新工程时，会同时部署一些必要组件，建议记录下组件的信息，方便在后续维护过程中查看。

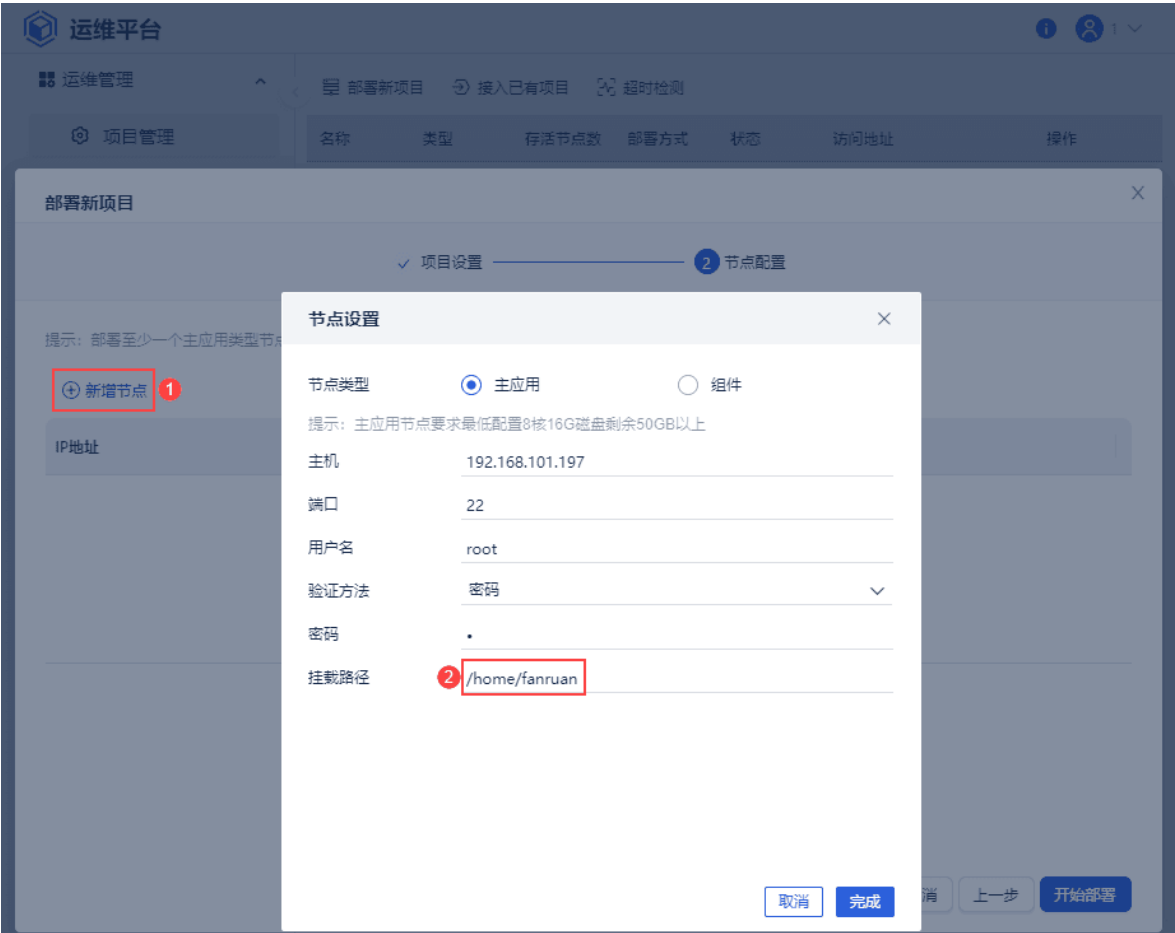
请在「部署新项目-项目设置」环节，记录下 MySQL、Redis、minio、elasticsearch 组件的相关信息。由于组件密码随机生成，请务必修改组件的密码，方便后续使用。





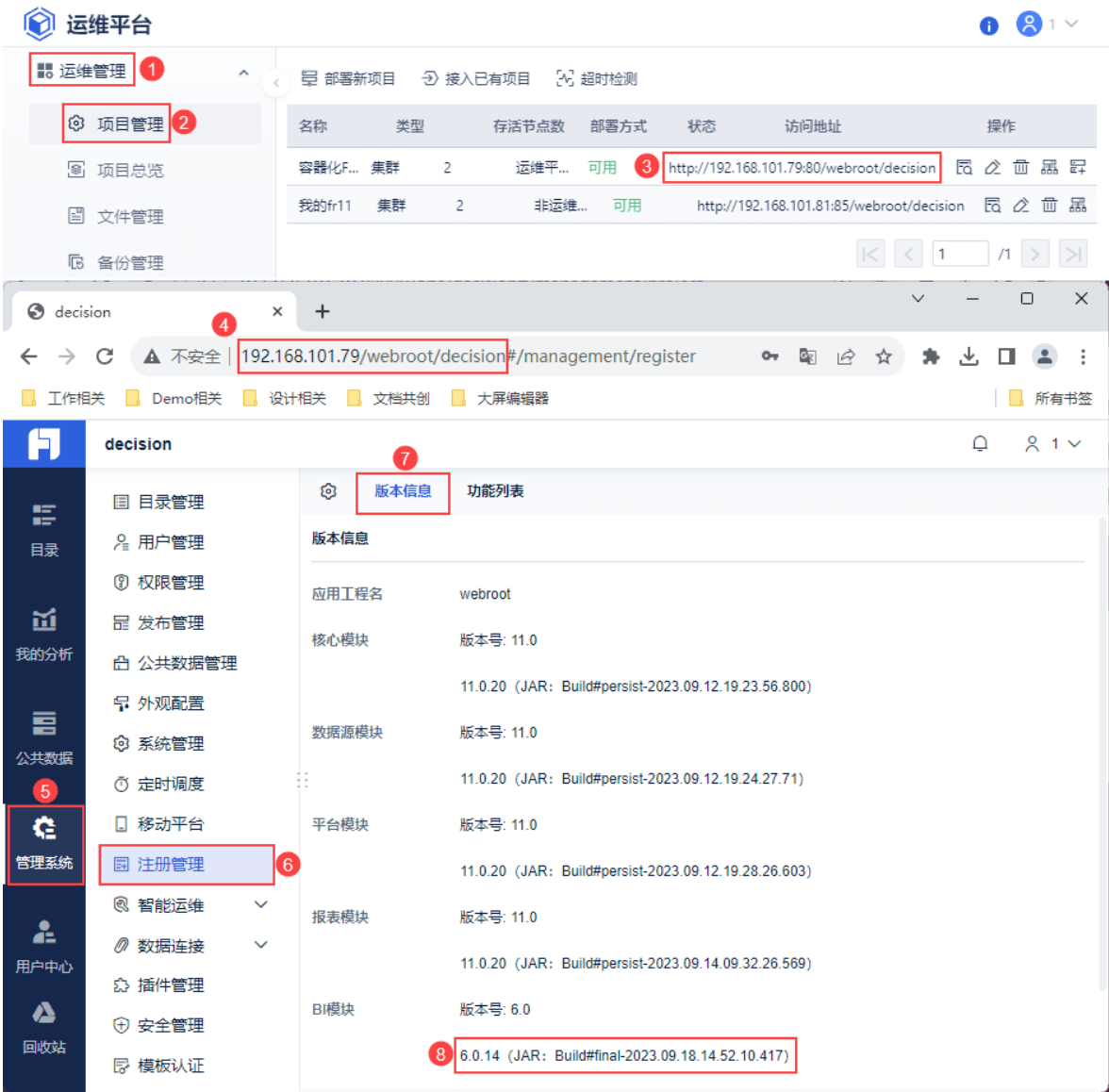
注意事项 3：迁移时，需要将老工程的一些文件拷贝到新工程中，因此需要记录下新工程存放文件的外挂目录位置。

请在「部署新项目-节点配置」环节，记录下集群每个工程节点的挂载路径。



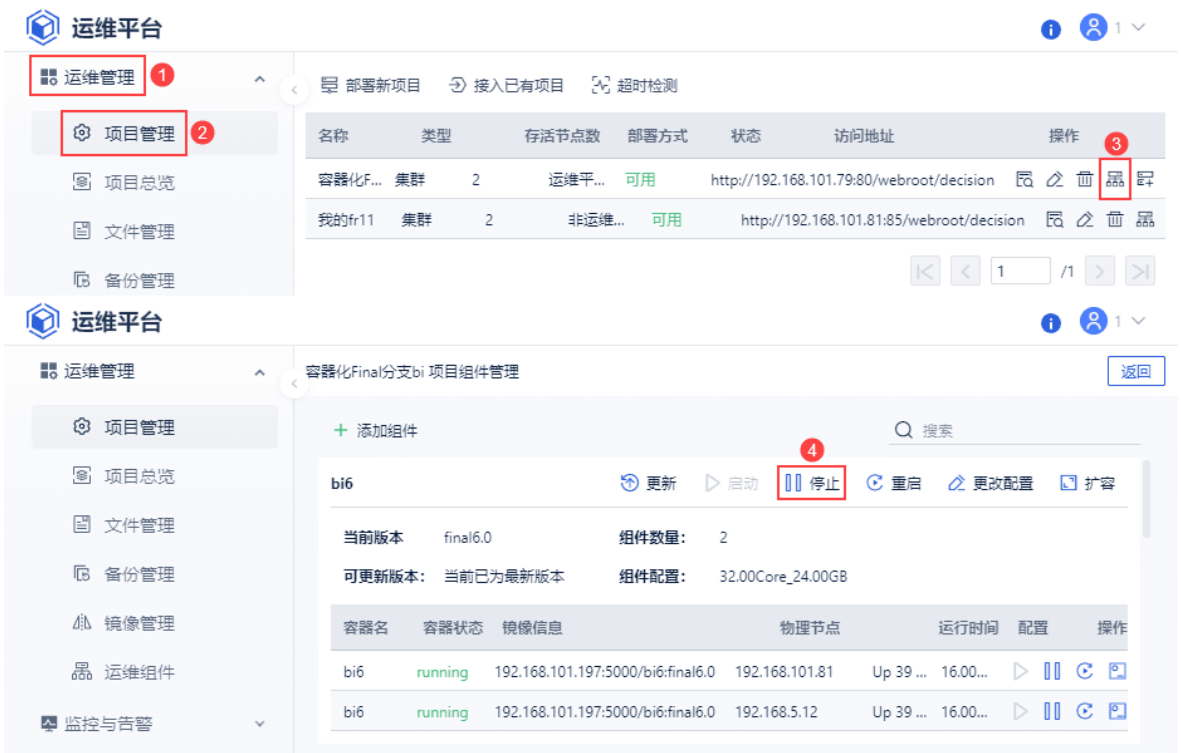
## 2.3 确认新工程版本

- 1) 管理员登录 FineOps 运维平台，在「运维管理>项目管理」中，可查看新工程的访问地址。
- 2) 管理员登录新容器化工程，点击「管理系统>注册管理>版本信息」，可查看新容器化工程的小版本号（精确到 JAR 包日期）。



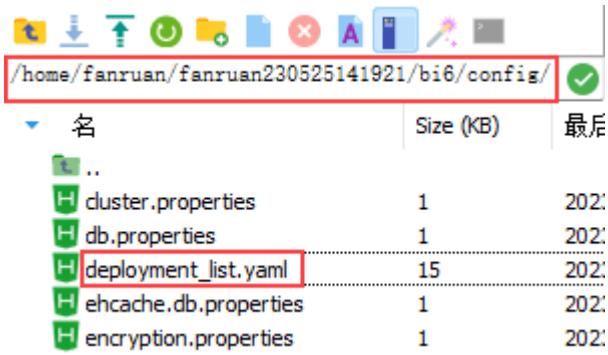
## 2.4 关闭新工程

- 1) 管理员登录 FineOps 运维平台，在「运维管理>项目管理」中，点击新容器化部署的项目后的「组件管理」按钮。
- 2) 找到 bi6/fr 组件，点击「停止」按钮，即可关闭新容器化工程。



2.6 备份新工程 config 文件夹

- 1) 使用终端设备访问新容器化集群每一工程节点所在服务器，进入 2.2 节记录的挂载路径/fanruanxxx/bi 目录下。
- 2) 找到 config 文件夹，文件夹中存放的 yaml 文件有着工程部署所用语句，请务必异地备份。



3. 原非容器化工程准备

3.1 备份原工程

详情请参见 [工程备份还原方案](#) 3.1 节，对工程进行备份。

3.2 升级原工程

1) 获取 JAR 包

付费用户请联系帆软技术支持，获取 2.3 节新工程同一小版本的工程 JAR 包（精确到 JAR 包日期）。

技术支持联系方式：前往「[服务](#)」，选择「在线支持」、电话「400-811-8890」。

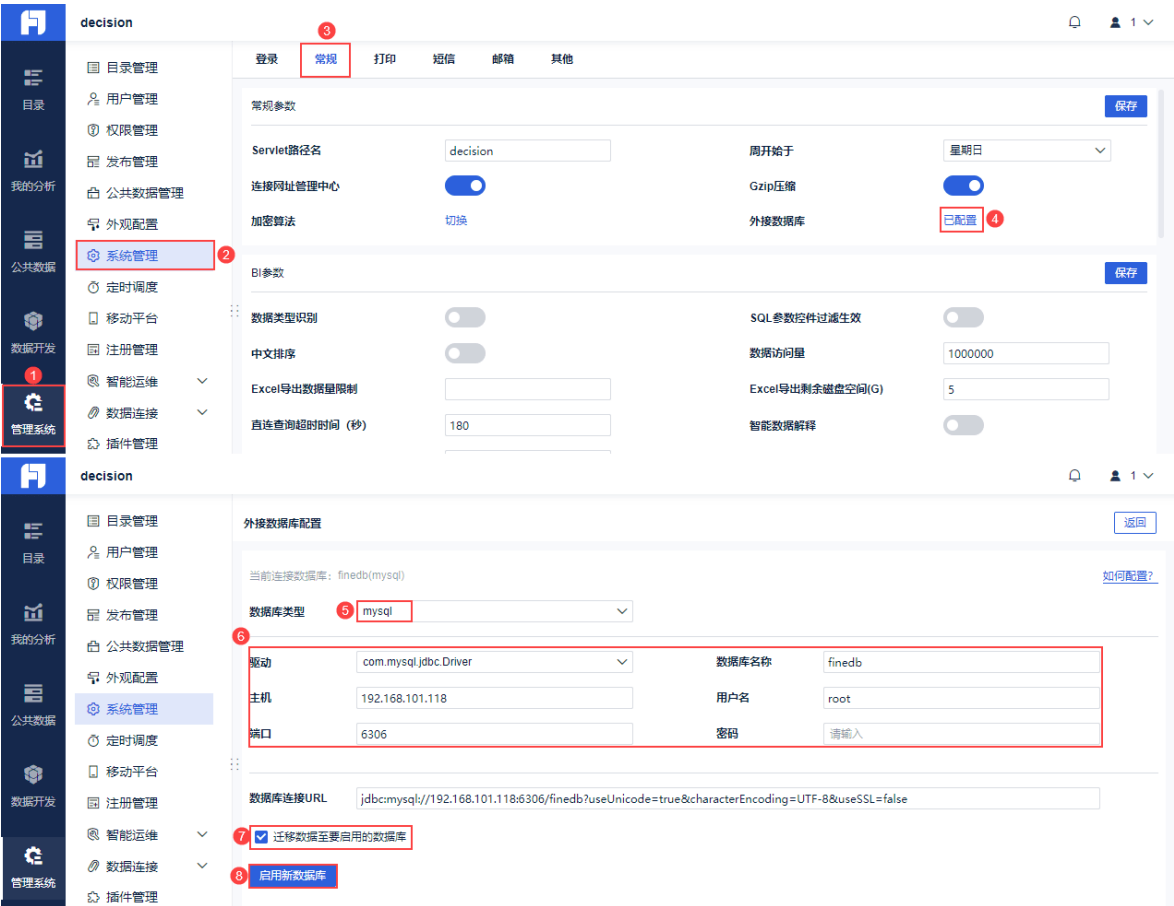
2) 升级原非容器化工程

请参考「[FineBI 小版本升级](#) / [FineReport 小版本升级](#)」文档升级工程至指定版本。

3.3 确认原工程配置外接数据库

管理员登录原非容器化工程，点击「管理系统>系统管理>常规」，检查外接数据库是否已配置。

情况	说明
已配置	说明工程已配置外接数据库，点击「已配置」，查看并记录下该外接数据库位 置信息  用户必须确保新容器化工程所在服务器与该外接数据库所在服务器互通
未配置	说明工程未配置外接数据库，请先为该非容器化工程配置一个外接数据库，可 使用 2.2 节的 MySQL 组件  请确保新容器化工程所在服务器与该外接数据库所在服务器互通，请在配置时 勾选「迁移数据至要启用的数据库」



3.4 确认原工程已安装插件

管理员登录原非容器化工程，点击「管理系统>插件管理」。

1) 必须确认安装以下插件，并升级至最新：minio 资源仓库、系统运维、资源调度熔断插件。

2) 建议升级其他已安装插件至最新版本。



3.5 对接集群组件

管理员登录原非容器化工程， 点击「管理系统>智能运维>集群配置」。

- 1) 开启「状态服务器」。选择「单机 Redis」，输入 2.2 节记录的 Redis 组件信息。
- 2) 开启「文件服务器共享」。选择「MinIO」，输入 2.2 节记录的 MinIO 组件信息。
- 3) 缓存模式选择「关闭缓存」， 集群通信协议选择「TCP」。
- 4) 开启「开启集群」按钮。

请参考 [关闭或重启 FineBI 工程](#) / [关闭或重启 FineReport 工程](#) 文档， 重启原非容器化部署的工程。

再次查看原非容器化工程的「管理系统>智能运维>集群配置」， 在节点管理处出现本工程节点， 说明单节点集群部署成功。



### 3.6 关闭原非容器化工程

请参考 [关闭或重启 FineBI 工程](#) / [关闭或重启 FineReport 工程](#) 文档，关闭原非容器化部署的工程。

## 4. 迁移工程

### 4.1 拷贝文件

- 1) 请提前检查原非容器化部署的工程，是否配置了外挂目录、是否配置共享文件服务器，确保拷贝到正确的文件。
- 2) 新容器化工程外挂文件所在位置， 为 2.2 节记录的挂载路径。
- 3) 将以下文件， 从原非容器化部署的工程， 拷贝到新容器化部署的工程中：
- 注 1： 请勿直接 docker cp 整个 webroot 文件夹到容器， 会导致功能异常。
- 注 2： 请勿直接拷贝整个 webroot 文件夹到容器化工程的外挂目录， 部分文件不可以被覆盖。
- 注 3： 请勿使用 3.1 节备份的文件， 而是使用工程最终的文件进行拷贝操作。
- 注 4： 文件需要一一拷贝到新容器化部署的每一个工程节点和文件服务器对应位置。

• **FineBI 需要拷贝的文件**

原非容器化工程目录	新容器化工程外挂目录	说明
%Tomcat_HOME%/logs	每个工程节点%BI_HOME%/fanruanxxx/bi6/tomcat_logs	作用：Tomcat 通用日志位置 是否必须拷贝：用户自行决定，Tomcat 历史访问日志，可不拷贝
%Tomcat_HOME%/webapps/webroot/bi-data	每个工程节点%BI_HOME%/fanruanxxx/bi6/bi-data	作用：FineBI 抽取数据 是否必须拷贝：用户自行决定 如果不拷贝, 在新工程启动后需要全局更新，重新抽取数据 如果拷贝，该文件夹中文件数量很多，拷贝耗时很久，请耐心等待
%Tomcat_HOME%/webapps/webroot/logs	每个工程节点%BI_HOME%/fanruanxxx/bi6/logs	作用：swift 日志 是否必须拷贝：用户自行决定 如果不拷贝, 会丢失工程历史操作日志（即 logdb）, 「管理系统>平台日志」功能无数据 如对历史使用情况无要求，无需拷贝
%Tomcat_HOME%/webapps/webroot/WEB-INF/classes	每个工程节点%BI_HOME%/fanruanxxx/bi6/c	作用：工程调用的默认和自定义 class

	lasses	<p>文件</p> <p>是否必须拷贝：必须拷贝</p> <p>有可能存在自定义的 class 文件，如不拷贝会影响工程正常使用</p>
%Tomcat_HOME%/webapps/webroot/WEB-INF/config	每个工程节点%BI_HOME%/fanruanxxx/bi6/config	<p>作用：配置库相关文件</p> <p>是否必须拷贝：必须拷贝</p> <p>平台调用的配置库 (finedb) 存储信息，如不拷贝会影响工程正常使用</p> <p>建议仅覆盖更新, 不要全量删除再拷贝进来，以防丢失必要文件</p>
<p>找到以下两个目录：</p> <p>%Tomcat_HOME%/webapps/webroot/WEB-INF/lib</p> <p>%Tomcat_HOME%/webapps/webroot/WEB-INF/customLib</p> <p>以下为工程自带 JAR，无需拷贝，除此之外的用户自定义 JAR 均需拷贝：</p> <p>点击展开更多</p> <p>fdl-bi-extension-4.0.jar</p> <p>fdl-boot-4.0.jar</p> <p>fdl-core-4.0.jar</p> <p>fdl-cron-4.0.jar</p> <p>fdl-datasource-4.0.jar</p> <p>fdl-third-4.0.jar</p> <p>fine-accumulator-11.0.jar</p> <p>fine-activator-11.0.jar</p> <p>fine-autuator-formula-11.0.jar</p> <p>fine-bi-adapter-6.0.jar</p> <p>fine-bi-datamining-6.0.jar</p> <p>fine-bi-datamining-third-6.0.jar</p> <p>fine-bi-engine-spider-6.0.jar</p> <p>fine-bi-engine-third-6.0.jar</p> <p>fine-bi-foundation-6.0.jar</p> <p>fine-bi-middle-6.0.jar</p>	每个工程节点%BI_HOME%/fanruanxxx/bi6/customlib	<p>作用：工程定制/外部引入的 JAR 包</p> <p>是否必须拷贝：必须拷贝，如不拷贝会影响模板访问</p>



<div>fine-bi-query-6.0.jar fine-bi-query-excel-6.0.jar fine-bi-query-third-6.0.jar fine-bi-scheduler-6.0.jar fine-bi-spider-cluster-6.0.jar fine-cbb-11.0.jar fine-core-11.0.jar fine-datasource-11.0.jar fine-decision-11.0.jar fine-decision-bi-11.0.jar fine-decision-report-11.0.jar fine-report-engine-11.0.jar fine-schedule-11.0.jar fine-schedule-bi-11.0.jar fine-schedule-report-11.0.jar fine-swift-log-adaptor-11.0.jar fine-third-11.0.jar fine-webui-11.0.jar h2-2.1.214.jar jtds-1.3.1.jar mssql-jdbc-9.4.1.jre8.jar mysql-connector-java-5.1.49-bin.jar ojdbc8.jar orai18n.jar sqlite-jdbc-3.35.4.jar jarfile-checksum.txt readme.txt version-bi.txt</div>		
<div>%Tomcat_HOME%/webapps/webroot/WEB-INF/plugins</div>	<div>每个工程节点%BI_HOME%/fanruanxxx/bi6/plugins</div>	<div>作用： 插件相关文件 是否必须拷贝： 必须拷贝， 如不拷贝会影响插件相关功能的实现</div>
<div>%Tomcat_HOME%/webapps/webroot/backup</div>	<div>minio 文件服务器文件存放路径/backup</div>	<div>作用： 工程历史备份文件 是否必须拷贝： 用户自行决定， 可存储在原始的目录下备份， 可不拷贝</div>
<div>%Tomcat_HOME%/webapps/webroot/WEB-INF/assets/temp_attach</div>	<div>minio 文件服务器文件存放路径/assets/temp_attach</div>	<div>作用： FineBI 数据表相关信息 是否必须拷贝： 必须拷贝 该文件夹中存放着工程 Excel 原始文件信息, 如不拷贝该文件夹会导致原始</div>

		Excel 丢失
%Tomcat_HOME%/webapps/webroot/WEB-INF/assets/vcs	minio 文件服务器文件存放路径/assets/vcs	作用：FineReport 模板备份文件  是否必须拷贝：用户自行决定  如果工程未使用 FineReport 模板，或不需要回退历史开发的 FineReport 模板，则无需拷贝
%Tomcat_HOME%/webapps/webroot/WEB-INF/assets/其他文件	minio 文件服务器文件存放路径/assets/其他文件	作用：通用的共享持久化目录  是否必须拷贝：必须拷贝  工程正常运行所需要的文件, 如不拷贝会影响工程正常使用
%Tomcat_HOME%/webapps/webroot/WEB-INF/reportlets	minio 文件服务器文件存放路径/reportlets	作用：FineReport 模板存放目录  是否必须拷贝：用户自行决定  如果工程未使用 FineReport 模板，则无需拷贝
%Tomcat_HOME%/webapps/webroot/WEB-INF/schedule	minio 文件服务器文件存放路径/schedule	作用：定时调度生成的文件  是否必须拷贝：用户自行决定  如果不拷贝, 定时任务挂载到决策平台的结果报表无法访问

• **FineReport 需要拷贝的文件**

原非容器化工程目录	新容器化工程外挂目录	说明
%Tomcat_HOME%/logs	每个工程节点%FR_HOME%/fanruanxxx/fr/tomcat_logs	作用：Tomcat 通用日志位置  是否必须拷贝：用户自行决定，Tomcat 历史访问日志，可不拷贝
%Tomcat_HOME%/webapps/webroot/logs	每个工程节点%FR_HOME%/fanruanxxx/fr/logs	作用：swift 日志  是否必须拷贝：用户自行决定  如果不拷贝，会丢失工程历史操作日志（即 logdb），「管理系统>平台日志」

		<p>功能无数据</p> <p>如对历史使用情况无要求，无需拷贝</p>
%Tomcat_HOME%/webapps/webroot/WEB-INF/classes	每个工程节点%FR_HOME%/fanruanxxx/fr/classes	<p>作用：工程调用的默认和自定义 class 文件</p> <p>是否必须拷贝：必须拷贝</p> <p>有可能存在自定义的 class 文件，如不拷贝会影响工程正常使用</p>
%Tomcat_HOME%/webapps/webroot/WEB-INF/config	每个工程节点%FR_HOME%/fanruanxxx/fr/config	<p>作用：配置库相关文件</p> <p>是否必须拷贝：必须拷贝</p> <p>平台调用的配置库 (finedb) 存储信息，如不拷贝会影响工程正常使用</p> <p>建议仅覆盖更新，不要全量删除再拷贝进来，以防丢失必要文件</p>
<p>找到以下两个目录：</p> <p>%Tomcat_HOME%/webapps/webroot/WEB-INF/lib</p> <p>%Tomcat_HOME%/webapps/webroot/WEB-INF/customLib</p> <p>以下为工程自带 JAR，无需拷贝，除此之外的用户自定义 JAR 均需拷贝：</p> <p>点击展开更多</p> <p>fine-accumulator-11.0.jar</p> <p>fine-activator-11.0.jar</p> <p>fine-cbb-11.0.jar</p> <p>fine-core-11.0.jar</p> <p>fine-datasource-11.0.jar</p> <p>fine-decision-11.0.jar</p> <p>fine-decision-report-11.0.jar</p> <p>fine-report-engine-11.0.jar</p> <p>fine-schedule-11.0.jar</p> <p>fine-schedule-report-11.0.jar</p> <p>fine-service-management-2.0.7.jar</p>	每个工程节点%FR_HOME%/fanruanxxx/fr/customlib	<p>作用：工程定制/外部引入的 JAR 包</p> <p>是否必须拷贝：必须拷贝，如不拷贝会影响模板访问</p>

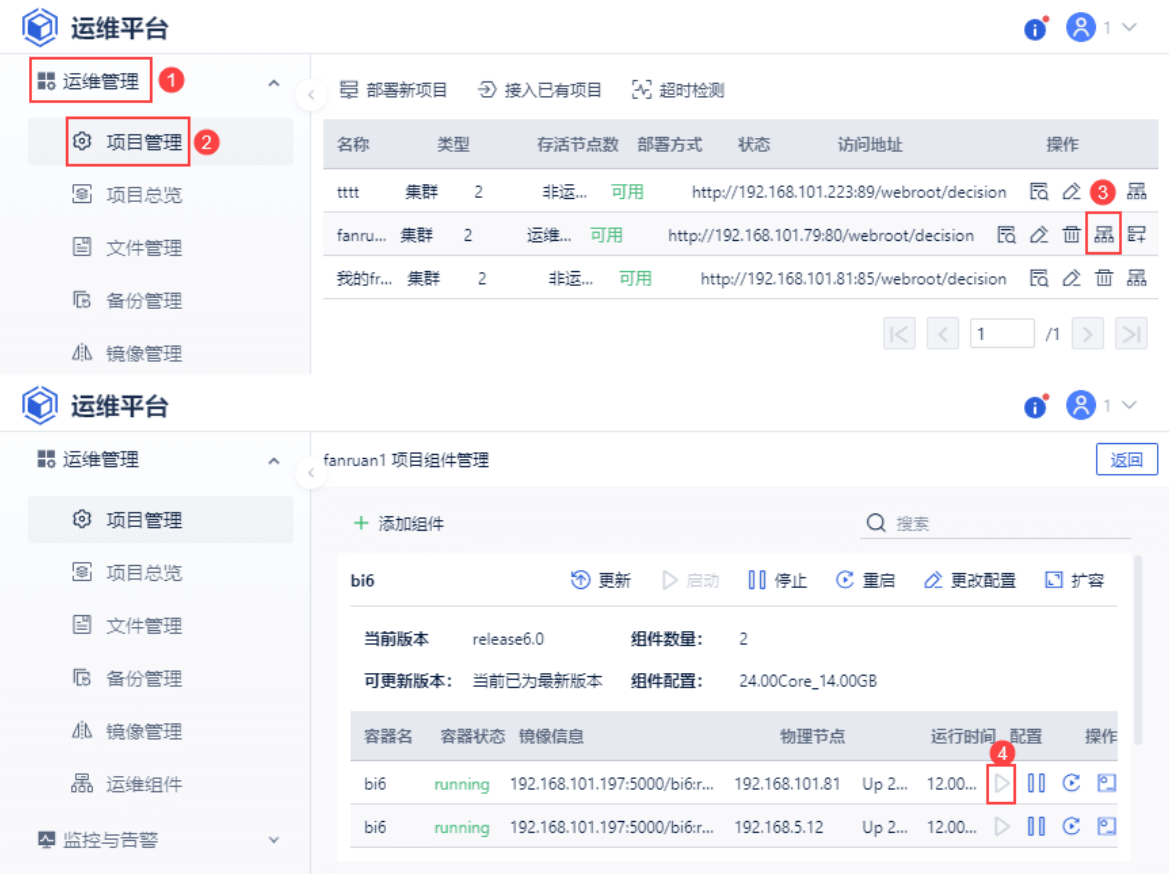
<div><div>fine-swift-log-adaptor-11.0.jar</div><div>fine-third-11.0.jar</div><div>fine-webui-11.0.jar</div><div>ifxjdbc_informix.jar</div><div>jtds-1.3.1.jar</div><div>mysql-connector-java-5.1.49-bin.jar</div><div>ojdbc8.jar</div><div>orai18n.jar</div><div>sqlite-jdbc-3.39.4.0.jar</div><div>sqljdbc.jar</div><div>sybase.jar</div><div>jarfile-checksum.txt</div><div>readme.txt</div></div>		
<div>%Tomcat_HOME%/webapps/webroot/WEB-INF/plugins</div>	<div>每个工程节点%FR_HOME%/fanruanxxx/fr/plugins</div>	<div>作用： 插件相关文件</div> <div>是否必须拷贝：必须拷贝，如不拷贝会影响插件相关功能的实现</div>
<div>%Tomcat_HOME%/webapps/webroot/backup</div>	<div>minio 文件服务器文件存放路径/backup</div>	<div>作用： 工程历史备份文件</div> <div>是否必须拷贝：用户自行决定，可存储在原始的目录下备份，可不拷贝</div>
<div>%Tomcat_HOME%/webapps/webroot/WEB-INF/assets/temp_attach</div>	<div>minio 文件服务器文件存放路径/assets/temp_attach</div>	<div>作用： 读写缓存存储路径</div> <div>是否必须拷贝： 必须拷贝</div> <div>该文件夹中存放着工程读写（图片）缓存，不拷贝可能会导致模板设置的背景图片预览为空</div>
<div>%Tomcat_HOME%/webapps/webroot/WEB-INF/assets/vcs</div>	<div>minio 文件服务器文件存放路径/assets/vcs</div>	<div>作用： FineReport 模板备份文件</div> <div>是否必须拷贝： 用户自行决定</div> <div>如果工程不需要回退历史开发的 FineReport 模板，则无需拷贝</div>
<div>%Tomcat_HOME%/webapps/webroot/WEB-INF/assets/其他文件</div>	<div>minio 文件服务器文件存放路径/assets/其他文件</div>	<div>作用： 通用的共享持久化目录</div> <div>是否必须拷贝： 必须拷贝</div> <div>工程正常运行所需要的文件，如不拷贝会影响工程正常使用</div>

%Tomcat_HOME%/webapps/webroot/WEB-INF/reportlets	minio 文件服务器文件存放路径/reportlets	作用：FineReport 模板存放目录 是否必须拷贝：必须拷贝，如不拷贝会导致工程所有模板丢失
%Tomcat_HOME%/webapps/webroot/WEB-INF/schedule	minio 文件服务器文件存放路径/schedule	作用：定时调度生成的文件 是否必须拷贝：用户自行决定 如果不拷贝，定时任务挂载到决策平台的结果报表无法访问

4.2 启动单个节点

用户登录运维平台，点击「运维管理>项目管理」，点击新容器化部署的工程项目后的「组件管理」按钮。

找到 bi6 组件，点击其中某个节点的「启动」按钮，即可启动该工程节点。如下图所示：



4.3 迁移新工程外接数据库

1) 新建空数据库

- 如果老工程外接数据库使用的非 MySQL8 类型数据库：

新工程外接数据库仍然部署在该数据库下的新库里，2.2 节的 MySQL 数据库

无需使用

请在老工程原外接数据库地址新建一个库名， 作为待迁移的空库

- 如果老工程外接数据库使用的 MySQL8 类型数据库：

新工程外接数据库建议部署在 2.2 节的 MySQL 数据库里

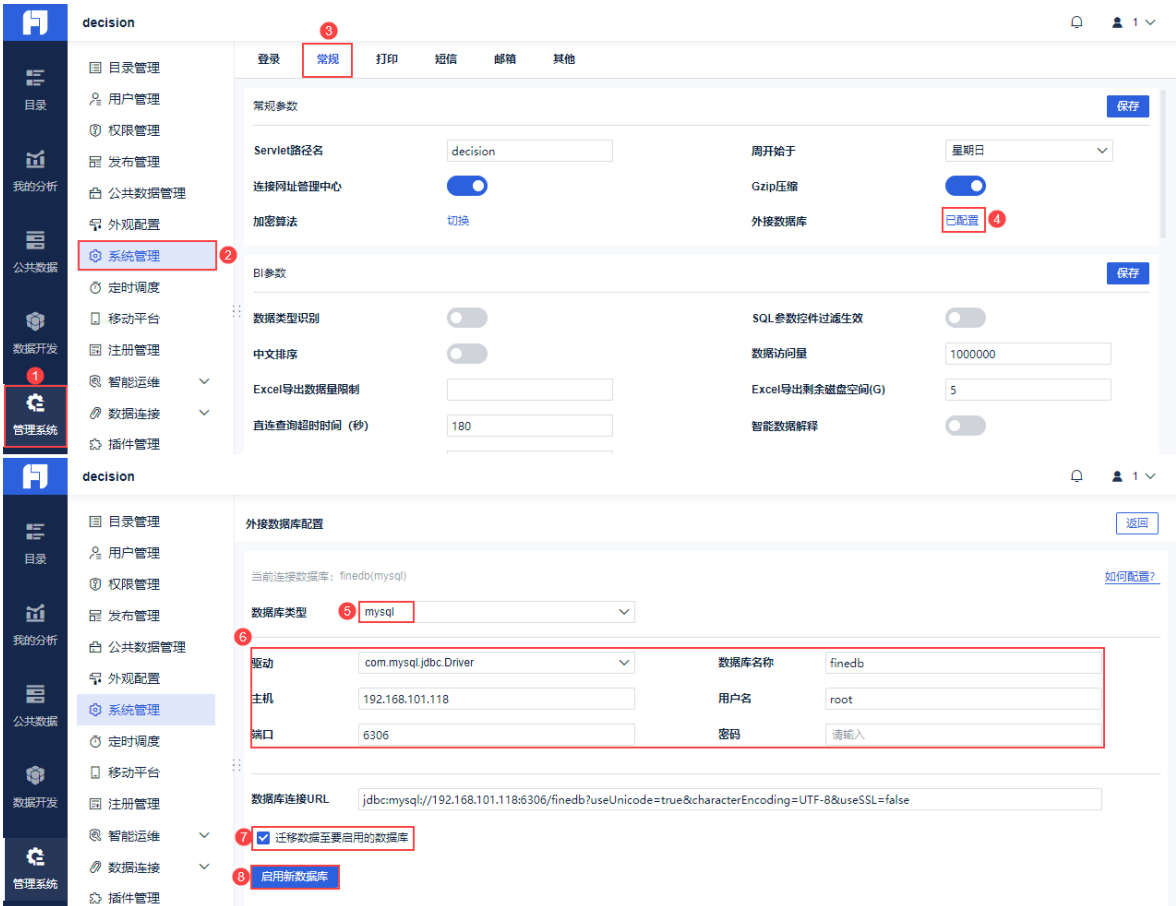
请在 2.2 节容器化部署的 MySQL8 组件新建一个库名， 作为待迁移的空库

2) 迁移数据库

管理员登录启动了的新容器化工程节点， 点击「管理系统>系统管理>常规>外接数据库」。

- 配置外接数据库为上文新建的空库
- 勾选「迁移数据至要启用的数据库」
- 点击「启用新数据库」

当出现提示「已成功切换至目标数据库!」， 则代表该节点的外接数据库切换成功。



4.4 修改抽取数据存储路径

1) 修改存储路径

在启动了的新容器化工程节点服务器中， 找到挂载目录 %BI\_HOME%/fanruanxxx/bi6/config

文件夹，修改/新建 spider.cluster.properties 文件。

在文件中添加一行：`spider_local_base_path=/usr/local/tomcat/webapps/webroot/bi-data/spider/db`

## 2) 删除死节点

管理员登录启动了的新容器化工程节点，点击「管理系统>系统管理>抽取集群」，对应节点状态变为了「down」，删除该节点。

## 3) 重启生成配置

用户登录运维平台，点击「运维管理>项目管理」，点击新容器化部署的工程项目后的「组件管理」按钮。

找到 bi6 组件，点击启动了的节点的「重启」按钮，即可重启该工程节点。

## 4) 抽取数据

如果 4.1 节未拷贝 bi-data 文件，工程启动后需要全局更新，重新抽取数据到该路径。

如果 4.1 节拷贝了 bi-data 文件，工程启动后将直接指向该路径。

## 4.5 关闭单个节点

用户登录运维平台，点击「运维管理>项目管理」，点击新容器化部署的工程项目后的「组件管理」按钮。

找到 bi6 组件，点击启动了的节点的「停止」按钮，即可关闭该工程节点。



#### 4.6 同步到其他节点

拷贝该工程节点%BI\_HOME%/fanruanxxx/bi6/config 文件夹到其他工程节点同位置。

#### 4.7 启动全部工程节点

用户登录运维平台，点击「运维管理>项目管理」，点击新容器化部署的工程项目后的「组件管理」按钮。

找到 bi6 组件，点击组件后的「启动」按钮，即可启动该集群所有工程节点。如下图所示：





#### 4.8 验证是否迁移成功

依次登录所有工程节点：

- 1) 查看「管理系统>智能运维>集群配置>节点管理」中是否出现所有集群节点，说明多节点集群全部配置成功
- 2) 查看各节点的工程配置是否相同，说明外接数据库配置成功



#### 5. 启动原非容器化工程

- 1) 将 3.1 节备份的 webroot 工程文件，覆盖原非容器化部署的工程的 webroot 工程
- 2) 参考 [关闭或重启 FineBI 工程](#) / [关闭或重启 FineReport 工程](#) 文档，启动原非容器化部署的工程。

## 6. 迁移授权

若老工程进行了注册，一般需要将授权迁移到新工程。容器化工程推荐使用 [容器私有云认证](#) 方式。

### 6.1 销毁原非容器化工程授权

- 1) 参考 [授权迁移插件](#) 的使用步骤，向商务索要一个授权迁移插件，销毁**原非容器化工程**的旧授权。
- 2) 找到**原非容器化工程**%BI\_HOME%/webroot/WEB-INF/resources 文件夹下的销毁凭证。

### 6.2 获取新容器化工程授权

参考 [容器私有云认证](#) 的使用步骤，向商务索要注册镜像文件 fanruan\_license\_server.tar，并参考文档为**新容器化工程**进行授权。

请在邮件中附上上一步获取的销毁凭证。并注明是「非容器化工程迁移容器化工程」

# 4.5.3 非容器化集群迁移容器化集群 (仅迁移工程)

## 1. 概述

### 1.1 版本

工程版本	功能变更
FineReport11.0	-
FineBI6.0	-

### 1.2 应用场景

相比于普通集群工程，容器化部署的工程，升级和维护成本较低。

本文方案帮助用户将原有非容器化部署的 FineBI 集群工程，迁移为容器化部署的集群工程。

只迁移 FineBI 工程，仍使用原有集群组件（状态服务器、文件服务器、外接数据库等）。

迁移后，原工程无法再启动使用。

- 注：全文以 FineBI 作为演示。FineReport 迁移步骤基本一致，仅需要两个点：
- 1) 拷贝文件时，FineReport 和 FineBI 需要拷贝的文件不完全相同，注意文档中的表格即可。
  - 2) 使用运维平台启停容器时，FineReport 关闭的是 fr 组件，FineBI 关闭的是 bi6 组件。
  - 3) 仅 FineBI 需要修改抽取数据存储路径，FineReport 无需执行本操作。

### 1.3 步骤简介

序号	说明
1 了解迁移步骤	请务必在执行迁移前，完整阅读本节步骤简介，了解每个步骤的作用
2 部署新容器化工程	容器化部署 FineReport/FineBI，需要通过 FineOps 运维平台前端实现因此用户需要先部署一个 FineOps 运维平台
2.1 部署新工程	迁移项目，是从原有非容器化工程，迁移到新的容器化工程因此用户需要先通过 FineOps 运维平台部署一个新的容器化集群工程
2.2 准备新工程	由于容器化部署的工程，一般为帆软发布的较新版本因此用户需要先确认新工程版本，确保后续升级旧工程到同一版本

3 原 非 容 器 化 工 程 准 备	程 版 本	
	2.4	
	备 份	容器化部署新工程时，会自动生成工程部署所用语句，后续运营维护时可能会需要使用/查
	部 署	迁移过程中该语句所在文件会被覆盖
	信 息	因此用户需要提前异地备份工程部署信息
	2.5	
	关 闭 新 工 程	迁移时，需要将老工程的一些文件拷贝到新工程中，这个操作需要在新工程停止运行的状态
		因此用户要通过 FineOps 运维平台前端快速关闭新工程
	3.1	
	备 份 原 工 程	在进行工程迁移前，为避免迁移失败，导致工程文件丢失无法回退等问题，请务必对原工程
4 迁 移		份后再进行后续操作
		为了防止迁移失败，建议记录下工程的配置信息，用于手动对接外接数据库组件和集群组
	3.2	
	升 级	工程迁移必须基于同版本的工程进行
	原 工 程	因此需要将原非容器化部署的工程升级到新工程同一版本，方可进行迁移
	3.3	
	销 毁 授 权	由于迁移后原工程无法再启动使用，因此如需迁移原工程授权，需要提前销毁授权
	3.4	
	关 闭 原 工 程	迁移时，需要将原工程的一些文件拷贝到新工程中，这个操作需要在原工程停止运行的状态
		因此用户需要手动关闭原工程
	3.5	
	修 改 抽 取 数 据 存 储 路 径	仅 FineBI 工程需要执行本节操作
		工程迁移后，FineBI 一般需要重新进行数据抽取，业务用户方可正常使用
		因此用户需要在抽取前确认并修改数据存储路径，再执行数据抽取
	4.1	
	本 节 即 为 正 式 迁 移 操 作	本节即为正式迁移操作
	4 迁 移	将原非容器化工程的必要文件，拷贝到新容器化工程中
	拷 贝	

移	文	
工	件	
	4.2	
程	启动	接下来的操作需要在新工程平台上执行
	新工程	因此用户要通过 FineOps 运维平台前端快速启动新工程
	4.3	
	验证	
	是否	通过查看新工程的平台配置，确保集群迁移成功
	迁移成功	
5	迁移授权	将老工程的授权迁移到新工程
		容器化工程推荐使用「容器私有云认证」方式

2. 新容器化工程准备

注：部署运维平台、新工程的服务器，老工程的集群组件、外接数据库的服务器，必须内网互通，会大幅降低迁移难度。

2.1 部署运维平台

详情请参见：[部署运维平台](#)

2.2 部署新工程

详情请参见：[部署新 FineReport 集群项目](#)、[部署新 FineBI 集群项目](#)

注意事项 1：迁移时，需要将老工程的一些必要文件、资源拷贝到新工程中，因此必须要确保新工程磁盘空间足够大。

- 1) 检查原非容器化工程%Tomcat\_home%\webapps\webroot 文件夹的大小。假设为 xG。
- 2) 在准备新容器化集群文件服务器时，请根据原工程大小，适度扩大新工程文件服务器所在节点剩余磁盘空间。

建议最小剩余磁盘空间大小为 max(2x , 500) G。

注意事项 2：部署新工程时，会同时部署一些必要组件，建议记录下组件的信息，方便在后续维护过程中查看。

请在「部署新项目-项目设置」环节，记录下 MySQL、Redis、minio、elasticsearch 组件的相关信息。由于组件密码随机生成，请务必修改组件的密码，方便后续使用。

部署新项目

1 项目设置

2 节点配置

项目名称

fanruan1

应用类型

☐ FineReport

☒ FineBI

☐ FineDataLink

应用类型

☒ 集群

☐ 单机

默认部署组件

☒ 运维基础套件

☒ Nginx

☒ Elasticsearch

自定义部署组件

☒ 文件服务器(MinIO)

☒ 外接数据库(MySQL)

☒ 状态服务器

部署配置

☐ 使用默认配置

☒ 自定义端口、组件用户名密码

访问路径

webroot

配置后应用的访问链接为(域名/IP)/(访问路径)/decision

部署配置自定义

主机端口

50101

→

容器端口

50101

主机端口

50200

→

容器端口

50200

redis

主机端口

6379

→

容器端口

6379

密码

\*\*\*\*\*

redis\_exporter

主机端口

9121

→

容器端口

9121

mysql

主机端口

3306

→

容器端口

3306

用户名

root

密码

\*\*\*\*\*

去设置

取消

完成

取消

下一步

**注意事项 3:** 迁移时, 需要将老工程的一些文件拷贝到新工程中, 因此需要记录下新工程存放文件的外挂目录位置。

请在「部署新项目-节点配置」环节，记录下集群每个工程节点的挂载路径。

The image shows a '节点设置' (Node Settings) dialog box in a deployment tool. The dialog is titled '节点设置' and has a close button 'X'. It contains the following fields and options:

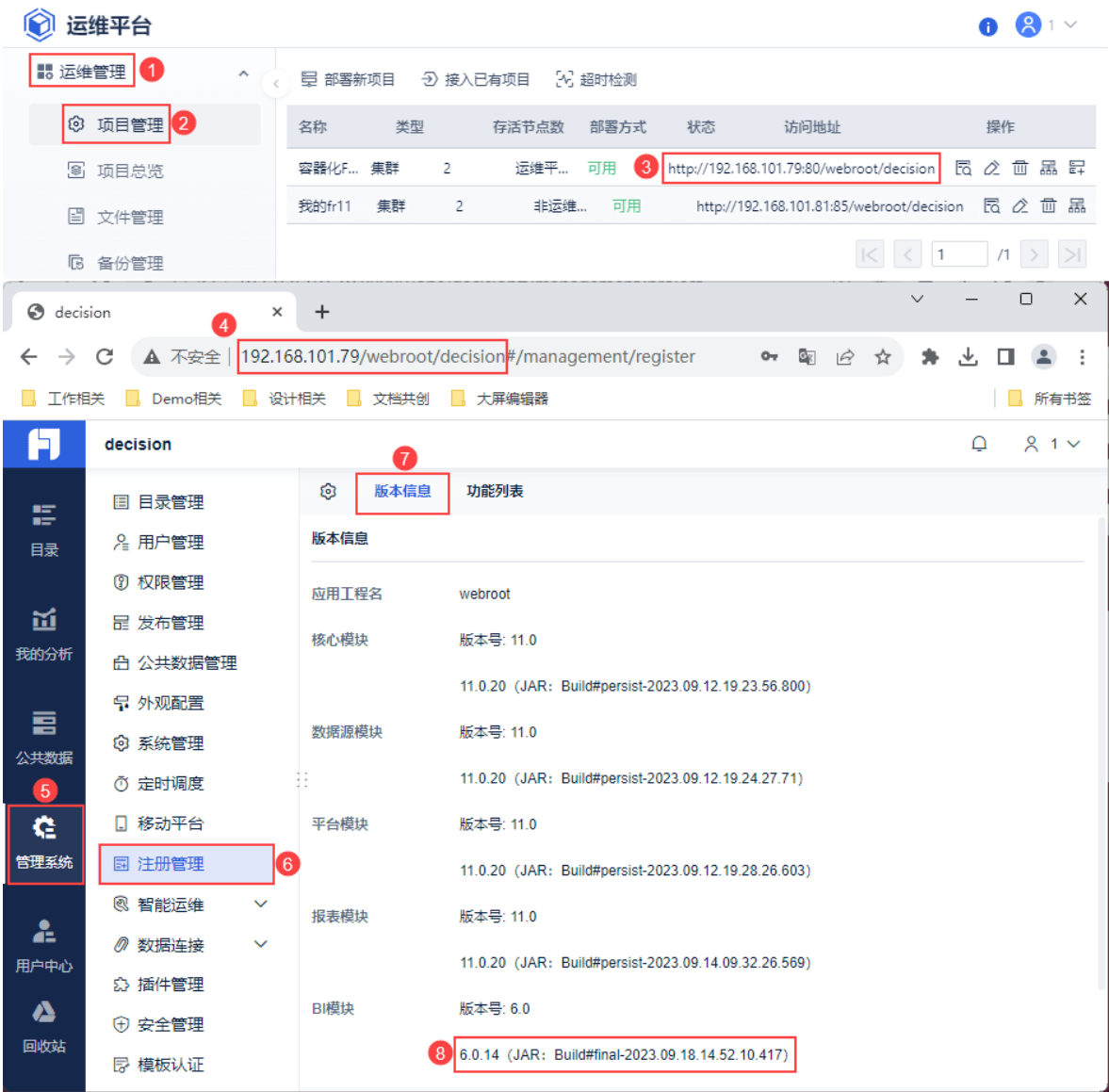
- 节点类型** (Node Type): Radio buttons for '主应用' (Main Application) and '组件' (Component). '主应用' is selected.
- 提示** (Hint): 主应用节点要求最低配置8核16G磁盘剩余50GB以上
- 主机** (Host): 192.168.101.197
- 端口** (Port): 22
- 用户名** (Username): root
- 验证方法** (Authentication Method): 密码 (Password)
- 密码** (Password): [masked]
- 挂载路径** (Mount Path): /home/fanruan

In the background, the '部署新项目' (Deploy New Project) workflow is visible, showing steps '项目设置' (Project Settings) and '节点配置' (Node Configuration). A red box highlights the '新增节点' (Add Node) button in the '节点配置' step, and another red box highlights the '挂载路径' (Mount Path) field in the dialog.

### 2.3 确认新工程版本

1) 管理员登录 FineOps 运维平台，在「运维管理>项目管理」中，可查看新工程的访问地址。

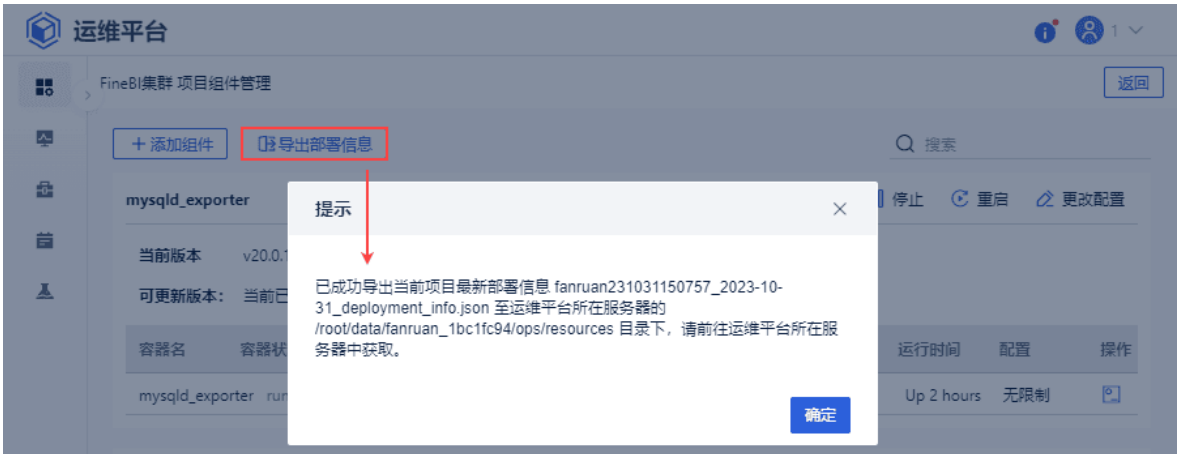
2) 管理员登录新容器化工程，点击「管理系统>注册管理>版本信息」，可查看新容器化工程的小版本号（精确到 JAR 包日期）。



## 2.4 备份部署信息

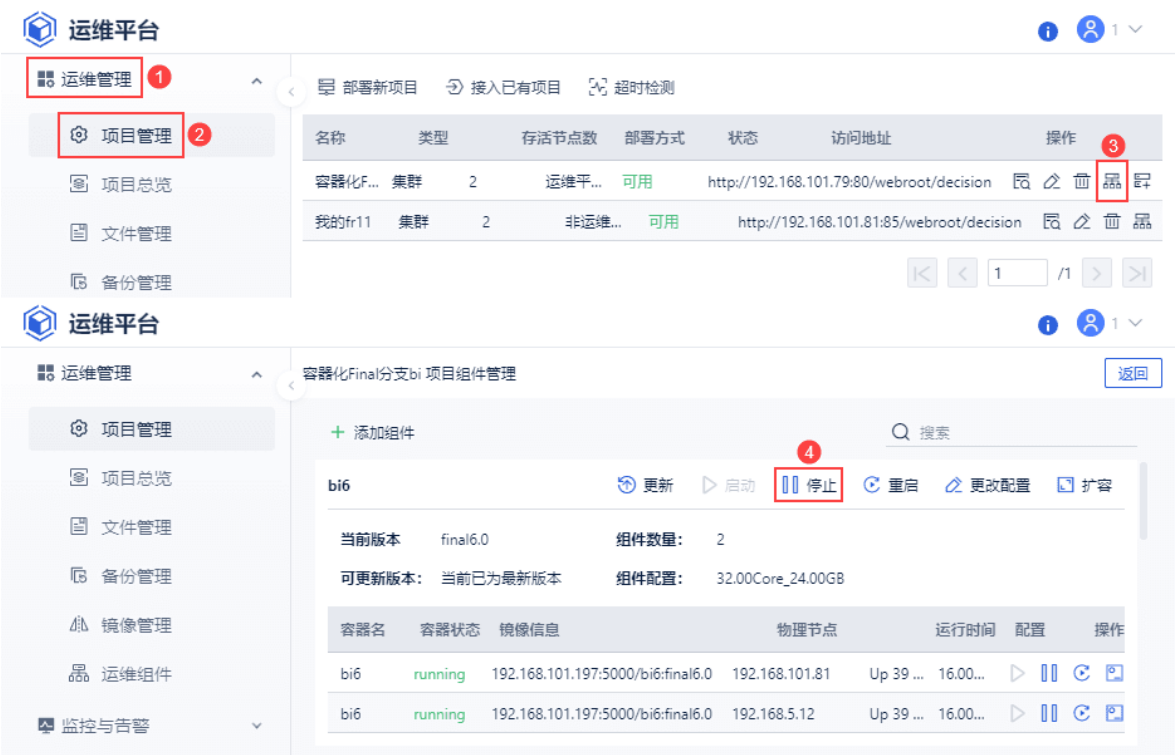
对于运维平台部署的组件/项目，在后期运维时，可能需要查阅原始部署信息。

- 1) 管理员登录 FineOps 运维平台，在「运维管理>项目管理」中，点击新容器化部署的项目后的「组件管理」按钮。
- 2) 点击「导出部署信息」，导出 json 文件。
- 3) 根据提示信息，请前往运维平台所在服务器中获取部署信息，异地备份。



2.5 关闭新工程

- 1) 管理员登录 FineOps 运维平台，在「运维管理>项目管理」中，点击新容器化部署的项目后的「组件管理」按钮。
- 2) 找到 bi6/fr 组件，点击「停止」按钮，即可关闭新容器化工程。



3. 原非容器化工程准备

3.1 备份原工程

1) 工程整体备份

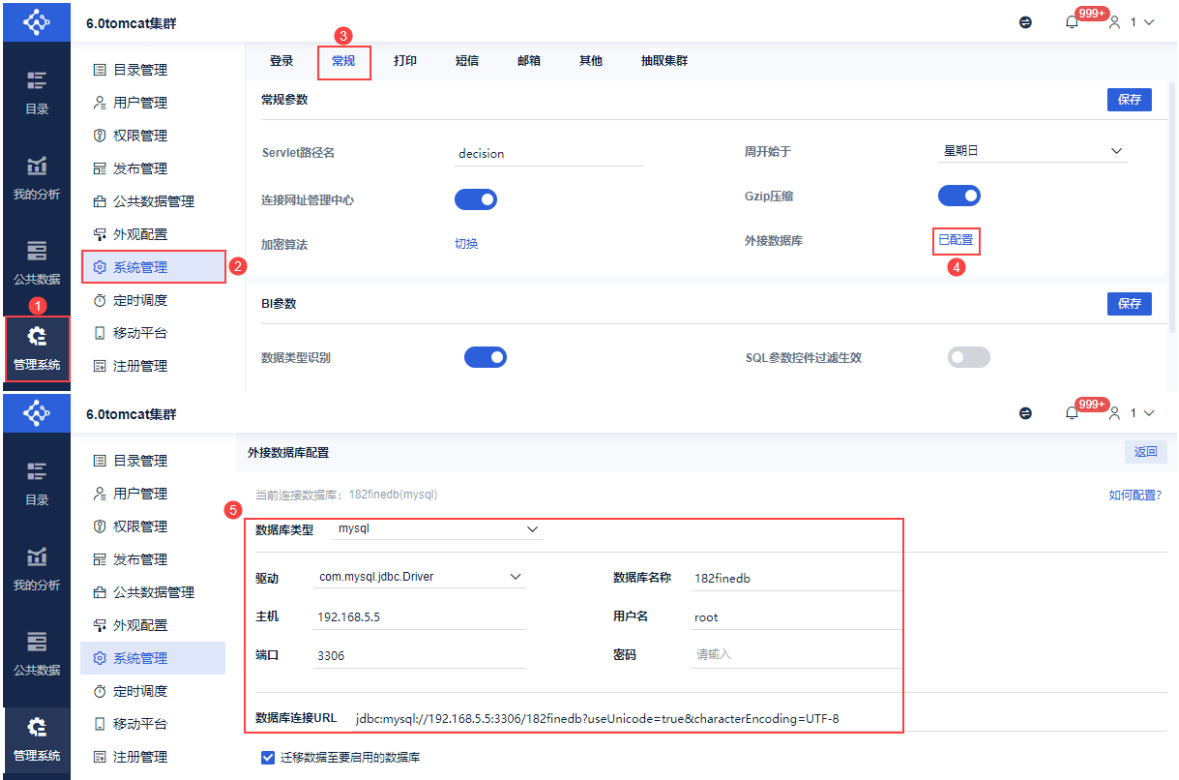
详情请参见 [工程备份还原方案](#) 3.1 节，对工程进行备份。

2) 记录原工程外接库信息

管理员登录原工程，点击「管理系统>系统管理>常规>外接数据库」。

记录下外接数据库的相关信息。确保外接数据库与待安装的新工程环境互通。





### 3) 记录原工程集群信息

管理员登录原工程， 点击「管理系统>智能运维>集群配置」。

记录下状态服务器和文件服务器的相关信息。确保状态服务器、文件服务器与待安装的新工程环境互通。



### 3.2 升级原工程

#### 1) 获取 JAR 包

付费用户请联系帆软技术支持，获取 2.3 节新工程同一小版本的工程 JAR 包（精确到 JAR 包日期）。

技术支持联系方式：前往「[服务](#)」，选择「在线支持」、电话「400-811-8890」。

#### 2) 升级原非容器化工程

请参考「[FineBI 小版本升级](#) / [FineReport 小版本升级](#)」文档升级工程至指定版本。

### 3.3 销毁授权

- 1) 参考 [授权迁移插件](#) 的使用步骤，向商务索要一个授权迁移插件，销毁原非容器化工程的旧授权。
- 2) 找到原非容器化工程%BI\_HOME%/webroot/WEB-INF/resources 文件夹下的销毁凭证并存储好。

### 3.4 关闭原工程

请参考「 [关闭或重启 FineBI 工程](#) / [关闭或重启 FineReport 工程](#) 」文档，关闭原非容器化集群下的每个工程节点。

### 3.5 修改抽取数据存储路径

仅 FineBI 工程需要执行本节操作

在原工程任一节点服务器中，找到%BI\_HOME%/webroot/WEB-INF/config 文件夹，修改/新建 spider.cluster.properties 文件。

在文件中添加一行： spider\_local\_base\_path=/usr/local/tomcat/webapps/webroot/bi-data/spider/db

## 4. 迁移工程

### 4.1 拷贝工程文件

- 1) 请提前检查原非容器化部署的工程，是否配置了外挂目录、是否配置共享文件服务器，确保拷贝到正确的文件。
- 2) 新容器化工程外挂文件所在位置，为 2.2 节记录的挂载路径。
- 3) 将以下文件，从 3.5 节的工程节点，拷贝到新容器化部署的每个工程节点中：

注 1：请勿直接 docker cp 整个 webroot 文件夹到容器，会导致功能异常。  
注 2：请勿直接拷贝整个 webroot 文件夹到容器化工程的外挂目录，部分文件不可以被覆盖。  
注 3：请勿使用 3.1 节备份的文件，而是使用 3.5 节工程最终的文件进行拷贝操作。  
注 4：工程下的文件需要从 3.5 节原工程节点一一拷贝到新容器化部署的每一个工程节点中。

- **FineBI 需要拷贝的文件**

原非容器化工程目录	新容器化工程外挂目录	说明
%Tomcat_HOME%/logs	每个工程节点%BI_HOME%/fanruanxxx/bi6/tomcat_logs	作用：Tomcat 通用日志位置
		是否必须拷贝：用户自行决定，可不拷贝
%Tomcat_HOME%/webapps/webroot/bi-data	每个工程节点%BI_HOME%/fanruanxxx/bi6/bi-data	作用：FineBI 抽取数据
		是否必须拷贝：用户自行决定，如果不拷贝，在新工程启动后抽取数据
		如果拷贝，该文件夹中文件较多，请耐心等待

		作用：swift 日志
		是否必须拷贝：用户自行决定
%Tomcat_HOME%/webapps/webroot/logs	每个工程节点%BI_HOME%/fanruanxxx/bi6/logs	如果不拷贝，会丢失工程历史数据
		「管理系统>平台日志」功能中可查询历史数据
		如历史使用情况无要求，可不拷贝
		作用：工程调用的默认和自定义的 class 文件
%Tomcat_HOME%/webapps/webroot/WEB-INF/classes	每个工程节点%BI_HOME%/fanruanxxx/bi6/classes	是否必须拷贝：必须拷贝
		有可能存在自定义的 class 文件，如不拷贝，工程无法正常启动
		作用：配置库相关文件
		是否必须拷贝：必须拷贝
%Tomcat_HOME%/webapps/webroot/WEB-INF/config	每个工程节点%BI_HOME%/fanruanxxx/bi6/config	平台调用的配置库（finedb）文件
		影响工程正常使用
		建议仅覆盖更新，不要全量删除
		丢失必要文件

找到以下两个目录：

%Tomcat\_HOME%/webapps/webroot/WEB-INF/lib

%Tomcat\_HOME%/webapps/webroot/WEB-INF/customLib

以下为工程自带 JAR，无需拷贝，除此之外

的用户自定义 JAR 均需拷贝：

点击展开更多

- fdl-bi-extension-4.0.jar
- fdl-boot-4.0.jar
- fdl-core-4.0.jar
- fdl-cron-4.0.jar
- fdl-datasource-4.0.jar
- fdl-third-4.0.jar
- fine-accumulator-11.0.jar
- fine-activator-11.0.jar
- fine-autuator-formula-11.0.jar
- fine-bi-adapter-6.0.jar
- fine-bi-datamining-6.0.jar
- fine-bi-datamining-third-6.0.jar
- fine-bi-engine-spider-6.0.jar

每个工程节点%BI\_HOME%/fanruanxxx/bi6/customlib

作用：工程定制/外部引入的 jar 包

是否必须拷贝：必须拷贝，如不拷贝，工程无法正常启动

fine-bi-engine-third-6.0.jar  
fine-bi-foundation-6.0.jar  
fine-bi-middle-6.0.jar  
fine-bi-query-6.0.jar  
fine-bi-query-excel-6.0.jar  
fine-bi-query-third-6.0.jar  
fine-bi-scheduler-6.0.jar  
fine-bi-spider-cluster-6.0.jar  
fine-cbb-11.0.jar  
fine-core-11.0.jar  
fine-datasource-11.0.jar  
fine-decision-11.0.jar  
fine-decision-bi-11.0.jar  
fine-decision-report-11.0.jar  
fine-report-engine-11.0.jar  
fine-schedule-11.0.jar  
fine-schedule-bi-11.0.jar  
fine-schedule-report-11.0.jar  
fine-swift-log-adaptor-11.0.jar  
fine-third-11.0.jar  
fine-webui-11.0.jar  
h2-2.1.214.jar  
jtds-1.3.1.jar  
mssql-jdbc-9.4.1.jre8.jar  
mysql-connector-java-5.1.49-bin.jar  
ojdbc8.jar  
orai18n.jar  
sqlite-jdbc-3.35.4.jar  
jarfile-checksum.txt  
readme.txt  
version-bi.txt

作用：插件相关文件

%Tomcat_HOME%/webapps/webroot/WEB-INF/plugins	每个工程节点%BI_HOME%/fanruanxxx/bi6/plugins	是否必须拷贝：必须拷贝，如功能的实现
-----------------------------------------------	----------------------------------------	--------------------

• **FineReport 需要拷贝的文件**

原非容器化工程目录	新容器化工程外挂目录	说明
		作用：Tomcat 通用日志位置
%Tomcat_HOME%/logs	每个工程节点%FR_HOME%/fanruanxxx/fr/tomcat_logs	是否必须拷贝：用户自行决定，Tomcat 历史访问日志，可不拷贝
%Tomcat_HOME%/webapps/webroot/logs	每个工程节点%FR_HOME%/fanruanxxx/fr/logs	作用：swift 日志
		是否必须拷贝：用户自行决定

	如果不拷贝，会丢失工程历史操作日志 (即 logdb), 「管理系统>平台日志」功能无数据
	如对历史使用情况无要求，无需拷贝
	作用：工程调用的默认和自定义 class 文件
%Tomcat_HOME%/webapps/ webroot/WEB-INF/classes	每个工程节点%FR_HOME%/fanruanxxx/fr/classes 是否必须拷贝：必须拷贝
	有可能存在自定义的 class 文件，如不拷贝会影响工程正常使用
	作用：配置库相关文件
	是否必须拷贝：必须拷贝
%Tomcat_HOME%/webapps/ webroot/WEB-INF/config	每个工程节点%FR_HOME%/fanruanxxx/fr/config 平台调用的配置库（finedb）存储信息，如不拷贝会影响工程正常使用
	建议仅覆盖更新，不要全量删除再拷贝进来，以防丢失必要文件
找到以下两个目录：	
%Tomcat_HOME%/webapps/webroot/WEB-INF/lib %Tomcat_HOME%/webapps/webroot/WEB-INF/customLib	
除以下工程自带 JAR 之外的自定义定制 JAR，全部拷贝：	
每个工程节点%FR_HOME%/fanruanxxx/fr/customlib	作用：工程定制/外部引入的 JAR 包 是否必须拷贝：必须拷贝，如不拷贝会影响模板访问
点击展开更多	
fine-accumulator-11.0.jar fine-activator-11.0.jar fine-cbb-11.0.jar fine-core-11.0.jar fine-datasource-11.0.jar fine-decision-11.0.jar fine-decision-report-11.0.jar fine-report-engine-11.0.jar	

fine-schedule-11.0.jar  
fine-schedule-report-11.0.jar  
fine-service-management-2.0.7.jar  
fine-swift-log-adaptor-11.0.jar  
fine-third-11.0.jar  
fine-webui-11.0.jar  
ifxjdbc\_informix.jar  
jtds-1.3.1.jar  
mysql-connector-java-5.1.49-bin.jar  
ojdbc8.jar  
orai18n.jar  
sqlite-jdbc-3.39.4.0.jar  
sqljdbc.jar  
sybase.jar  
jarfile-checksum.txt  
readme.txt

作用：插件相关文件

%Tomcat\_HOME%/webapps/ 每个工程节点%FR\_HOME%webroot/WEB-INF/plugins E%/fanruanxxx/fr/plugins 是否必须拷贝：必须拷贝，如不拷贝会影响插件相关功能的实现

4.2 启动全部工程节点

用户登录运维平台，点击「运维管理>项目管理」，点击新容器化部署的工程项目后的「组件管理」按钮。

找到 bi6 组件，点击组件后的「启动」按钮，即可启动该集群所有工程节点。如下图所示：



### 4.3 验证是否迁移成功

依次登录所有工程节点：

- 1) 查看「管理系统>智能运维>集群配置>节点管理」中是否出现所有集群节点，说明多节点集群全部配置成功。
- 2) 查看各节点的工程配置是否相同，说明外接数据库配置成功。



### 5. 迁移授权

参考 [注册新项目](#) 的使用步骤，向商务索要注册镜像文件 fanruan\_license\_server.tar，并参考文档为**新容器化工程**进行授权。

请在邮件中附上 2.5 节获取的销毁凭证。并注明是「非容器化工程迁移容器化工程」。





## 4.5.4 非容器化集群迁移容器化集群(迁移工程和集群组件)

### 1. 概述

#### 1.1 版本

工程版本	功能变更
FineReport11.0	-
FineBI6.0	

#### 1.2 应用场景

相比于普通集群工程，容器化部署的工程，升级和维护成本较低。

本文方案帮助用户将原有非容器化部署的 FineBI 集群工程，迁移为容器化部署的集群工程。

迁移 FineBI 工程和集群组件（状态服务器、文件服务器、外接数据库等）。

迁移后，原工程可以再启动使用。

注：全文以 FineBI 作为演示。FineReport 迁移步骤基本一致，仅需要两个点：

- 1) 拷贝文件时，FineReport 和 FineBI 需要拷贝的文件不完全相同，注意文档中的表格即可。
- 2) 使用运维平台启停容器时，FineReport 关闭的是 fr 组件，FineBI 关闭的是 bi6 组件。

### 2. 新容器化工程准备

#### 2.1 部署运维平台

容器化 FineBI 工程，需要通过运维平台部署。因此用户需要先部署一个 FineOps 运维平台。

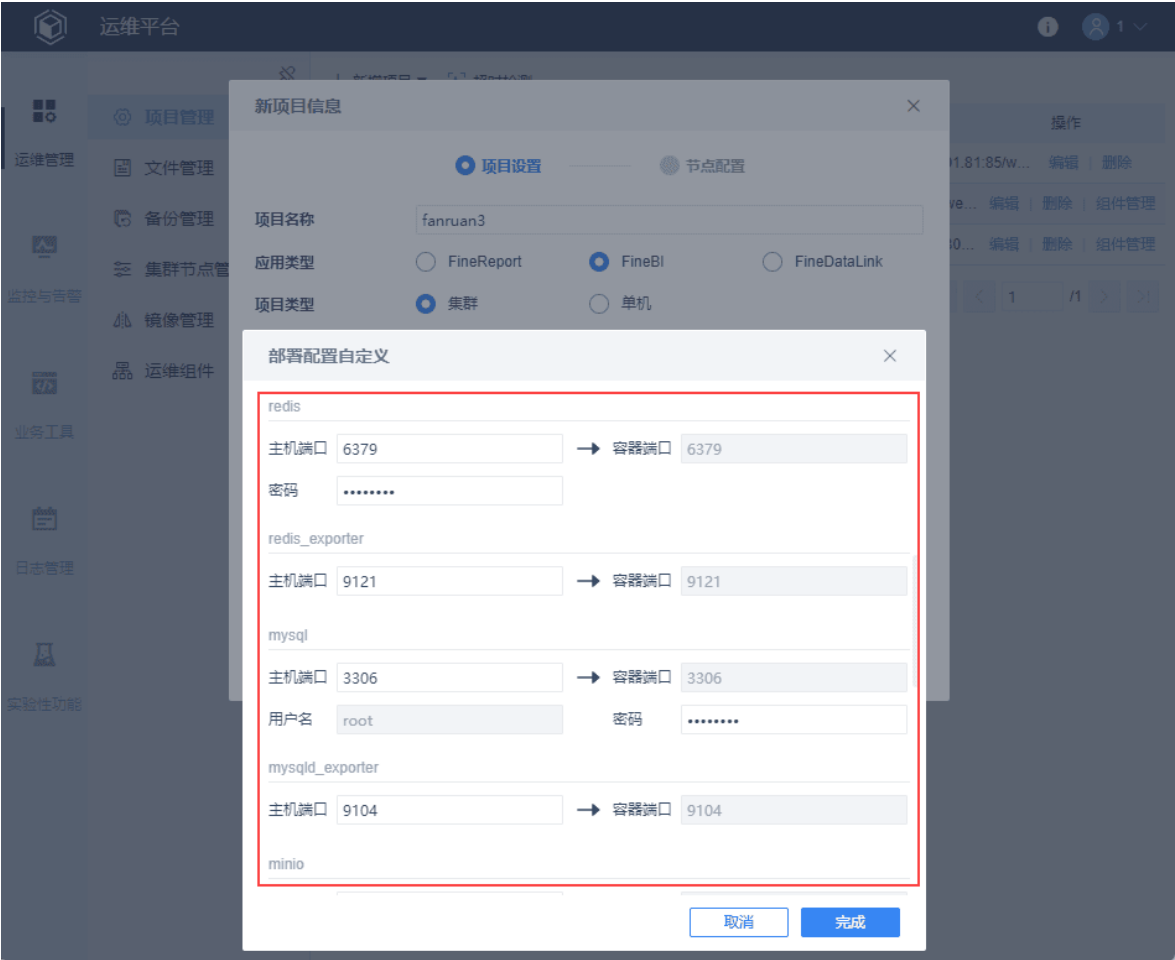
详情请参见：[部署运维平台](#)

#### 2.2 部署新容器化工程

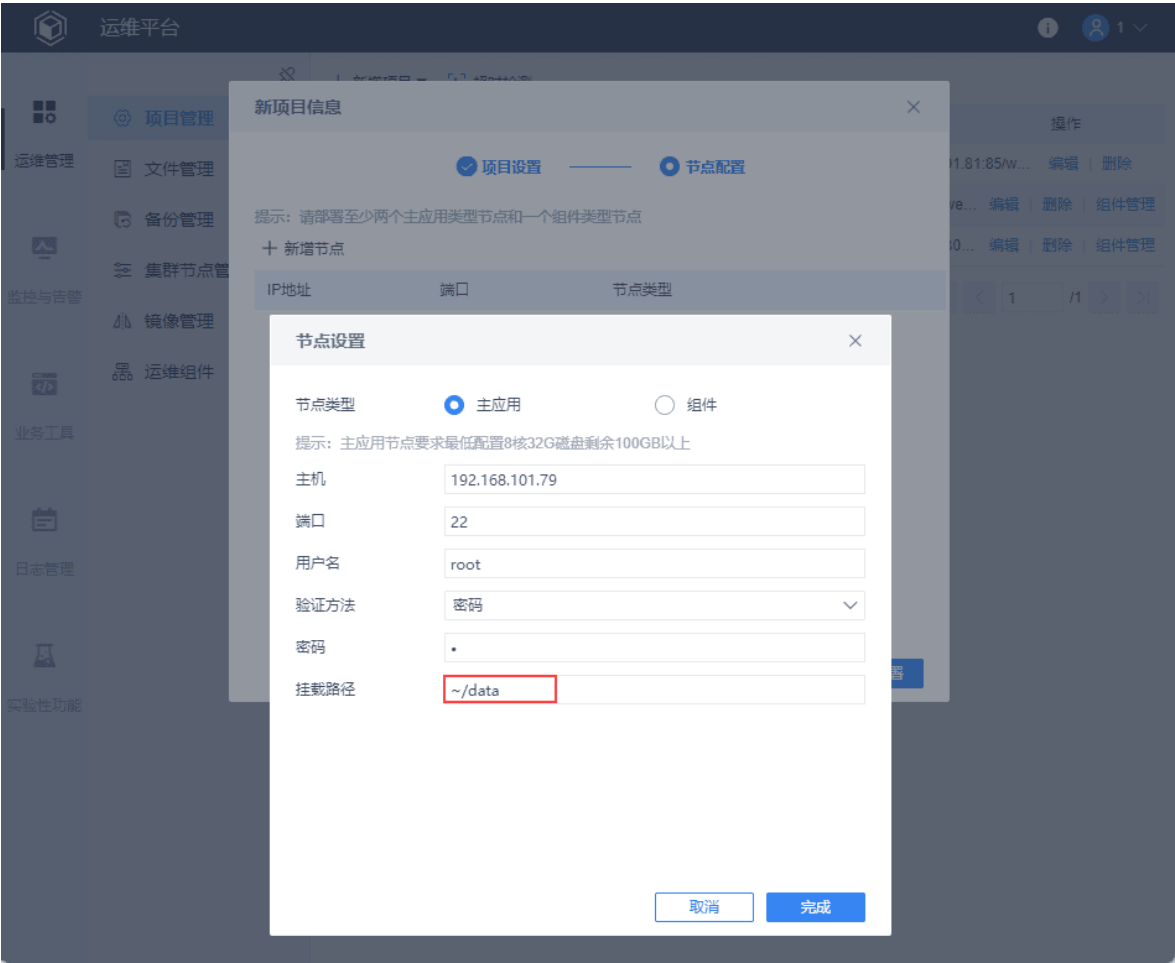
进行迁移前，用户需要先通过运维平台部署一个容器化集群工程。

详情请参见：[部署新项目-FineBI](#)、[部署新项目-FineReport](#)

注 1：请记录下 MySQL、Redis、minio 组件的相关信息，由于密码随机生成，请务必修改 MySQL、Redis、minio 组件的密码，方便后续使用。



注 2：请记录下 FineBI 工程的挂载路径，用户可自定义路径。这是工程文件的存放路径，后面步骤中需要用到。



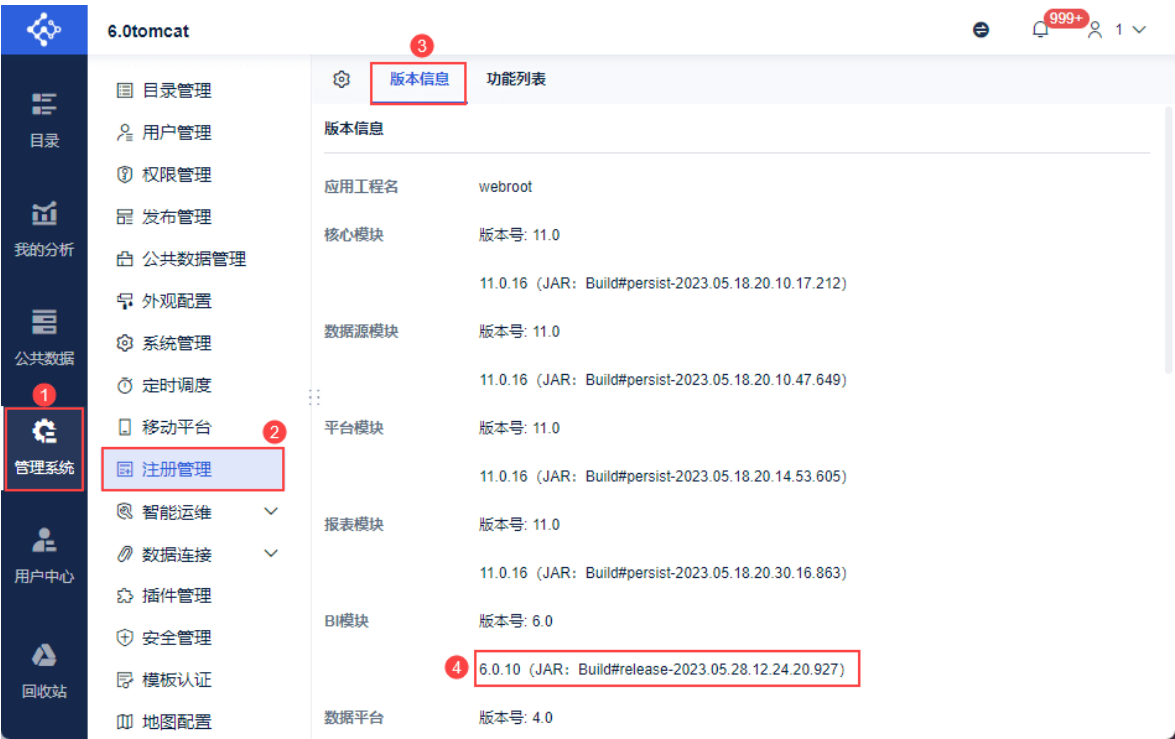
### 2.3 确认新工程服务器剩余磁盘空间

- 1) 检查原非容器化工程的大小。
- 2) 根据原工程大小，适度扩大新容器化工程所在服务器节点剩余磁盘空间，建议大于原工程的两倍，最少剩余磁盘大小 500G。

2.4 确认新工程版本

由于容器化部署的 FineBI 工程，一般为帆软发布的较新版本。因此用户需要先确认新工程版本，确保后续升级旧工程到同一版本。

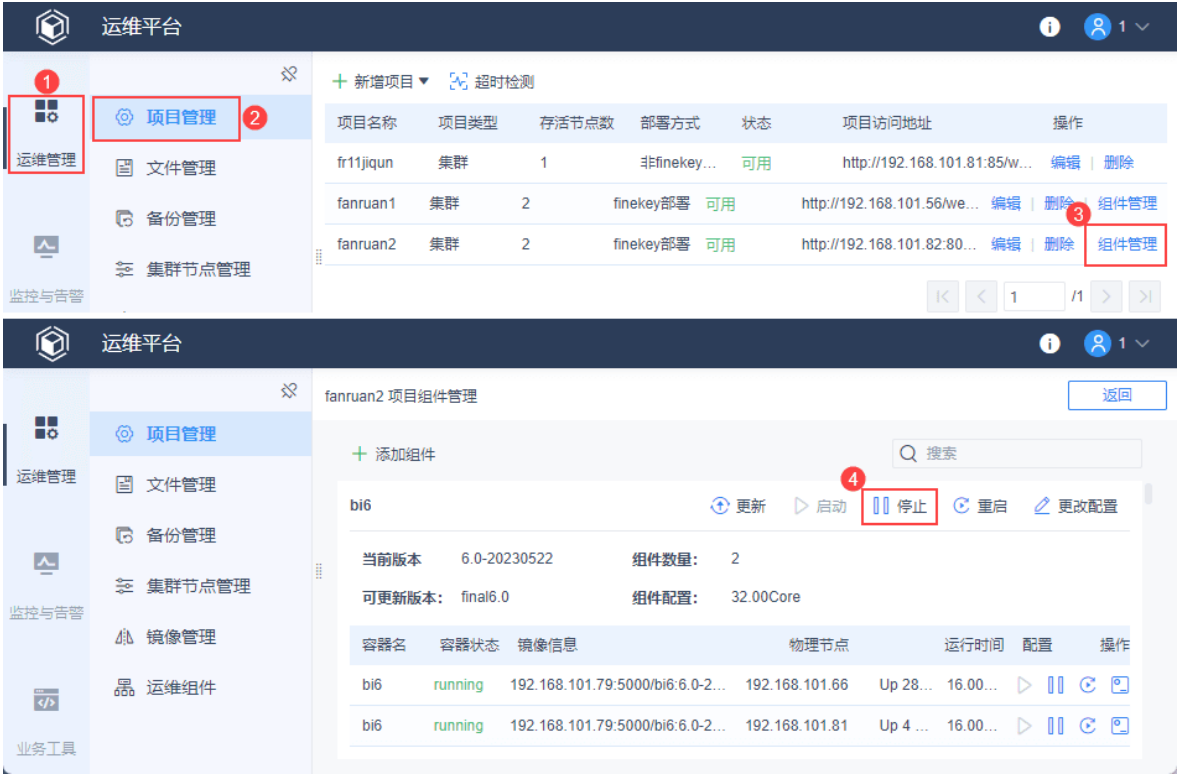
管理员登录新容器化工程，点击「管理系统>注册管理>版本信息」，查看新容器化工程的小版本号（精确到 JAR 包日期）。



2.5 关闭新容器化工程

用户登录运维平台，点击「运维管理>项目管理」，点击新容器化部署的工程项目后的「组件管理」按钮。

找到 bi6 组件，点击「停止」按钮，即可关闭新容器化集群的全部工程节点。如下图所示：



## 2.6 备份 config

请在 FineBI 工程的挂载路径/fanruanxxx/bi 目录下，找到 config 文件夹，并另行备份。

该文件夹中存放的 yaml 文件中，有着工程部署所用语句，后续运营维护时可能会需要使用/查找，请务必异地备份。

## 3. 原非容器化工程准备

### 3.1 备份原工程

在进行工程迁移前，为避免迁移失败，导致工程文件丢失无法回退等问题，请务必对原工程进行备份后再进行后续操作。

请参考 [工程备份还原方案](#) 3.1 节，对工程进行备份。

### 3.2 升级原工程

由于容器化部署的 FineBI 工程，一般为帆软发布的较新版本本。

因此用户需要将原非容器化部署的工程升级到同一新版本本，方可进行同版本间的迁移。

#### 1) 获取 JAR 包

付费用户请联系帆软技术支持，获取 2.4 节新工程同一小版本的工程 JAR 包（精确到 JAR 包日期）。

技术支持联系方式：前往「[服务](#)」，选择「在线支持」、电话「400-811-8890」。

#### 2) 升级原非容器化工程

请参考「[FineBI 小版本升级](#) / [FineReport 小版本升级](#)」文档升级工程至指定版本。

### 3.3 关闭其他工程节点

请参考 [关闭或重启 FineBI 工程](#) / [关闭或重启 FineReport 工程](#) 文档，关闭原非容器化部署的其他工程节点，只保留一个工程节点存活。

### 3.4 迁移外接数据库

为了保证新老工程的平台配置完全一致，需要同步两个工程的配置库。但不能让新老工程使用同一个外接数据库。

#### 1) 新建空数据库

- 如果老工程外接数据库使用的非 MySQL8 类型数据库：

新工程外接数据库仍然部署在该数据库下的新库里，2.2 节的 MySQL 数据库

无需使用

请在老工程原外接数据库地址新建一个库名，作为待迁移的空库

- 如果老工程外接数据库使用的 MySQL8 类型数据库：

新工程外接数据库建议部署在 2.2 节的 MySQL 数据库里

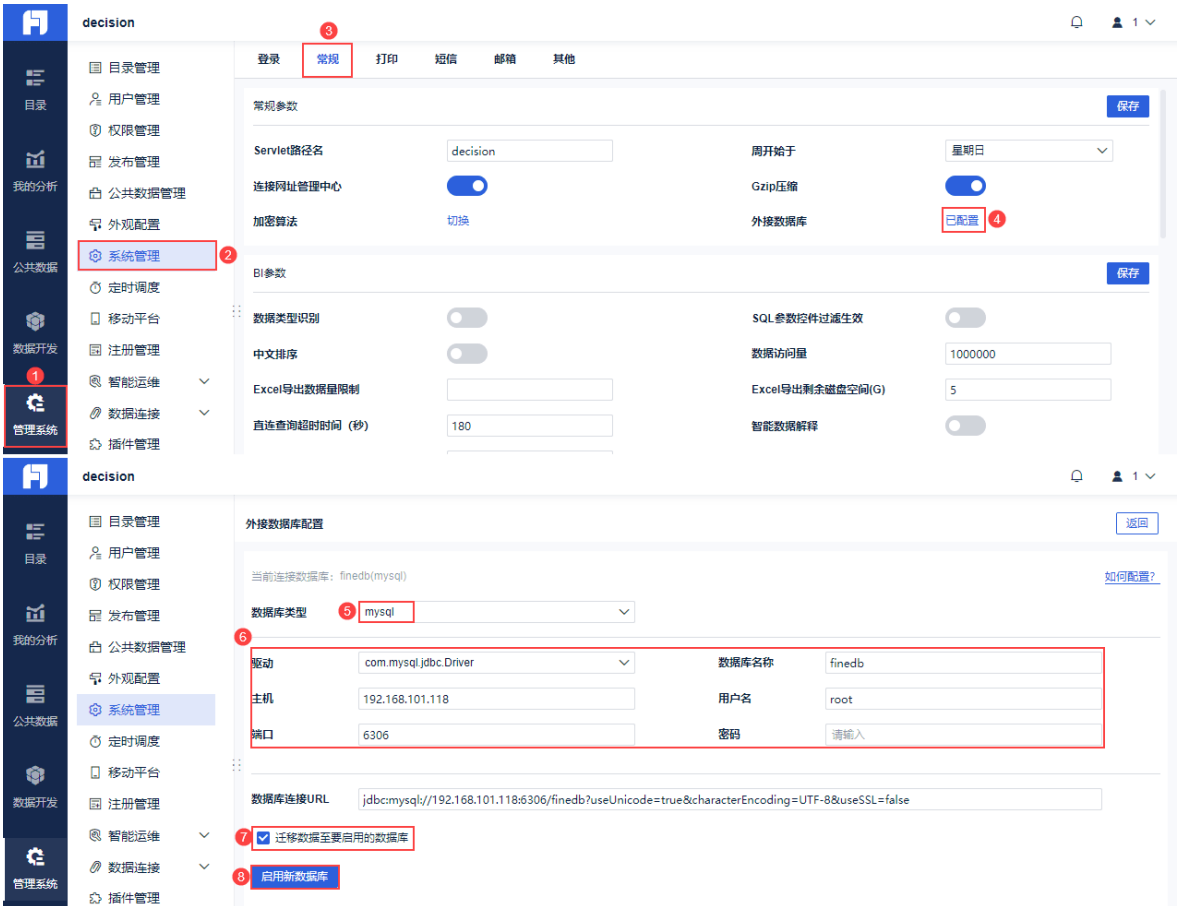
请在 2.2 节容器化部署的 MySQL8 组件新建一个库名，作为待迁移的空库

#### 2) 迁移数据库

管理员登录原工程唯一存活的节点，点击「管理系统>系统管理>常规>外接数据库」。

- 配置外接数据库为上一步的空库
- 勾选「迁移数据至要启用的数据库」
- 点击「启用新数据库」

当出现提示「已成功切换至目标数据库!」，则代表外接数据库切换成功。



### 3.5 安装 minio 资源仓库插件

由于需要让原非容器工程对接容器化部署的集群组件，因此需要先安装必要插件。

管理员登录原工程唯一存活的节点，点击「管理系统>插件管理」，安装「minio 资源仓库」插件。如下图所示：



### 3.6 对接集群组件

管理员登录原工程唯一存活的节点，点击「管理系统>智能运维>集群配置」。

- 1) 开启「状态服务器」。选择「单机 Redis」，输入 2.2 节记录的 Redis 组件信息。
- 2) 开启「文件服务器共享」。选择「MinIO」，输入 2.2 节记录的 MinIO 组件信息。

3) 缓存模式选择「关闭缓存」，集群通信协议选择「TCP」。

4) 开启「开启集群」按钮。

参考 [关闭或重启 FineBI 工程](#) 文档，重启该工程节点。

再次查看「管理系统>智能运维>集群配置」，查看集群配置是否已更新。



### 3.7 关闭原工程



请参考 [关闭或重启 FineBI 工程](#) / [关闭或重启 FineReport 工程](#) 文档，关闭原工程全部工程节点。

## 4. 迁移工程

### 4.1 拷贝文件

将原非容器化集群最后存活的那个工程节点和原文件服务器中的必要文件，拷贝到容器化集群的各个节点和 minio 服务器中。

- 1) 请提前检查原非容器化部署的集群，是否配置了外挂目录、是否配置共享文件服务器，确保拷贝到正确的文件。
- 2) 新容器化工程外挂文件所在位置，为 2.2 节记录的挂载路径。

注 1：请勿直接 docker cp 整个 webroot 文件夹到容器，会导致功能异常。

注 2：请勿直接拷贝整个 webroot 文件夹到容器化工程的外挂目录，部分文件不可以被覆盖。

注 2：请勿使用 3.1 节备份的文件，而是使用工程最终的文件进行拷贝操作。

- FineBI 需要拷贝的文件**

原非容器化工程目录	新容器化工程外挂目录	说明
%Tomcat_HOME%/logs	%BI_HOME%/fanruanxxx/bi6/tomcat_logs	作用：Tomcat 通用日志位置 是否必须拷贝：用户自行决定，Tomcat 历史访问日志，可不拷贝
%Tomcat_HOME%/webapps/webroot/bi-data	%BI_HOME%/fanruanxxx/bi6/bi-data	作用：FineBI 抽取数据 是否必须拷贝： 如果新老工程在同一服务器上，可重新-v 挂载原工程的 bi-data 文件夹，或在新工程启动后重新抽取数据，无需拷贝 如果新老工程不在同一服务器上，不建议拷贝，该文件夹一般数据量太大，拷贝困难，可在新工程启动后重新抽取数

		据
%Tomcat_HOME%/webapps/webroot/logs	%BI_HOME%/fanruanxxx/bi6/logs	<p>作用： swift 日志</p> <p>是否必须拷贝： 用户自行决定</p> <p>如果不拷贝， 会丢失工程历史操作日志（即 logdb），「管理系统&gt;平台日志」功能无数据</p> <p>如对历史使用情况无要求， 无需拷贝</p>
%Tomcat_HOME%/webapps/webroot/WEB-INF/classes	%BI_HOME%/fanruanxxx/bi6/classes	<p>作用： 工程调用的默认和自定义 class 文件</p> <p>是否必须拷贝： 必须拷贝</p> <p>有可能存在自定义的 class 文件, 如不拷贝会影响工程正常使用</p>
%Tomcat_HOME%/webapps/webroot/WEB-INF/config	%BI_HOME%/fanruanxxx/bi6/config	<p>作用： 配置库相关文件</p> <p>是否必须拷贝： 必须拷贝</p> <p>平台调用的配置库（finedb）， 如不拷贝会影响工程正常使用</p>
<p>找到以下两个目录：</p> <p>%Tomcat_HOME%/webapps/webroot/WEB-INF/lib</p> <p>%Tomcat_HOME%/webapps/webroot/WEB-INF/customLib</p> <p>除以下工程自带 JAR 之外的自定义定制 JAR， 全部拷贝：</p> <p>点击展开更多</p> <p>fdl-bi-extension-4.0.jar</p> <p>fdl-boot-4.0.jar</p> <p>fdl-core-4.0.jar</p> <p>fdl-cron-4.0.jar</p> <p>fdl-datasource-4.0.jar</p> <p>fdl-third-4.0.jar</p> <p>fine-accumulator-11.0.jar</p> <p>fine-activator-11.0.jar</p>	%BI_HOME%/fanruanxxx/bi6/customlib	<p>作用： 工程定制/外部引入的 JAR 包</p> <p>是否必须拷贝： 必须拷贝， 如不拷贝会影响模板访问</p>

<div>fine-autuator-formula-11.0.jar</div> <div>fine-bi-adapter-6.0.jar</div> <div>fine-bi-datamining-6.0.jar</div> <div>fine-bi-datamining-third-6.0.jar</div> <div>fine-bi-engine-spider-6.0.jar</div> <div>fine-bi-engine-third-6.0.jar</div> <div>fine-bi-foundation-6.0.jar</div> <div>fine-bi-middle-6.0.jar</div> <div>fine-bi-query-6.0.jar</div> <div>fine-bi-query-excel-6.0.jar</div> <div>fine-bi-query-third-6.0.jar</div> <div>fine-bi-scheduler-6.0.jar</div> <div>fine-bi-spider-cluster-6.0.jar</div> <div>fine-cbb-11.0.jar</div> <div>fine-core-11.0.jar</div> <div>fine-datasource-11.0.jar</div> <div>fine-decision-11.0.jar</div> <div>fine-decision-bi-11.0.jar</div> <div>fine-decision-report-11.0.jar</div> <div>fine-report-engine-11.0.jar</div> <div>fine-schedule-11.0.jar</div> <div>fine-schedule-bi-11.0.jar</div> <div>fine-schedule-report-11.0.jar</div> <div>fine-swift-log-adaptor-11.0.jar</div> <div>fine-third-11.0.jar</div> <div>fine-webui-11.0.jar</div> <div>h2-2.1.214.jar</div> <div>jtds-1.3.1.jar</div> <div>mssql-jdbc-9.4.1.jre8.jar</div> <div>mysql-connector-java-5.1.49-bin.jar</div> <div>ojdbc8.jar</div> <div>orai18n.jar</div> <div>sqlite-jdbc-3.35.4.jar</div> <div>jarfile-checksum.txt</div> <div>readme.txt</div> <div>version-bi.txt</div>		
<div>%Tomcat_HOME%/webapps/webroot/WEB-INF/plugins</div>	<div>%BI_HOME%/fanruanxxx/bi6/plugins</div>	<div>作用： 插件相关文件</div> <div>是否必须拷贝： 必须拷贝， 如不拷贝会影响插件相关功能的实现</div>
<div>%ftp%/backup</div>	<div>minio 文件服务器文件存放路径/backup</div>	<div>作用： 工程历史备份文件</div> <div>是否必须拷贝： 用户自行决定， 可存储</div>

原非容器化工程目录	新容器化工程外挂目录	说明
%Tomcat_HOME%/logs	%FR_HOME%/fanruanxxx/fr/tomcat_logs	作用：Tomcat 通用日志位置 是否必须拷贝：用户自行决定，Tomcat

		历史访问日志，可不拷贝
%Tomcat_HOME%/webapps/webroot/logs	%FR_HOME%/fanruanxxx/fr/logs	<p>作用：swift 日志</p> <p>是否必须拷贝：用户自行决定</p> <p>如果不拷贝，会丢失工程历史操作日志（即 logdb），「管理系统&gt;平台日志」功能无数据</p> <p>如对历史使用情况无要求，无需拷贝</p>
%Tomcat_HOME%/webapps/webroot/WEB-INF/classes	%FR_HOME%/fanruanxxx/fr/classes	<p>作用：工程调用的默认和自定义 class 文件</p> <p>是否必须拷贝：必须拷贝</p> <p>有可能存在自定义的 class 文件，如不拷贝会影响工程正常使用</p>
%Tomcat_HOME%/webapps/webroot/WEB-INF/config	%FR_HOME%/fanruanxxx/fr/config	<p>作用：配置库相关文件</p> <p>是否必须拷贝：必须拷贝</p> <p>平台调用的配置库（finedb），如不拷贝会影响工程正常使用</p>
<p>找到以下两个目录：</p> <p>%Tomcat_HOME%/webapps/webroot/WEB-INF/lib</p> <p>%Tomcat_HOME%/webapps/webroot/WEB-INF/customLib</p> <p>除以下工程自带的 JAR 之外的自定义定制 JAR，全部拷贝：</p> <p>点击展开更多</p> <p>fine-accumulator-11.0.jar</p> <p>fine-activator-11.0.jar</p> <p>fine-cbb-11.0.jar</p> <p>fine-core-11.0.jar</p> <p>fine-datasource-11.0.jar</p> <p>fine-decision-11.0.jar</p> <p>fine-decision-report-11.0.jar</p>	%FR_HOME%/fanruanxxx/fr/customlib	<p>作用：工程定制/外部引入的 JAR 包</p> <p>是否必须拷贝：必须拷贝，如不拷贝会影响模板访问</p>

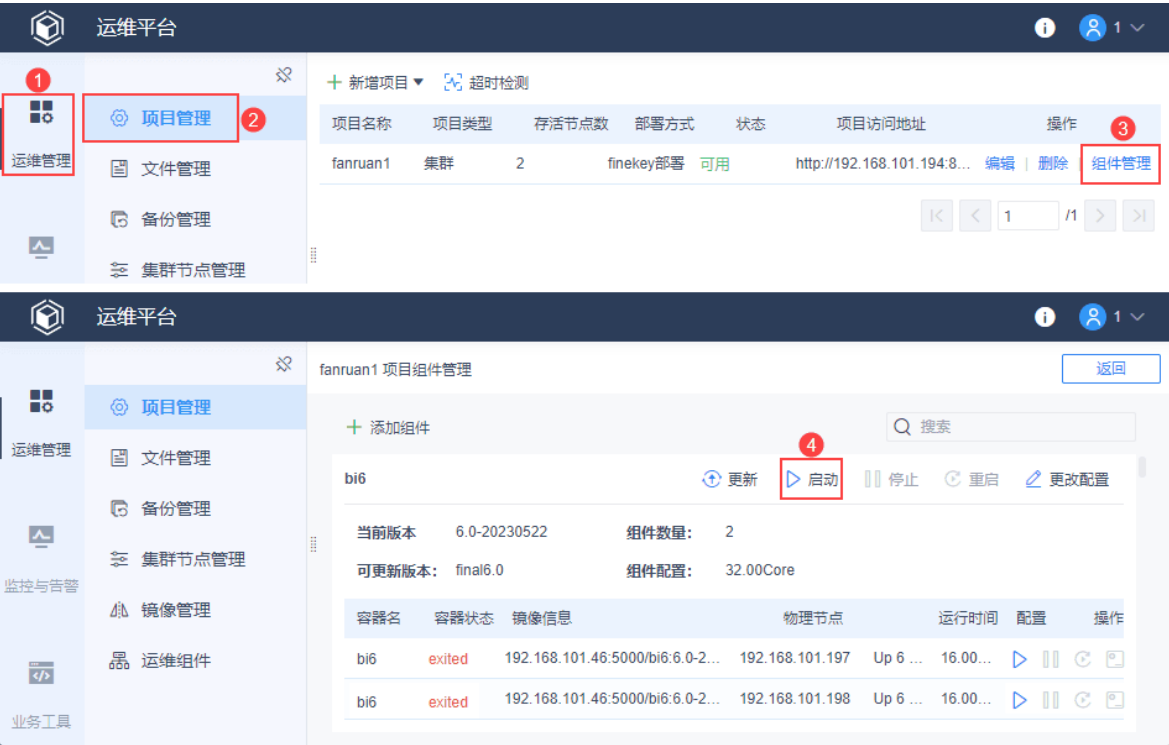
<div>fine-report-engine-11.0.jar</div> <div>fine-schedule-11.0.jar</div> <div>fine-schedule-report-11.0.jar</div> <div>fine-service-management-2.0.7.jar</div> <div>fine-swift-log-adaptor-11.0.jar</div> <div>fine-third-11.0.jar</div> <div>fine-webui-11.0.jar</div> <div>ifxjdbc_informix.jar</div> <div>jtids-1.3.1.jar</div> <div>mysql-connector-java-5.1.49-bin.jar</div> <div>ojdbc8.jar</div> <div>orai18n.jar</div> <div>sqlite-jdbc-3.39.4.0.jar</div> <div>sqljdbc.jar</div> <div>sybase.jar</div> <div>jarfile-checksum.txt</div> <div>readme.txt</div>		
<div>%Tomcat_HOME%/webapps/webroot/WEB-INF/plugins</div>	<div>%FR_HOME%/fanruanxxx/fr/plugins</div>	<div>作用： 插件相关文件</div> <div>是否必须拷贝： 必须拷贝， 如不拷贝会影响插件相关功能的实现</div>
<div>%ftp%/backup</div>	<div>minio 文件服务器文件存放路径/backup</div>	<div>作用： 工程历史备份文件</div> <div>是否必须拷贝： 用户自行决定， 可存储在原始的目录下备份， 可不拷贝</div>
<div>%ftp%/assets/temp_attach</div>	<div>minio 文件服务器文件存放路径/assets/temp_attach</div>	<div>作用： 读写缓存存储路径</div> <div>是否必须拷贝： 必须拷贝</div> <div>该文件夹中存放着工程读写（图片）缓存， 不拷贝可能会导致模板设置的背景图片预览为空</div>
<div>%ftp%/assets/vcs</div>	<div>minio 文件服务器文件存放路径/assets/vcs</div>	<div>作用： FineReport 模板备份文件</div> <div>是否必须拷贝： 用户自行决定</div> <div>如果工程不需要回退历史开发的 FineReport 模板， 则无需拷贝</div>
<div>%ftp%/assets/其他文件</div>	<div>minio 文件服务器文件存放路径/assets/其他文件</div>	<div>作用： 通用的共享持久化目录</div>

		是否必须拷贝：必须拷贝  工程正常运行所需要的文件，如不拷贝会影响工程正常使用
%ftp%/reportlets	minio 文件服务器文件存放路径/reportlets	作用：FineReport 模板存放目录  是否必须拷贝：必须拷贝，如不拷贝会导致工程所有模板丢失
%ftp%/schedule	minio 文件服务器文件存放路径/schedule	作用：定时调度生成的文件  是否必须拷贝：用户自行决定  如果不拷贝，定时任务挂载到决策平台的结果报表无法访问

4.2 启动新工程

用户登录运维平台，点击「运维管理>项目管理」，点击新容器化部署的工程项目后的「组件管理」按钮。

找到 bi6 组件，点击组件后的「启动」按钮，即可启动该集群所有工程节点。如下图所示：



4.3 验证是否迁移成功

依次登录所有工程节点：

- 1) 查看「管理系统>智能运维>集群配置>节点管理」中是否出现所有集群节点，说明多节点集群全部配置成功

2) 查看各节点的工程配置是否相同，说明外接数据库配置成功

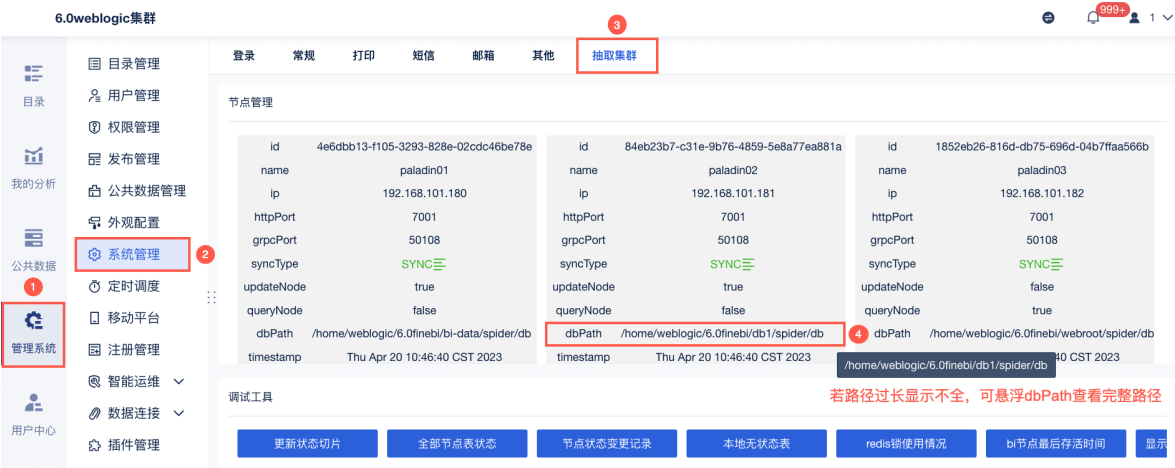


4.4 数据抽取

工程迁移后，一般需要重新进行数据抽取。

请先登录新工程，检查并修改数据存放路径。新服务器和老服务器的路径不一定一致，请确保是您所需要存放数据的路径。

新容器化集群默认为抽取集群，数据存放路径修改方法请参考：[数据存放路径](#)



5. 启动原非容器化工程

1) 将 3.1 节备份的 webroot 工程文件，覆盖原非容器化部署的工程各个节点的 webroot 工程

2) 参考 [关闭或重启 FineBI 工程](#) / [关闭或重启 FineReport 工程](#) 文档，启动原非容器化部署的工程。

6. 迁移授权

若老工程进行了注册，一般需要将授权迁移到新工程。容器化工程推荐使用 [容器私有云认证](#) 方式。



## 6.1 销毁原非容器化工程授权

- 1) 参考 [授权迁移插件](#) 的使用步骤，向商务索要一个授权迁移插件，销毁**原非容器化工程**的旧授权。
- 2) 找到**原非容器化工程**%BI\_HOME%/webroot/WEB-INF/resources 文件夹下的销毁凭证。

## 6.2 获取新容器化工程授权

参考 [容器私有云认证](#) 的使用步骤，向商务索要注册镜像文件 fanruan\_license\_server.tar，并参考文档为**新容器化工程**进行授权。

请在邮件中附上上一步获取的销毁凭证。并注明是「非容器化工程迁移容器化工程」

## 4.5.5 容器化单机迁移容器化单机

### 1. 概述

#### 1.1 版本

工程版本	功能变更
FineReport11.0	-
FineBI6.0	-

#### 1.2 应用场景

有时用户需要复用一个工程，或者迁移工程所在服务器。

本文方案，帮助用户将原有容器化部署的单机工程，迁移到另一个容器化部署的单机工程。

注：全文以 FineBI 作为演示。FineReport 迁移步骤基本一致，仅需要两个点：

- 1) 拷贝文件时，FineReport 和 FineBI 需要拷贝的文件不完全相同，注意文档中的表格即可。
- 2) 使用运维平台启停容器时，FineReport 关闭的是 fr 组件，FineBI 关闭的是 bi6 组件。

### 2. 新工程准备

#### 2.1 部署运维平台

容器化 FineBI 工程，需要通过运维平台部署。因此用户需要先部署一个 FineOps 运维平台。

详情请参见：[部署运维平台](#)

#### 2.2 部署新容器化工程

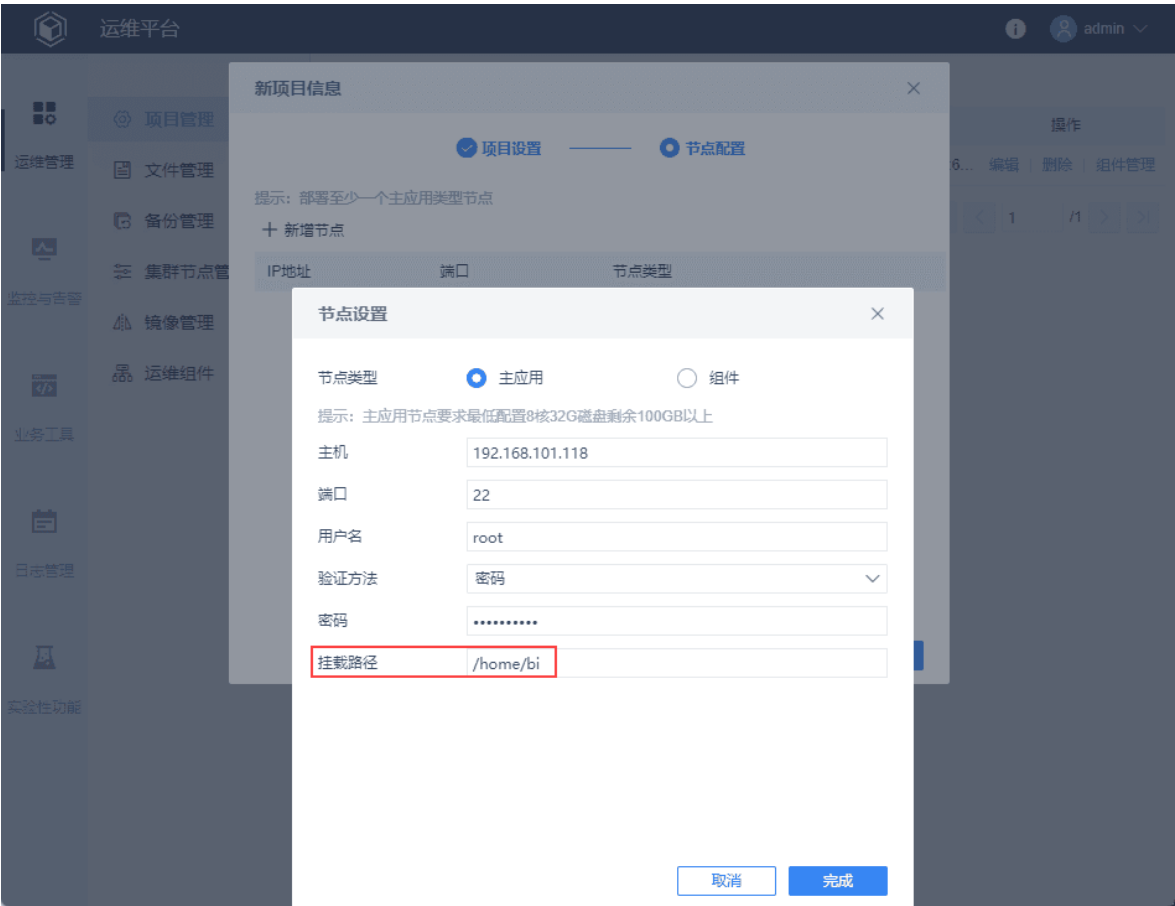
进行迁移前，用户需要先通过运维平台部署一个容器化单机工程。

详情请参见：[部署新项目-FineBI](#)、[部署新项目-FineReport](#)

注 1：请记录下 MySQL 组件的相关信息，由于密码随机生成，请务必修改 MySQL 组件的密码，方便后续使用。



注 2：请记录下 FineBI 工程的挂载路径，用户可自定义路径。这是工程文件的存放路径，后面步骤中需要用到。



### 2.3 确认新工程服务器剩余磁盘空间

- 1) 检查原工程的大小。
- 2) 根据原工程大小，适度扩大新工程所在服务器节点剩余磁盘空间，建议大于原工程的两倍，最少剩余磁盘大小 500G。

### 2.4 关闭新工程

用户登录运维平台，点击「运维管理>项目管理」，点击新工程项目后的「组件管理」按钮。

找到 bi6 组件，点击「停止」按钮，即可关闭新工程。如下图所示：



## 2.5 备份 config

请在新工程的挂载路径/fanruanxxx/bi 目录下，找到 config 文件夹，并另行备份。

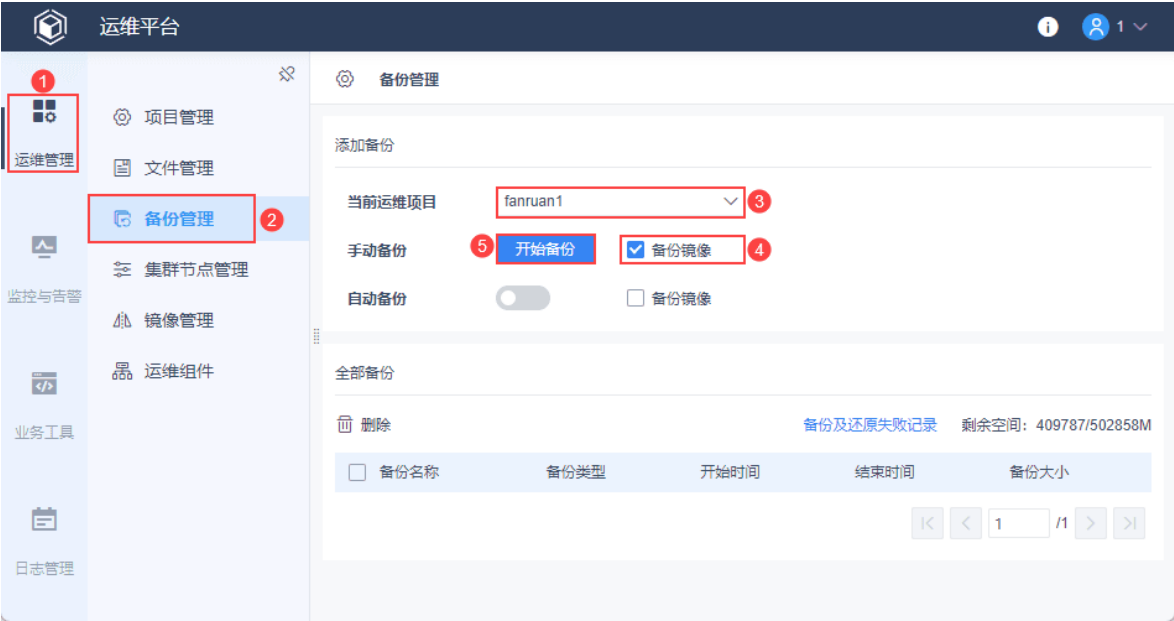
该文件夹中存放的 yaml 文件中，有着工程部署所用语句，后续运营维护时可能会需要使用/查找，请务必异地备份。

## 3. 原工程准备

### 3.1 备份原工程

在进行工程迁移前，为避免迁移失败，导致工程文件丢失无法回退等问题，请务必对原工程进行备份后再进行后续操作。

请使用运维平台的 [备份管理](#) 功能对原工程进行手动备份。



### 3.2 升级原工程

工程迁移的前提是，两个工程必须处于同一版本。

管理员登录运维平台，点击「运维管理>项目管理」点击新老工程后的「组件管理」，查看 bi6 组件的当前版本是否相同（精确到日期）。

若两个工程版本不相同，可参考该文档更新工程版本：[外网升级指定版本/内网升级项目](#)



### 3.3 关闭原工程

用户登录运维平台，点击「运维管理>项目管理」，点击原工程项目后的「组件管理」按钮。

找到 bi6 组件，点击「停止」按钮，即可关闭新工程。如下图所示：



### 4. 迁移工程

### 4.1 拷贝外接数据库

将原容器化项目中的 MySQL 组件的外挂目录数据，拷贝到新工程项目中的 MySQL 组件的外挂目录里。

注：MySQL 组件的外挂目录地址，一般在项目挂载路径/fanruanxxx/mysql 下。

```
[root@mail ~]# cd /home/bi/fanruan230529143543/mysql/
[root@mail mysql]# ll
总用量 4
drwxr-x--- 2 polkitd root    6 5月  29 14:35 conf
drwxr-x--- 6 polkitd root 4096 5月  29 14:35 data
drwxr-x--- 2 polkitd root   23 5月  29 14:35 log
[root@mail mysql]#
```

### 4.2 拷贝工程文件

将原容器化项目中的 bi6 组件的外挂目录数据，拷贝到新工程项目中的 bi6 组件的外挂目录里。

- 1) 请提前检查原工程，是否修改了外挂目录位置、是否配置共享文件服务器，确保拷贝到正确的文件。
- 2) 新工程外挂文件所在位置，为 2.2 节记录的挂载路径。
- 3) 将以下文件，从原工程外挂目录，拷贝到新工程外挂目录中：

注 1：请勿直接拷贝整个外挂目录文件夹到新工程，部分文件不可以被覆盖。

注 2：请勿使用 3.1 节备份的文件，而是使用工程最终的文件进行拷贝操作。

#### • FineBI 需要拷贝的文件

外挂目录文件位置	说明
	作用：FineBI 数据表相关信息
	是否必须拷贝：必须拷贝
%BI_HOME%/fanruanxxx/bi6/assets/temp_attach	该文件夹中存放着工程 Excel 原始文件信息，如不拷贝该文件夹会导致原始 Excel 丢失
	作用：FineReport 模板备份文件
	是否必须拷贝：用户自行决定
%BI_HOME%/fanruanxxx/bi6/assets/vcs	如果工程未使用 FineReport 模板，或不需要回退历史开发的 FineReport 模板，则无需拷贝

	作用：通用的共享持久化目录
%BI_HOME%/fanruanxxx/bi6/assets/其他文件	<p>是否必须拷贝：必须拷贝</p> <p>工程正常运行所需要的文件，如不拷贝会影响工程正常使用</p> <p>作用：工程历史备份文件</p>
%BI_HOME%/fanruanxxx/bi6/backup	<p>是否必须拷贝：用户自行决定，可存储在原始的目录下备份，可不拷贝</p> <p>作用：FineBI 抽取数据</p> <p>是否必须拷贝：</p>
%BI_HOME%/fanruanxxx/bi6/bi-data	<p>如果新老工程在同一服务器上，可重新-v 挂载原工程的 bi-data 文件夹，或在新工程启动后重新抽取数据，无需拷贝</p> <p>如果新老工程不在同一服务器上，不建议拷贝，该文件夹一般数据量太大，拷贝困难，可在新工程启动后重新抽取数据</p> <p>作用：工程调用的默认和自定义 class 文件</p>
%BI_HOME%/fanruanxxx/bi6/classes	<p>是否必须拷贝：必须拷贝</p> <p>有可能存在自定义的 class 文件，如不拷贝会影响工程正常使用</p>
%BI_HOME%/fanruanxxx/bi6/customlib	<p>作用：工程定制/外部引入的 JAR 包</p> <p>是否必须拷贝：必须拷贝，如不拷贝会影响模板访问</p> <p>作用：FineDataLink 任务相关的配置文件</p> <p>是否必须拷贝：必须拷贝，如不拷贝会影响 FineDataLink 任务使用</p>
%BI_HOME%/fanruanxxx/bi6/dpworks	<p>（如不存在该文件夹，说明工程没有使用 FDL 相关功能，略过）</p> <p>作用：swift 日志</p>
%BI_HOME%/fanruanxxx/bi6/logs	<p>是否必须拷贝：用户自行决定</p> <p>如果不拷贝，会丢失工程历史操作日志（即 logdb），「管理系统&gt;平台日志」功能无数据</p>

	如对历史使用情况无要求，无需拷贝
	作用：插件相关文件
%BI_HOME%/fanruanxxx/bi6/plugins	是否必须拷贝：必须拷贝，如不拷贝会影响插件相关功能的实现
	作用：FineReport 模板存放目录
%BI_HOME%/fanruanxxx/bi6/reportlets	是否必须拷贝：用户自行决定
	如果工程未使用 FineReport 模板，则无需拷贝
	作用：定时调度生成的文件
%BI_HOME%/fanruanxxx/bi6/schedule	是否必须拷贝：用户自行决定
	如果不拷贝，定时任务挂载到决策平台的结果报表无法访问
	作用：Tomcat 通用日志位置
%BI_HOME%/fanruanxxx/bi6/tomcat_logs	是否必须拷贝：用户自行决定，Tomcat 历史访问日志，可不拷贝

• **FineReport 需要拷贝的文件**

外挂目录文件位置	说明
	作用：读写缓存存储路径
%FR_HOME%/fanruanxxx/fr/assets/temp_attach	是否必须拷贝：必须拷贝
	该文件夹中存放着工程读写（图片）缓存，不拷贝可能会导致模板设置的背景图片预览为空
	作用：FineReport 模板备份文件
%FR_HOME%/fanruanxxx/fr/assets/vcs	是否必须拷贝：用户自行决定
	如果工程不需要回退历史开发的 FineReport 模板，则无需拷贝
	作用：通用的共享持久化目录
%FR_HOME%/fanruanxxx/fr/assets/其他文件	是否必须拷贝：必须拷贝
	工程正常运行所需要的文件，如不拷贝会影响工程正常使用
%FR_HOME%/fanruanxxx/fr/backup	作用：工程历史备份文件



	是否必须拷贝：用户自行决定，可存储在原始的目录下备份，可不拷贝
	作用：工程调用的默认和自定义 class 文件
%FR_HOME%/fanruanxxx/f	是否必须拷贝：必须拷贝
r/classes	有可能存在自定义的 class 文件，如不拷贝会影响工程正常使用
	作用：工程定制/外部引入的 JAR 包
%FR_HOME%/fanruanxxx/f	是否必须拷贝：必须拷贝，如不拷贝会影响模板访问
r/customlib	作用：FineDataLink 任务相关的配置文件
	是否必须拷贝：必须拷贝，如不拷贝会影响 FineDataLink 任务使用
%FR_HOME%/fanruanxxx/f	(如不存在该文件夹，说明工程没有使用 FDL 相关功能，略过)
r/dpworks	作用：swift 日志
	是否必须拷贝：用户自行决定
%FR_HOME%/fanruanxxx/f	如果不拷贝，会丢失工程历史操作日志（即 logdb），「管理系统>平台日志」功能无数据
r/logs	如对历史使用情况无要求，无需拷贝
	作用：插件相关文件
%FR_HOME%/fanruanxxx/f	是否必须拷贝：必须拷贝，如不拷贝会影响插件相关功能的实现
r/plugins	作用：FineReport 模板存放目录
	是否必须拷贝：必须拷贝，如不拷贝会导致工程所有模板丢失
%FR_HOME%/fanruanxxx/f	作用：定时调度生成的文件
r/reportlets	是否必须拷贝：用户自行决定
	如果不拷贝，定时任务挂载到决策平台的结果报表无法访问
%FR_HOME%/fanruanxxx/f	作用：Tomcat 通用日志位置
r/tomcat_logs	

是否必须拷贝：用户自行决定，Tomcat 历史访问日志，可不拷贝

5. 启动新老工程

用户登录运维平台，点击「运维管理>项目管理」，点击新老工程项目后的「组件管理」按钮。

找到 bi6 组件，点击「启动」按钮，即可启动新老工程。如下图所示：



6. 迁移授权

容器化工程一般使用「容器私有云认证」方式进行注册。

请参考文档第四章进行授权迁移操作：[容器私有云认证](#)

## 4.5.6 容器化单机迁移容器化集群

### 1. 概述

#### 1.1 版本

工程版本	功能变更
FineReport11.0	-
FineBI6.0	-

#### 1.2 应用场景

相比于单机工程，集群具有高可用性、高性能、易于管理、可伸缩性和安全保障等优点，适用于企业级的报表生成和管理需求。

本文方案，帮助用户将原有容器化部署的单机工程，迁移为容器化部署的集群工程。

注：全文以 FineBI 作为演示。FineReport 迁移步骤基本一致，仅需要两个点：

- 1) 拷贝文件时，FineReport 和 FineBI 需要拷贝的文件不完全相同，注意文档中的表格即可。
- 2) 使用运维平台启停容器时，FineReport 关闭的是 fr 组件，FineBI 关闭的是 bi6 组件。

### 2. 新集群工程准备

#### 2.1 部署运维平台

容器化 FineBI 工程，需要通过运维平台部署。因此用户需要先部署一个 FineOps 运维平台。

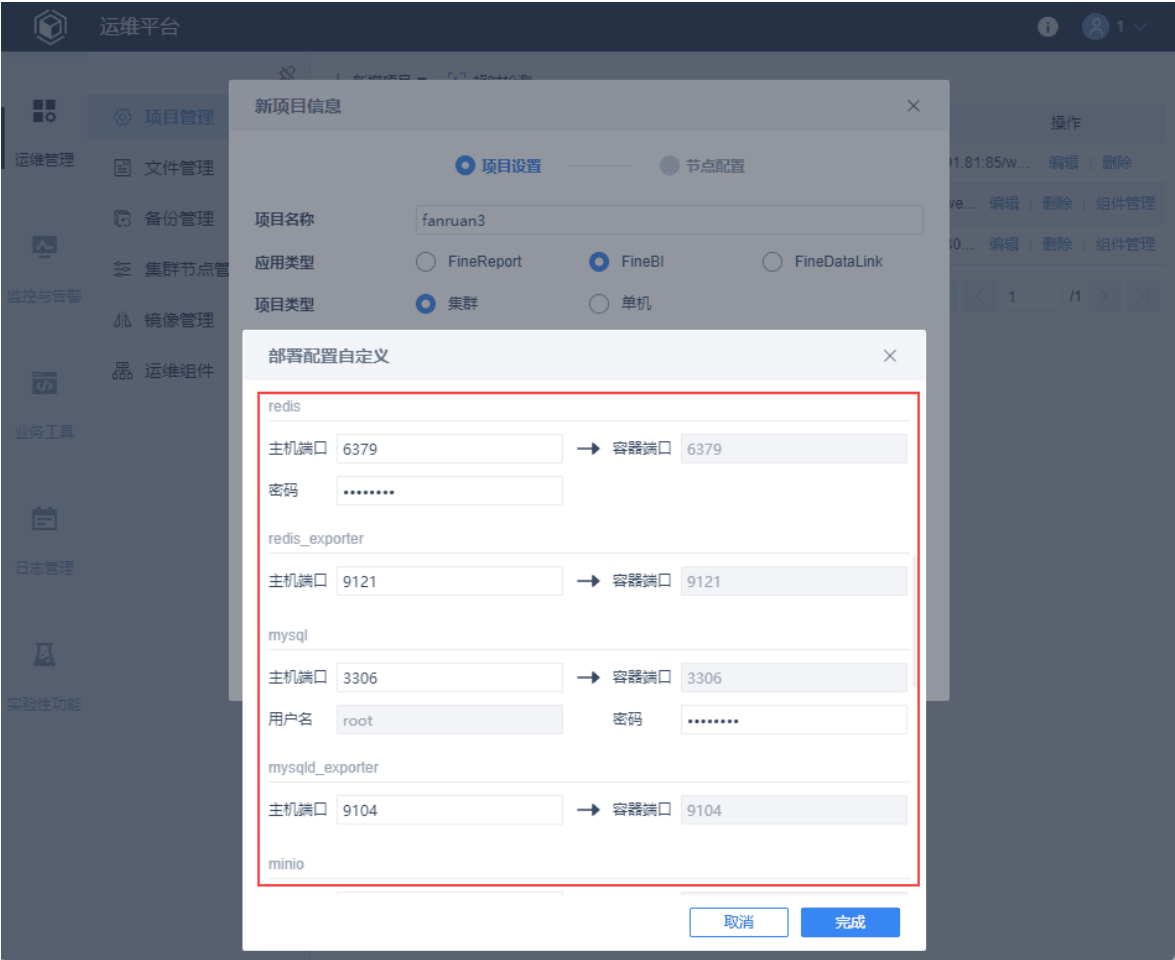
详情请参见：[部署运维平台](#)

#### 2.2 部署新工程

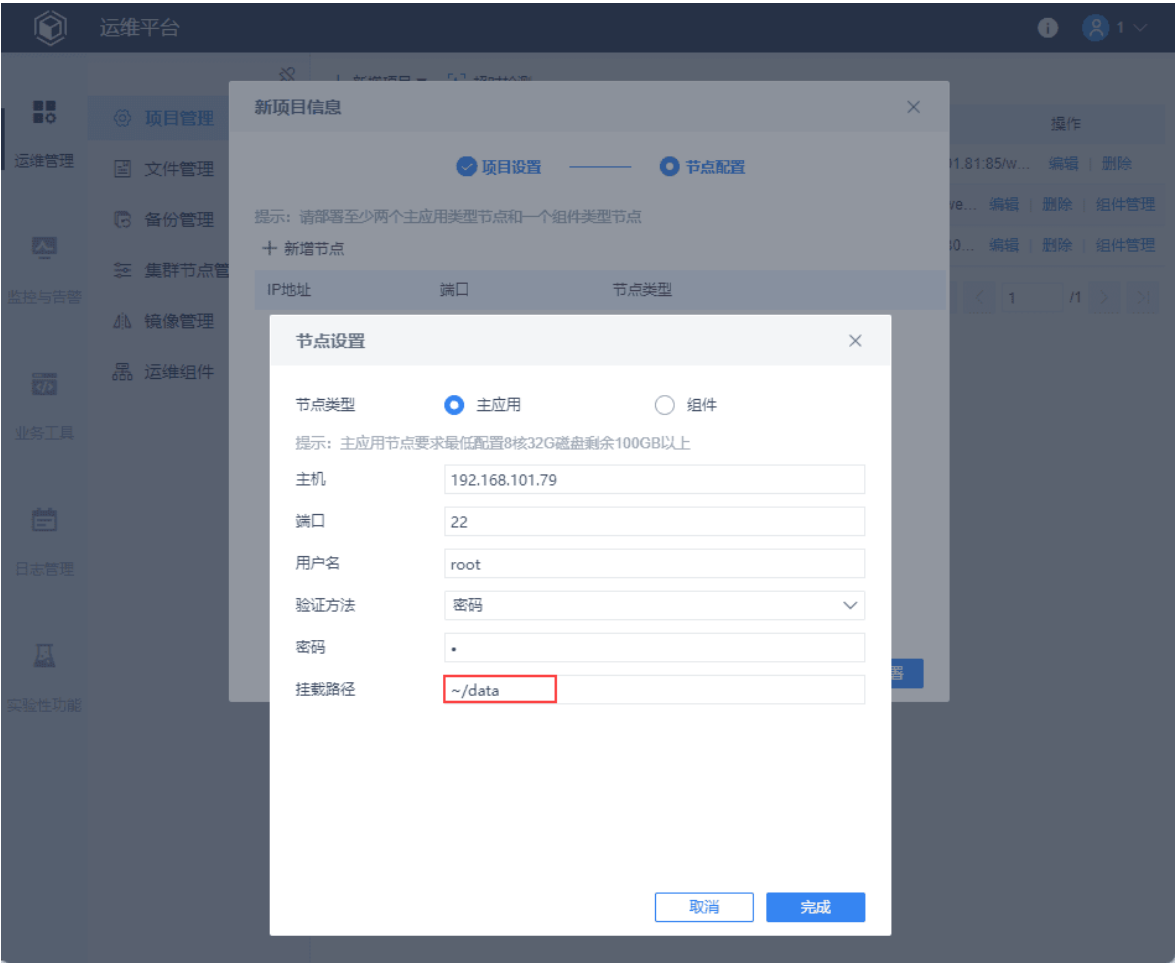
进行迁移前，用户需要先通过运维平台部署一个容器化集群工程。

详情请参见：[部署新项目-FineBI](#)、[部署新项目-FineReport](#)

注 1：请记录下 MySQL、Redis、minio 组件的相关信息，由于密码随机生成，请务必修改 MySQL、Redis、minio 组件的密码，方便后续使用。



注 2：请记录下 FineBI 工程的挂载路径，用户可自定义路径。这是工程文件的存放路径，后面步骤中需要用到。



### 2.3 确认新工程服务器剩余磁盘空间

- 1) 检查原工程的大小。
- 2) 根据原工程大小，适度扩大新工程所在服务器节点剩余磁盘空间，建议大于原工程的两倍，最少剩余磁盘大小 500G。

2.4 记录新工程集群信息

管理员登录新工程，点击「管理系统>智能运维>集群配置」。

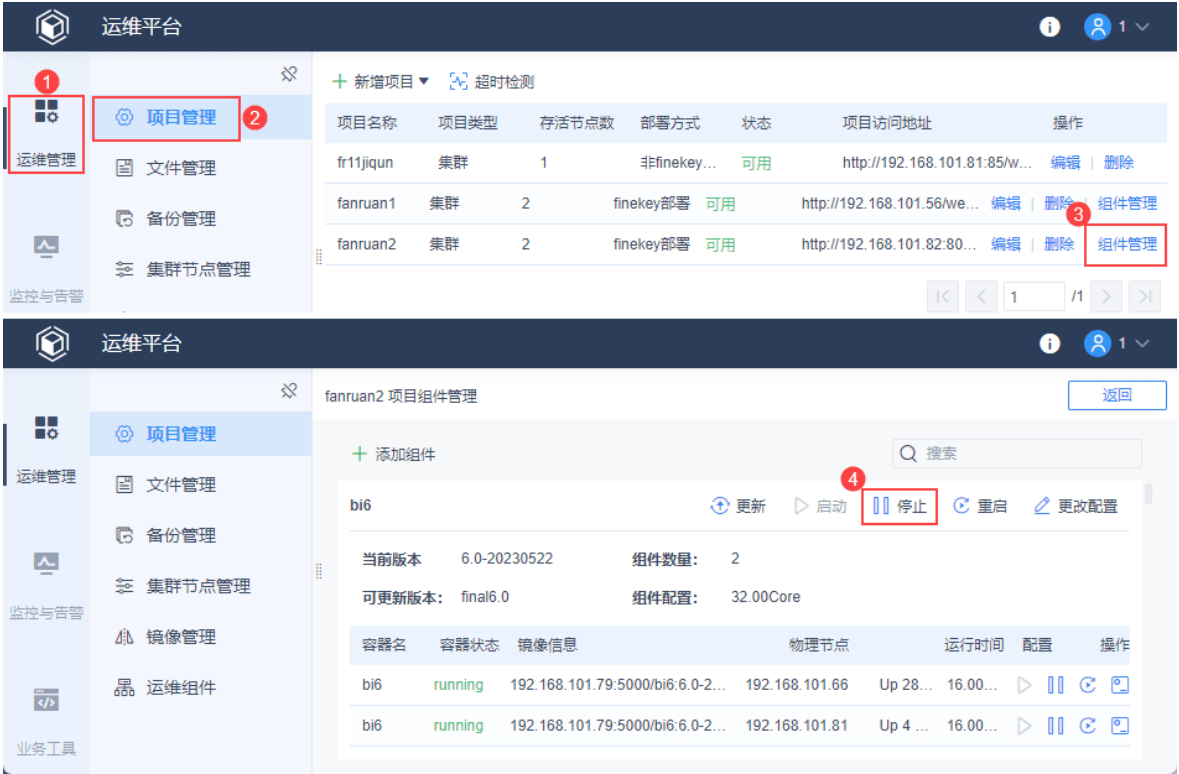
记录下状态服务器 Redis 和文件服务器 MinIO 的相关信息（密码是 2.2 节记录下的密码）。



2.5 关闭新工程

用户登录运维平台，点击「运维管理>项目管理」，点击新工程项目后的「组件管理」按钮。

找到 bi6 组件，点击「停止」按钮，即可关闭新工程。如下图所示：



## 2.6 备份 config

请在新工程任意节点的挂载路径/fanruanxxx/bi 目录下，找到 config 文件夹，并另行备份。

该文件夹中存放的 yaml 文件中，有着工程部署所用语句，后续运营维护时可能会需要使用/查找，请务必异地备份。

## 3. 原单机工程准备

### 3.1 备份原工程

在进行工程迁移前，为避免迁移失败，导致工程文件丢失无法回退等问题，请务必对原工程进行备份后再进行后续操作。

请使用运维平台的 [备份管理](#) 功能对原工程进行手动备份。



### 3.2 升级原工程

工程迁移的前提是，两个工程必须处于同一版本。

管理员登录运维平台，点击「运维管理>项目管理」点击新老工程后的「组件管理」，查看 bi6 组件的当前版本是否相同（精确到日期）。

若两个工程版本不相同，可参考该文档更新工程版本：[外网升级指定版本/内网升级项目](#)



### 3.3 映射集群所需端口

#### 1) FineBI

单机环境重新 docker run 修改 7800，7830，7840，7850，7870 配置

```
docker run -e "DB_CACHE_EXTERNAL_PORT=7850" -e "MEMBER_EXTERNAL_PORT=7870" -e "FILE_SYNC_EXTERNAL_PORT=7830" -e "GENERAL_CACHE_EXTERNAL_PORT=7840" -e "CORE_EXTERNAL_PORT=7800"
```

单机环境重新 docker run 添加映射端口，添加 50050、50051、50100、50101、50200 配置

```
docker run -p IP:7800:7800 -p IP:7830:7830 -p IP:7840:7840 -p IP:7850:7850 -p IP:7870:7870 -p IP:50050:50050 -p IP:50051:50051 -p IP:50100:50100 -p IP:50101:50101 -p IP:50200:50200 -e "BI_SERVER_PORT_EXPOSE=50101" -e "SPIDER_SERVER_PORT_EXPOSE=50100" -e "BI_CONF_RPC_PORT_EXPOSE=50200" -e "SPIDER_GRP_C_PORT_EXPOSE=50051" -e "BI_GRP_C_PORT_EXPOSE=50050"
```

#### 2) FineReport

单机环境重新 docker run 修改 7800，7830，7840，7850，7870 配置

```
docker run -e "DB_CACHE_EXTERNAL_PORT=7850" -e "MEMBER_EXTERNAL_PORT=7870" -e "FILE_SYNC_EXTERNAL_PORT=7830" -e "GENERAL_CACHE_EXTERNAL_PORT=7840"
```

e "CORE\_EXTERNAL\_PORT=7800"

单机环境重新 docker run 添加映射端口

```
docker run -p IP:7800:7800 -p IP:7830:7830 -p IP:7840:7840 -p IP:7850:7850 -
p IP:7870:7870
```

注意：

以上命令并非完整的 docker run 命令，用户需要基于自己本地原有配置，将以上命令添加/修改进去运行。

此处提供一个完整命令示例，仅作为示例！切勿直接复制使用！

点击展开更多

```
docker run -it -d --log-driver local --restart=on-failure:10 \
--name fanruan230508163535_bi6 -v /home/jasmine/finekeydata/fanruan230508163535/bi6/lo
ogs:/usr/local/tomcat/webapps/webroot/logs \
-v /home/jasmine/finekeydata/fanruan230508163535/bi6/bi-data:/usr/local/tomcat/webapps/w
ebroot/bi-data \
-v /home/jasmine/finekeydata/fanruan230508163535/bi6/assist:/usr/local/tomcat/webapps/we
broot/WEB-INF/assist \
-v /home/jasmine/finekeydata/fanruan230508163535/bi6/mount:/usr/local/tomcat/mount \
-v /home/jasmine/finekeydata/fanruan230508163535/bi6/config:/usr/local/tomcat/webapps/we
broot/WEB-INF/config \
-v /home/jasmine/finekeydata/fanruan230508163535/bi6/tomcat_logs:/usr/local/tomcat/logs \
-v /home/jasmine/finekeydata/fanruan230508163535/bi6/plugins:/usr/local/tomcat/webapps/w
ebroot/WEB-INF/plugins \
-v /home/jasmine/finekeydata/fanruan230508163535/bi6/customlib:/usr/local/tomcat/webapps
/webroot/WEB-INF/customlib \
-v /home/jasmine/finekeydata/fanruan230508163535/bi6/classes:/usr/local/tomcat/webapps/w
ebroot/WEB-INF/classes \
-v /home/jasmine/finekeydata/fanruan230508163535/bi6/assets:/usr/local/tomcat/webapps/we
broot/WEB-INF/assets \
-v /home/jasmine/finekeydata/fanruan230508163535/bi6/schedule:/usr/local/tomcat/webapps/
webroot/WEB-INF/schedule \
-v /home/jasmine/finekeydata/fanruan230508163535/bi6/dashboards:/usr/local/tomcat/webap
ps/webroot/WEB-INF/dashboards \
-v /home/jasmine/finekeydata/fanruan230508163535/bi6/backup:/usr/local/tomcat/webapps/w
ebroot/backup \
-v /home/jasmine/finekeydata/fanruan230508163535/bi6/dpworks:/usr/local/tomcat/webapps/
webroot/WEB-INF/dpworks \
-v /home/jasmine/finekeydata/fanruan230508163535/bi6/reportlets:/usr/local/tomcat/webapps
/webroot/WEB-INF/reportlets \
-v /home/jasmine/finekeydata/fanruan230508163535/bi6/resources:/usr/local/tomcat/webapps
/webroot/WEB-INF/resources \
-p 192.168.101.83:7800:7800 -p 192.168.101.83:7830:7830 -p 192.168.101.83:7840:7840 \
-p 192.168.101.83:7850:7850 -p 192.168.101.83:7870:7870 \
-p 192.168.101.83:50050:50050 -p 192.168.101.83:50051:50051 \
-p 192.168.101.83:50100:50100 -p 192.168.101.83:50101:50101 -p 192.168.101.83:50200:50200 \
-p 192.168.101.83:8080:8080 -p 192.168.101.83:12100:12100 -e "ops_sign_key=cc0f9
2d0a6274e4e" \
-e "DB_CACHE_EXTERNAL_PORT=7850" -e "MEMBER_EXTERNAL_PORT=7870" -e \
"TZ=Asia/Shanghai" -e "AGENT_ADDRESSES=http://192.168.101.83:9370/" \
-e "CONTEXT_PATH=webroot" -e "CATALINA_OPTS=-Xmx8192m" -e "USER_ID=0" \
-e "FILE_SYNC_EXTERNAL_PORT=7830" -e "GROUP_ID=0" -e "GENERAL_CACHE_EXTERNAL_P
```



```
ORT=7840" \  
-e "jgroups.external_addr=192.168.101.83" -e "EXTERNAL_HTTP_PORT=8080" \  
-e "BI_SERVER_PORT_EXPOSE=50101" -e "SPIDER_SERVER_PORT_EXPOSE=50100" -e "BI_CO  
NF_RPC_PORT_EXPOSE=50200" \  
-e "SPIDER_GRPc_PORT_EXPOSE=50051" -e "BI_GRPc_PORT_EXPOSE=50050" \  
-e "CORE_EXTERNAL_PORT=7800" --ulimit nofile=65536:65536 --ulimit memlock=-1:-1 \  
192.168.101.83:5000/bi6:6.0-20230425
```

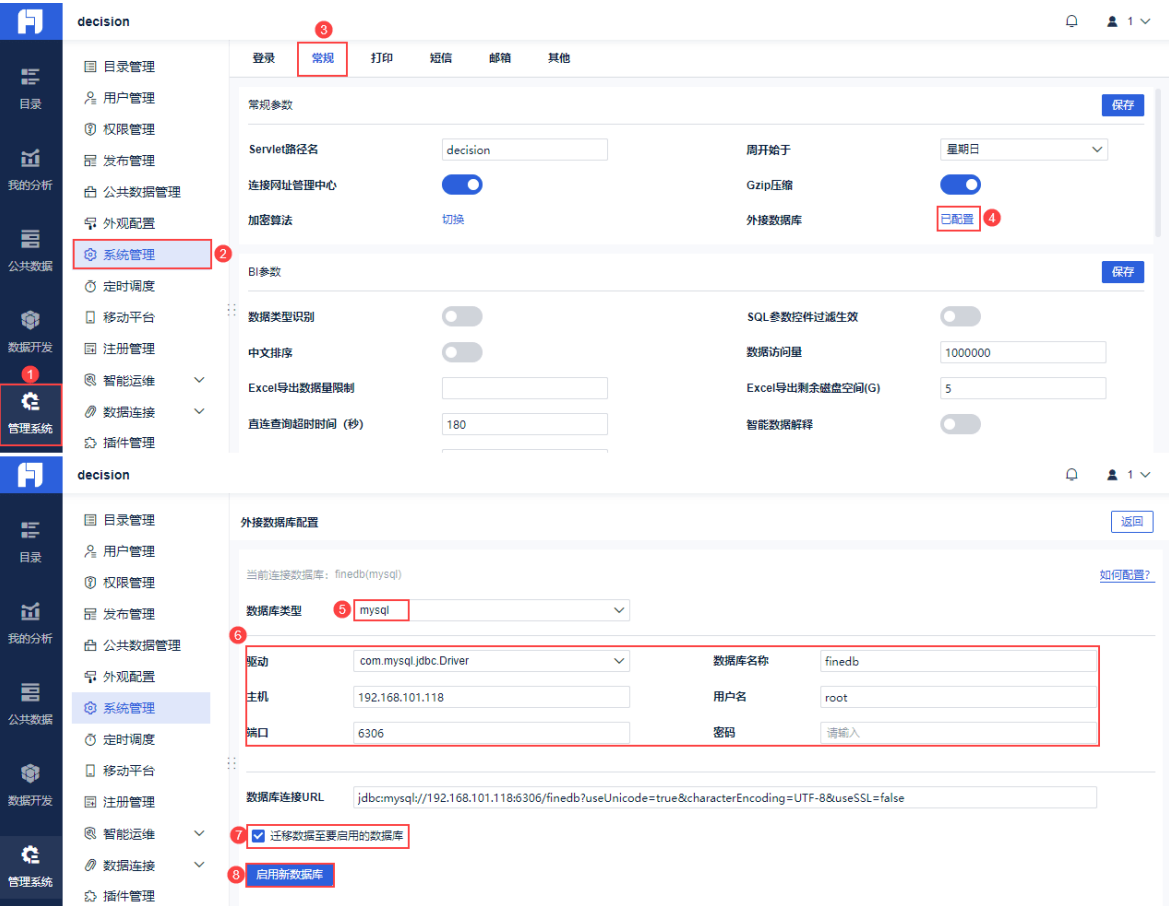
3.4 迁移外接数据库

为了保证新老工程的平台配置完全一致，需要同步两个工程的配置库。因此需要让原工程 finedb 迁移到新工程项目的 MySQL 组件。

管理员登录原工程，点击「管理系统>系统管理>常规>外接数据库」。

- 配置外接数据库为 2.2 节部署的 MySQL 组件，其中数据库名称为 finedb
- 勾选「迁移数据至要启用的数据库」
- 点击「启用新数据库」

当出现提示「已成功切换至目标数据库!」，则代表外接数据库切换成功。



3.5 安装 minio 资源仓库插件

由于需要让原单机工程对接新集群组件，因此需要先安装必要插件。

管理员登录原工程，点击「管理系统>插件管理」，安装「minio 资源仓库」插件。如下图所示：



3.6 对接集群组件

- 1) 管理员登录原单机工程，点击「管理系统>智能运维>集群配置」。
- 2) 开启「状态服务器」。选择「单机 Redis」，输入 2.4 节记录的 Redis 组件信息。
- 3) 开启「文件服务器共享」。选择「MinIO」，输入 2.4 节记录的 MinIO 组件信息。
- 4) 缓存模式选择「关闭缓存」，集群通信协议选择「TCP」。
- 5) 开启「开启集群」按钮。
- 6) 用户登录运维平台，点击「运维管理>项目管理」，点击原单机工程项目后的「组件管理」按钮。找到 bi6 组件，点击「重启」按钮，即可重启工程。
- 7) 再次查看原工程的「管理系统>智能运维>集群配置」，在节点管理处出现本工程节点，说明单节点集群部署成功。



### 3.7 关闭原工程

用户登录运维平台，点击「运维管理>项目管理」，点击原工程项目后的「组件管理」按钮。

找到 bi6 组件，点击「停止」按钮，即可关闭新工程。如下图所示：



## 4. 迁移工程

### 4.1 拷贝工程文件

将原工程 bi6 组件的外挂目录数据，拷贝到新项目各个工程节点的 bi6 组件的外挂目录里。

- 1) 请提前检查原工程，是否修改了外挂目录位置、是否配置共享文件服务器，确保拷贝到正确的文件。
- 2) 新工程外挂文件所在位置，为 2.2 节记录的挂载路径。
- 3) 将以下文件，从原工程外挂目录，拷贝到新工程外挂目录中：

注 1：请勿直接拷贝整个外挂目录文件夹到新工程，部分文件不可以被覆盖。

注 2：请勿使用 3.1 节备份的文件，而是使用工程最终的文件进行拷贝操作。

#### • FineBI 需要拷贝的文件

原工程目录	新工程外挂目录	说明
%BI_HOME%/fanruanxxx/bi6/bi-data	%BI_HOME%/fanruanxxx/bi6/bi-data	<p>作用：FineBI 抽取数据</p> <p>是否必须拷贝：</p> <p>如果新老工程在同一服务器上，可重新-v 挂载原工程的 bi-data 文件夹，或在新工程启动后重新抽取数据，无需拷贝</p> <p>如果新老工程不在同一服务器上，不建议拷贝，该文件夹一般数据量太大，拷贝困难，可在新工程启动后重新</p>

		抽取数据
%BI_HOME%/fanruanxxx/bi6/classes	%BI_HOME%/fanruanxxx/bi6/classes	作用：工程调用的默认和自定义 class 文件  是否必须拷贝：必须拷贝  有可能存在自定义的 class 文件，如不拷贝会影响工程正常使用
%BI_HOME%/fanruanxxx/bi6/config	%BI_HOME%/fanruanxxx/bi6/config	作用：配置库相关文件  是否必须拷贝：必须拷贝  平台调用的配置库 (finedb)，如不拷贝会影响工程正常使用
%BI_HOME%/fanruanxxx/bi6/customLib	%BI_HOME%/fanruanxxx/bi6/customLib	作用：工程定制/外部引入的 JAR 包  是否必须拷贝：必须拷贝，如不拷贝会影响模板访问
%BI_HOME%/fanruanxxx/bi6/logs	%BI_HOME%/fanruanxxx/bi6/logs	作用：swift 日志  是否必须拷贝：用户自行决定  如果不拷贝，会丢失工程历史操作日志 (即 logdb), 「管理系统>平台日志」功能无数据  如对历史使用情况无要求，无需拷贝
%BI_HOME%/fanruanxxx/bi6/plugins	%BI_HOME%/fanruanxxx/bi6/plugins	作用：插件相关文件  是否必须拷贝：必须拷贝，如不拷贝会影响插件相关功能的实现
%BI_HOME%/fanruanxxx/bi6/tomcat_logs	%BI_HOME%/fanruanxxx/bi6/tomcat_logs	作用：Tomcat 通用日志位置  是否必须拷贝：用户自行决定，Tomcat 历史访问日志，可不拷贝
%BI_HOME%/fanruanxxx/bi6/backup	minio 文件服务器文件存放路径 /backup	作用：工程历史备份文件  是否必须拷贝：用户自行决定，可存储在原始的目录下备份，可不拷贝
%BI_HOME%/fanruanxxx/bi6/assets/temp_att	minio 文件服务器文件存放路径 /assets/temp_att	作用：FineBI 数据表相关信息

ach	ach	是否必须拷贝： 必须拷贝  该文件夹中存放着工程 Excel 原始文件信息，如不拷贝  该文件夹会导致原始 Excel 丢失
%BI_HOME%/fanruanxxx/bi6/assets/vcs	minio 文件服务器文件存放路径 /assets/vcs	作用： FineReport 模板备份文件  是否必须拷贝： 用户自行决定  如果工程未使用 FineReport 模板, 或不需要回退历史开发的 FineReport 模板，则无需拷贝
%BI_HOME%/fanruanxxx/bi6/assets/其他文件	minio 文件服务器文件存放路径 /assets/其他文件	作用： 通用的共享持久化目录  是否必须拷贝： 必须拷贝  工程正常运行所需要的文件, 如不拷贝会影响工程正常使用
%BI_HOME%/fanruanxxx/bi6/reportlets	minio 文件服务器文件存放路径 /reportlets	作用： FineReport 模板存放目录  是否必须拷贝： 用户自行决定  如果工程未使用 FineReport 模板，则无需拷贝
%BI_HOME%/fanruanxxx/bi6/schedule	minio 文件服务器文件存放路径 /schedule	作用： 定时调度生成的文件  是否必须拷贝： 用户自行决定  如果不拷贝, 定时任务挂载到决策平台的结果报表无法访问

• **FineReport 需要拷贝的文件**

原工程目录	新工程外挂目录	说明
%FR_HOME%/fanruanxxx/fr/classes	%FR_HOME%/fanruanxxx/fr/classes	作用： 工程调用的默认和自定义 class 文件  是否必须拷贝： 必须拷贝  有可能存在自定义的 class 文件，如不拷贝会影响工程正常使用
%FR_HOME%/fanruanxxx/fr/config	%FR_HOME%/fanruanxxx/fr/config	作用： 配置库相关文件  是否必须拷贝： 必须拷贝  平台调用的配置库 (finedb)，如不拷贝会影响

		响工程正常使用
%FR_HOME%/fanruanxx/fr/customLib	%FR_HOME%/fanruanxx/fr/customlib	作用： 工程定制/外部引入的 JAR 包  是否必须拷贝： 必须拷贝， 如不拷贝会影响模板访问
%FR_HOME%/fanruanxx/fr/logs	%FR_HOME%/fanruanxx/fr/logs	作用： swift 日志  是否必须拷贝： 用户自行决定  如果不拷贝，会丢失工程历史操作日志（即 logdb），「管理系统>平台日志」功能无数据  如对历史使用情况无要求， 无需拷贝
%FR_HOME%/fanruanxx/fr/plugins	%FR_HOME%/fanruanxx/fr/plugins	作用： 插件相关文件  是否必须拷贝： 必须拷贝， 如不拷贝会影响插件相关功能的实现
%FR_HOME%/fanruanxx/fr/tomcat_logs	%FR_HOME%/fanruanxx/fr/tomcat_logs	作用： Tomcat 通用日志位置  是否必须拷贝： 用户自行决定， Tomcat 历史访问日志， 可不拷贝
%FR_HOME%/fanruanxx/fr/backup	minio 文件服务器文件存放路径/backup	作用： 工程历史备份文件  是否必须拷贝： 用户自行决定， 可存储在原始的目录下备份， 可不拷贝
%FR_HOME%/fanruanxx/fr/assets/temp_attach	minio 文件服务器文件存放路径/assets/temp_attach	作用： 读写缓存存储路径  是否必须拷贝： 必须拷贝  该文件夹中存放着工程读写（图片）缓存，不拷贝可能会导致模板设置的背景图片预览为空
%FR_HOME%/fanruanxx/fr/assets/vcs	minio 文件服务器文件存放路径/assets/vcs	作用： FineReport 模板备份文件  是否必须拷贝： 用户自行决定  如果工程不需要回退历史开发的 FineRepo

		rt 模板，则无需拷贝
%FR_HOME%/fanruanxx x/fr/assets/其他文件	minio 文件服务器文件 存放路径/assets/其他 文件	作用：通用的共享持久化目录  是否必须拷贝：必须拷贝  工程正常运行所需要的文件，如不拷贝会 影响工程正常使用
%FR_HOME%/fanruanxx x/fr/reportlets	minio 文件服务器文件 存放路径/reportlets	作用：FineReport 模板存放目录  是否必须拷贝：必须拷贝，如不拷贝会导致 工程所有模板丢失
%FR_HOME%/fanruanxx x/fr/schedule	minio 文件服务器文件 存放路径/schedule	作用：定时调度生成的文件  是否必须拷贝：用户自行决定  如果不拷贝，定时任务挂载到决策平台的 结果报表无法访问

4.2 启动新工程

用户登录运维平台，点击「运维管理>项目管理」，点击新工程项目后的「组件管理」按钮。

找到 bi6 组件，点击「启动」按钮，即可启动新工程。如下图所示：



4.3 验证是否迁移成功

依次登录新集群所有工程节点：

- 1) 查看「管理系统>智能运维>集群配置>节点管理」中是否出现所有集群节点，说明



多节点集群全部配置成功

2) 查看各节点的工程配置是否相同，说明外接数据库配置成功



## 5. 启动原工程

由于此时原工程使用着新工程的外接数据库和集群组件，不能直接启动。

- 1) 使用 3.1 节的备份，将原工程还原到单机状态。
- 2) 再使用运维平台的「项目管理>组件管理」功能，启动原工程。

## 6. 迁移授权

若老工程进行了注册，一般需要将授权迁移到新工程。容器化工程推荐使用 [容器私有云认证](#) 方式。

### 6.1 销毁原工程授权

- 1) 参考 [授权迁移插件](#) 的使用步骤，向商务索要一个授权迁移插件，销毁原工程的旧授权。
- 2) 找到原工程%BI\_HOME%/webroot/WEB-INF/resources 文件夹下的销毁凭证。

### 6.2 获取新工程授权

参考 [容器私有云认证](#) 的使用步骤，向商务索要注册镜像文件 fanruan\_license\_server.tar，并参考文档为新集群工程进行授权。  
请在邮件中附上上一步获取的销毁凭证。

## 4.5.7 容器化集群迁移容器化集群

### 1. 概述

#### 1.1 版本

工程版本	功能变更
FineReport11.0	-
FineBI6.0	-

#### 1.2 应用场景

有时用户需要复用一个工程，或者迁移工程所在服务器。

本文方案，帮助用户将原有容器化部署的集群 FineBI 工程，迁移到另一个容器化部署的集群工程。

### 2. 新工程准备

#### 2.1 部署运维平台

容器化 FineBI 工程，需要通过运维平台部署。因此用户需要先部署一个 FineOps 运维平台。

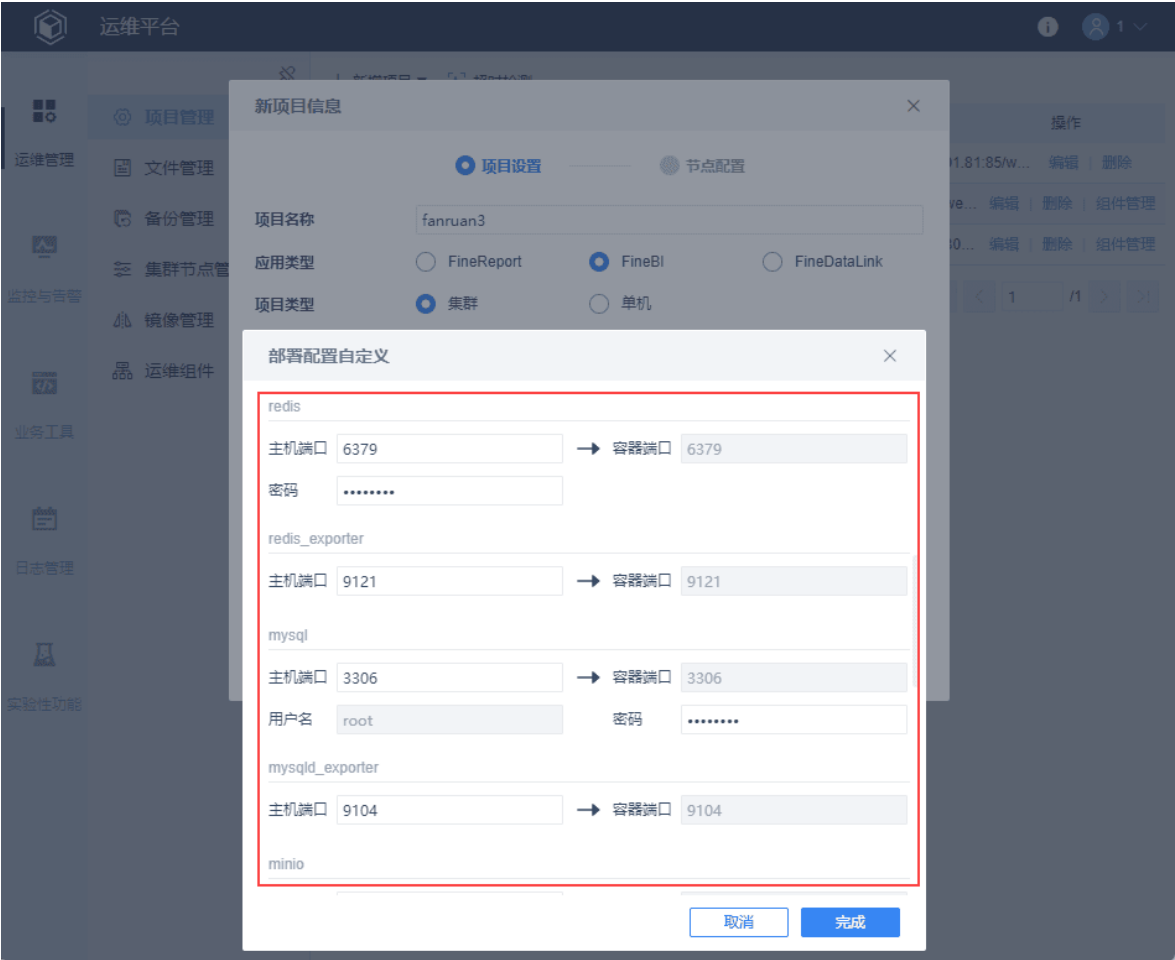
详情请参见：[部署运维平台](#)

#### 2.2 部署新工程

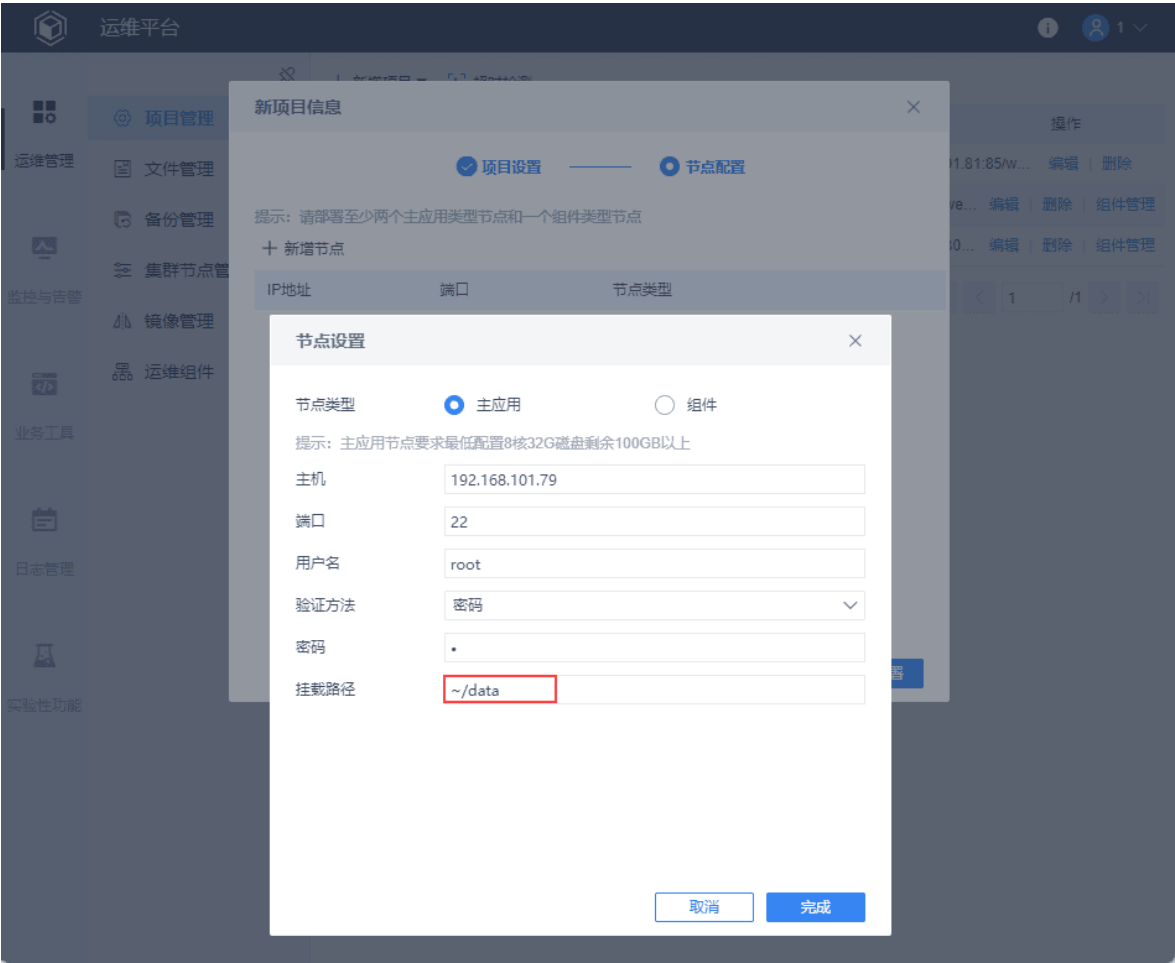
进行迁移前，用户需要先通过运维平台部署一个容器化集群工程。

详情请参见：[部署新项目-FineBI](#)、[部署新项目-FineReport](#)

注 1：请记录下 MySQL、Redis、minio 组件的相关信息，由于密码随机生成，请务必修改 MySQL、Redis、minio 组件的密码，方便后续使用。



注 2：请记录下 FineBI 工程的挂载路径，用户可自定义路径。这是工程文件的存放路径，后面步骤中需要用到。



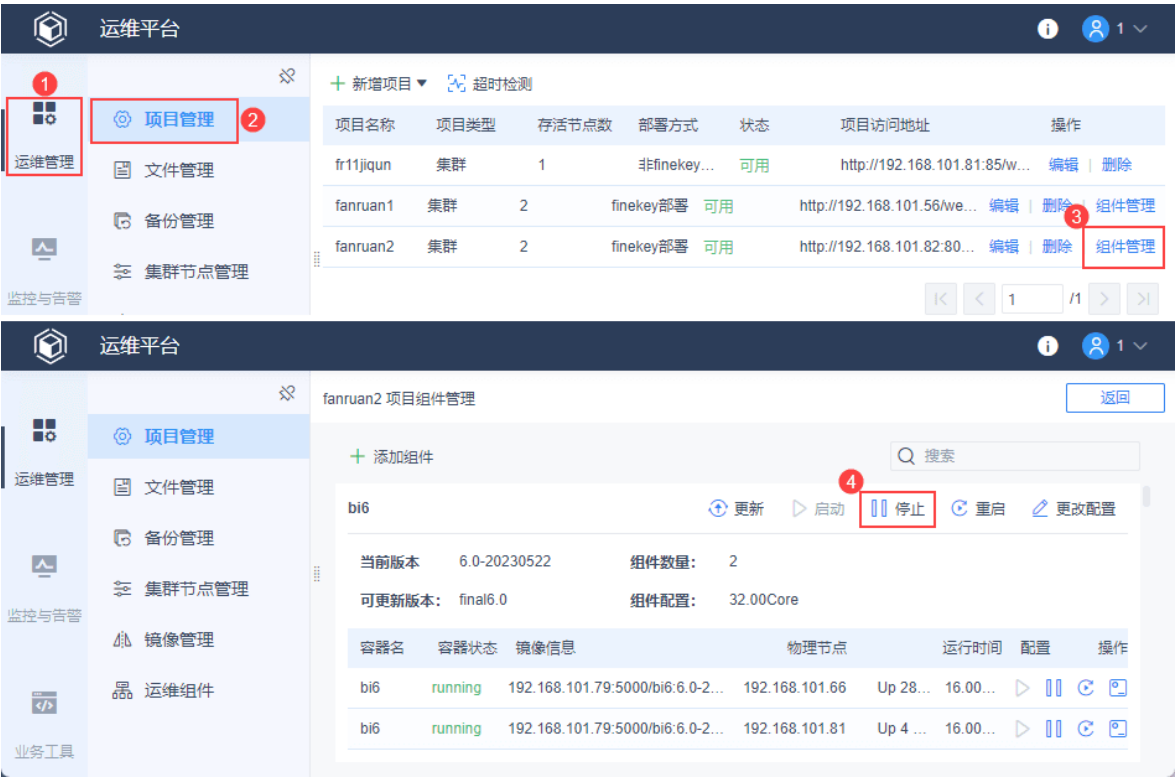
## 2.3 确认新工程服务器剩余磁盘空间

- 1) 检查原工程的大小。
- 2) 根据原工程大小，适度扩大新工程所在服务器节点剩余磁盘空间，建议大于原工程的两倍，最少剩余磁盘大小 500G。

2.4 关闭新工程

用户登录运维平台，点击「运维管理>项目管理」，点击新容器化部署的工程项目后的「组件管理」按钮。

找到 bi6 组件，点击「停止」按钮，即可关闭新集群的全部工程节点。如下图所示：



2.5 备份 config

请在新工程任一节点的挂载路径/fanruanxxx/bi 目录下，找到 config 文件夹，并另行备份。

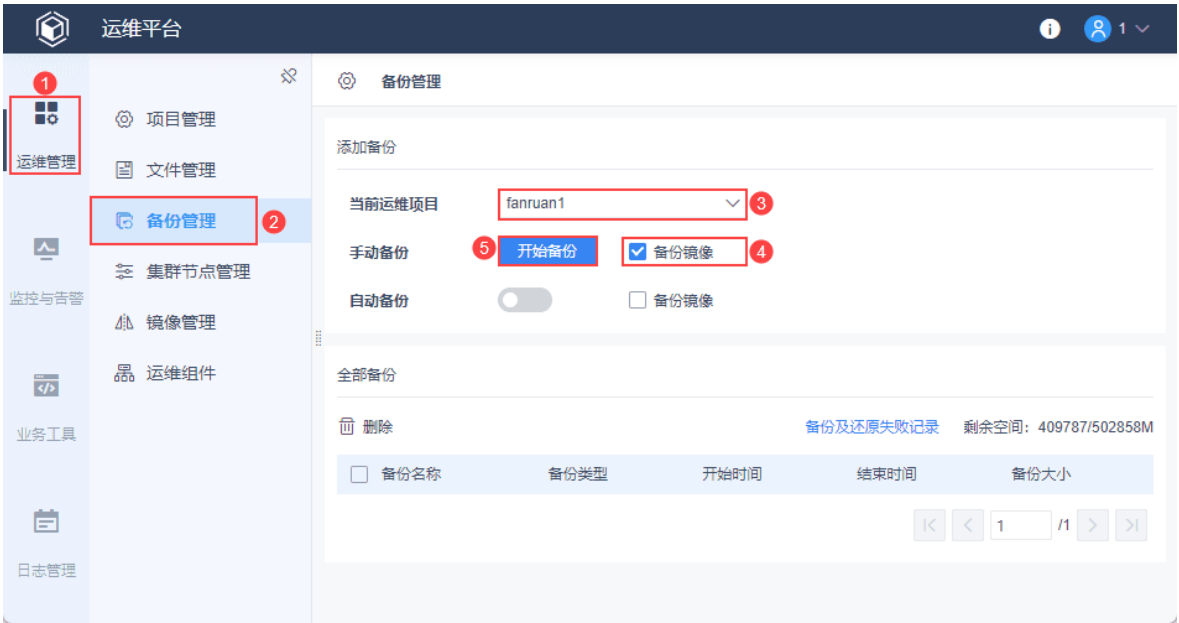
该文件夹中存放的 yaml 文件中，有着工程部署所用语句，后续运营维护时可能会需要使用/查找，请务必异地备份。

3. 原工程准备

3.1 备份原工程

在进行工程迁移前，为避免迁移失败，导致工程文件丢失无法回退等问题，请务必对原工程进行备份后再进行后续操作。

请使用运维平台的 [备份管理](#) 功能对原工程进行手动备份。

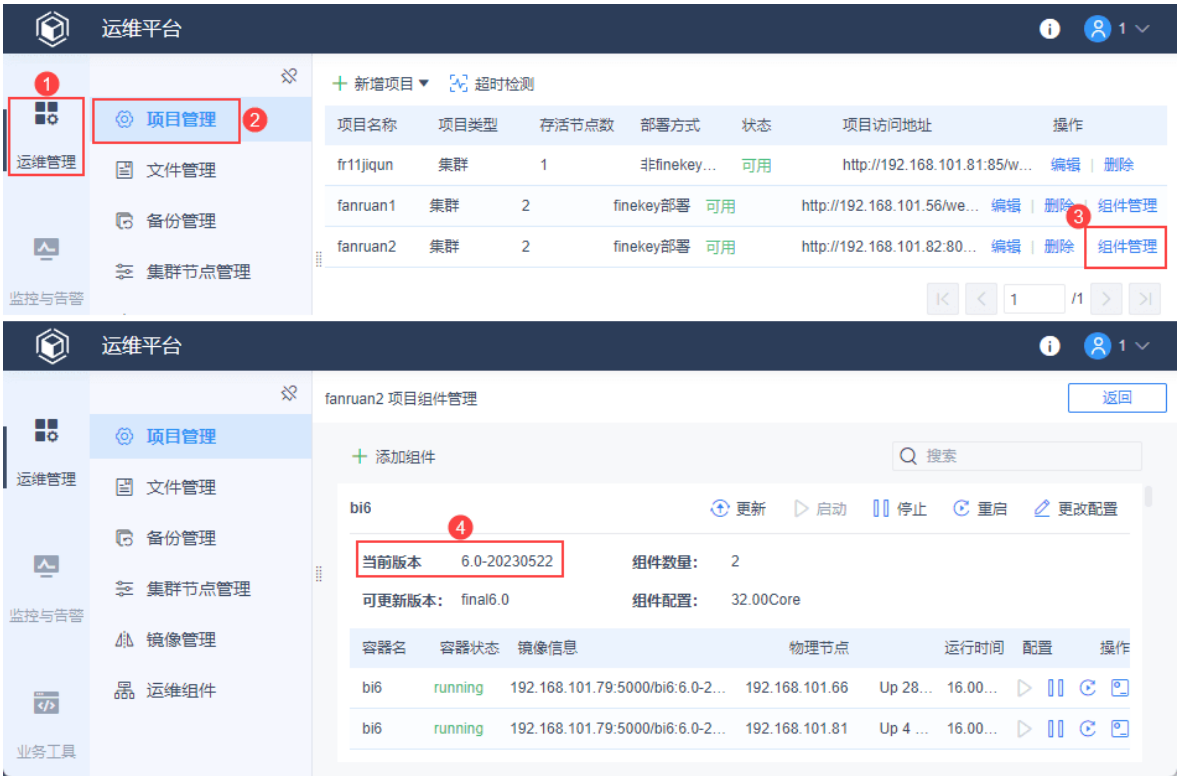


### 3.2 升级原工程

工程迁移的前提是，两个工程必须处于同一版本。

管理员登录运维平台，点击「运维管理>项目管理」点击新老工程后的「组件管理」，查看 bi6 组件的当前版本是否相同（精确到日期）。

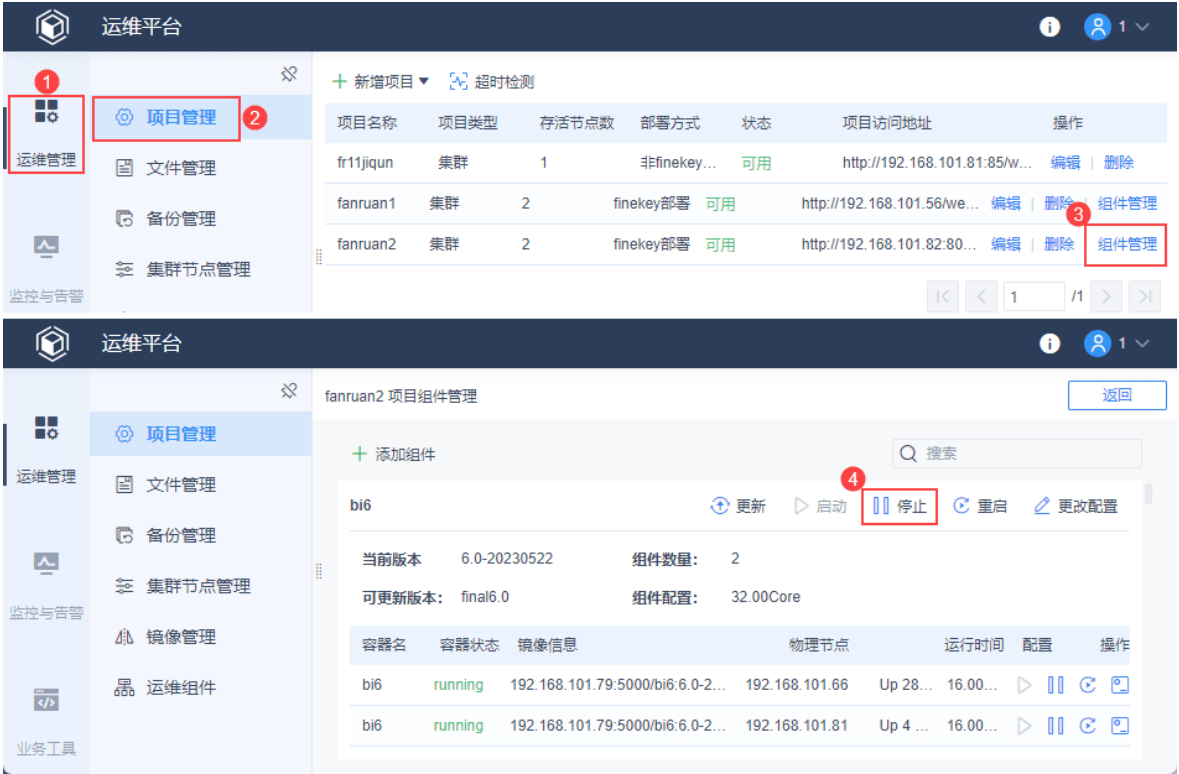
若两个工程版本不相同，可参考该文档更新工程版本：[外网升级指定版本/内网升级项目](#)



### 3.3 关闭原工程

用户登录运维平台，点击「运维管理>项目管理」，点击原工程项目后的「组件管理」按钮。

找到 bi6 组件，点击「停止」按钮，即可关闭新工程。如下图所示：



#### 4. 拷贝文件

##### 4.1 拷贝外接数据库数据

将原工程项目的 MySQL 组件的外挂目录数据，拷贝到新工程项目中的 MySQL 组件的外挂目录里。

注：MySQL 组件的外挂目录地址，一般在项目挂载路径/fanruanxxx/mysql 下。

```
[root@mail ~]# cd /home/bi/fanruan230529143543/mysql/
[root@mail mysql]# ll
总用量 4
drwxr-x--- 2 polkitd root    6 5月 29 14:35 conf
drwxr-x--- 6 polkitd root 4096 5月 29 14:35 data
drwxr-x--- 2 polkitd root   23 5月 29 14:35 log
[root@mail mysql]#
```

##### 4.2 拷贝 MinIO 文件服务器数据

将原工程项目的 MinIO 服务器文件，拷贝到新工程项目的 MinIO 服务器同位置。

注：MinIO 服务器的地址用户密码，可参考 2.2 节记录下的信息。

文件夹	说明
	作用：工程历史备份文件
minio 文件服务器文件 存放路径/backup	是否必须拷贝：用户自行决定，可存储在原始的目录下备份，可不拷贝
	作用：FineBI 数据表相关信息/FR 读写缓存存储路径
minio 文件服务器文件 存放路径/assets/temp_	是否必须拷贝：必须拷贝
attach	如不拷贝该文件夹会导致原始 Excel 丢失，或模板设置的背景

	图片预览为空
	作用：FineReport 模板备份文件
minio 文件服务器文件	是否必须拷贝：用户自行决定
存放路径/assets/vcs	如果工程未使用 FineReport 模板，或不需要回退历史开发的 Fi
	neReport 模板，则无需拷贝
	作用：通用的共享持久化目录
minio 文件服务器文件	是否必须拷贝：必须拷贝
存放路径/assets/其他文件	工程正常运行所需要的文件，如不拷贝会影响工程正常使用
	作用：FineReport 模板存放目录
minio 文件服务器文件	是否必须拷贝：必须拷贝，如不拷贝会导致工程所有 FineRepo
存放路径/reportlets	rt 模板丢失
	作用：定时调度生成的文件
minio 文件服务器文件	是否必须拷贝：用户自行决定
存放路径/schedule	如果不拷贝，定时任务挂载到决策平台的结果报表无法访问

4.3 拷贝工程文件

将原工程的必要文件，拷贝到新工程中。

- 1) 请提前检查原工程，是否修改了外挂目录位置、是否配置共享文件服务器，确保拷贝到正确的文件。
- 2) 新工程外挂文件所在位置，为 2.2 节记录的挂载路径。
- 3) 将以下文件，从原工程外挂目录，拷贝到新工程外挂目录中：

注 1：请勿直接拷贝整个外挂目录文件夹到新工程，部分文件不可以被覆盖。

注 2：请勿使用 3.1 节备份的文件，而是使用工程最终的文件进行拷贝操作。

• FineBI 需要拷贝的文件

外挂目录文件位置	说明
	作用：FineBI 抽取数据
%BI_HOME%/fanruanxxx	是否必须拷贝：
/bi6/bi-data	如果新老工程在同一服务器上，可重新-v 挂载原工程的 bi-data
	文件夹，或在新工程启动后重新抽取数据，无需拷贝

	如果新老工程不在同一服务器上，不建议拷贝，该文件夹一般数据量太大，拷贝困难，可在新工程启动后重新抽取数据
	作用：工程调用的默认和自定义 class 文件
%BI_HOME%/fanruanxxx/bi6/classes	是否必须拷贝：必须拷贝
	有可能存在自定义的 class 文件，如不拷贝会影响工程正常使用
	作用：工程定制/外部引入的 JAR 包
%BI_HOME%/fanruanxxx/bi6/customlib	是否必须拷贝：必须拷贝，如不拷贝会影响模板访问
	作用：swift 日志
	是否必须拷贝：用户自行决定
%BI_HOME%/fanruanxxx/bi6/logs	如果不拷贝，会丢失工程历史操作日志（即 logdb），「管理系统>平台日志」功能无数据
	如对历史使用情况无要求，无需拷贝
	作用：插件相关文件
%BI_HOME%/fanruanxxx/bi6/plugins	是否必须拷贝：必须拷贝，如不拷贝会影响插件相关功能的实现
	作用：Tomcat 通用日志位置
%BI_HOME%/fanruanxxx/bi6/tomcat_logs	是否必须拷贝：用户自行决定，Tomcat 历史访问日志，可不拷贝

• **FineReport 需要拷贝的文件**

外挂目录文件位置	说明
	作用：工程调用的默认和自定义 class 文件
	是否必须拷贝：必须拷贝
%FR_HOME%/fanruanxxx/fr/classes	有可能存在自定义的 class 文件，如不拷贝会影响工程正常使用
	作用：工程定制/外部引入的 JAR 包
%FR_HOME%/fanruanxxx/fr/customlib	是否必须拷贝：必须拷贝，如不拷贝会影响模板访问



作用：swift 日志

是否必须拷贝： 用户自行决定

%FR\_HOME%/fanruanxxx/fr  
/logs

如果不拷贝，会丢失工程历史操作日志（即 logdb），「管理系统>平台日志」功能无数据

如对历史使用情况无要求，无需拷贝

作用：插件相关文件

%FR\_HOME%/fanruanxxx/fr  
/plugins

是否必须拷贝： 必须拷贝， 如不拷贝会影响插件相关功能的实现

作用：Tomcat 通用日志位置

%FR\_HOME%/fanruanxxx/fr  
/tomcat\_logs

是否必须拷贝： 用户自行决定，Tomcat 历史访问日志，可不拷贝

#### 4.4 启动新工程

用户登录运维平台， 点击「运维管理>项目管理」， 点击新工程项目后的「组件管理」按钮。

找到 bi6 组件， 点击「启动」按钮， 即可启动新工程。如下图所示：



#### 4.5 验证是否迁移成功

依次登录所有工程节点：

- 1) 查看「管理系统>智能运维>集群配置>节点管理」中是否出现所有集群节点，说明

多节点集群全部配置成功

2) 查看各节点的工程配置是否相同，说明外接数据库配置成功



5. 启动原工程

用户登录运维平台，点击「运维管理>项目管理」，点击老工程项目后的「组件管理」按钮。

找到 bi6 组件，点击「启动」按钮，即可启动老工程。如下图所示：



6. 迁移授权

若老工程进行了注册，一般需要将授权迁移到新工程。容器化工程推荐使用 [容器私有云认证](#) 方式。

6.1 销毁原工程授权

1) 参考 [授权迁移插件](#) 的使用步骤，向商务索要一个授权迁移插件，销毁原工程的旧

授权。

2) 找到原工程%BI\_HOME%/webroot/WEB-INF/resources 文件夹下的销毁凭证。

## 6.2 获取新工程授权

参考 [容器私有云认证](#) 的使用步骤, 向商务索要注册镜像文件 fanruan\_license\_server.tar,

并参考文档为**新集群工程**进行授权。

请在邮件中附上上一步获取的销毁凭证。

## 4.6 集群运维项目扩容

## 4.7 常见问题

## 4.7.1 运维平台项目异常问题排查

### 概述

### 应用场景

本文简单介绍运维平台接入项目异常时的原因分析和解决方案。

### 部署新项目异常

The current server limit minimum number of cpu cores is 8

### 问题描述：

节点设置时，出现报错提示「The current server limit minimum number of cpu cores is 8」，如下图所示：



### 原因分析：

部署新项目时，FineOps 会对服务器的系统配置进行检测，不满足要求即会报错。

- FineReport 主应用节点要求最低配置 8 核 16G、磁盘剩余 50GB 以上
- FineBI 主应用节点要求最低配置 8 核 32G、磁盘剩余 100GB 以上
- FineDataLink 主应用节点要求最低配置 4 核 8G、磁盘剩余 50GB 以上
- 组件节点要求最低配置 4 核 8G、磁盘剩余 50GB 以上

### 解决方案：

请升级服务器配置后，再进行项目部署。

Server IP try to bind ports 9114,9100,9200,9080,9070 failed

### 问题描述：

项目部署时，出现报错提示「Server IP try to bind ports 9114,9100,9200,9080,9070 failed!」，如下图所示：

提示



项目部署信息出错

Server 121.37.182.66 try to bind ports 9114,9100,9200,9080,9070 failed!

确定

原因分析：

- 1) 项目服务器上的这些端口被占用。
- 2) 项目服务器的这些端口与 FineOps 所在服务器内网不互通。

解决方案：

- 1) 请确认默认的映射端口是否已被使用，以及项目服务器与 FineOps 所在服务器是否内网互通

端口占用和防火墙开放操作方法可参考：[端口占用检查与防火墙配置](#)。

- 2) 如端口已被使用，请安排好其他空闲端口。

在项目设置-部署配置中，选择「自定义端口、组件用户名密码」，并修改相关端口。

部署新项目

1 项目设置

2 节点配置

项目名称

fanruan1

应用类型

☒ FineReport

☐ FineBI

☐ FineDataLink

应用类型

☐ 集群

☒ 单机

默认部署组件

☒ 运维基础套件

自定义部署组件

☒ 外接数据库(MySQL) ☒ ElasticSearch

部署配置

☐ 使用默认配置

☒ 自定义端口、组件用户名密码

去设置

访问路径

webroot

配置后应用的访问链接为{域名/IP}/{访问路径}/decision

部署配置自定义

FineReport

主机端口 8080 → 容器端口 8080

主机端口 12100 → 容器端口 12100

mysql

主机端口 3306 → 容器端口 3306

用户名 root 密码 .....

mysqld\_exporter

主机端口 9104 → 容器端口 9104

elasticsearch

主机端口 9200 → 容器端口 9200

取消

完成

取消

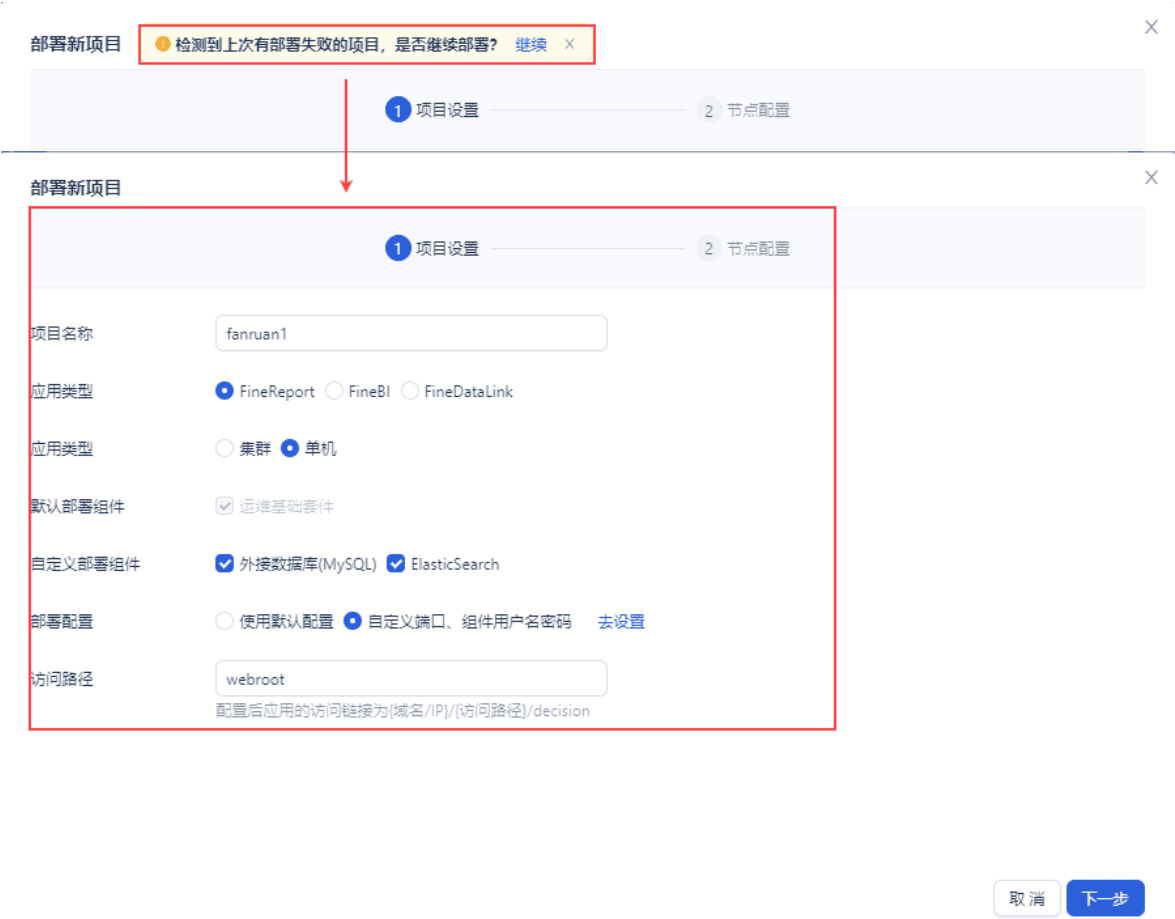
下一步

部署失败再次部署自动填充信息

V1.9.0 及之后版本支持该功能：

若运维平台上一次部署项目失败，再次部署新项目时，会出现提示「检测到上次有部署失败的项目，是否继续部署？」

若点击「继续」按钮，则自动填充上次部署的项目设置和节点配置，节约用户时间。



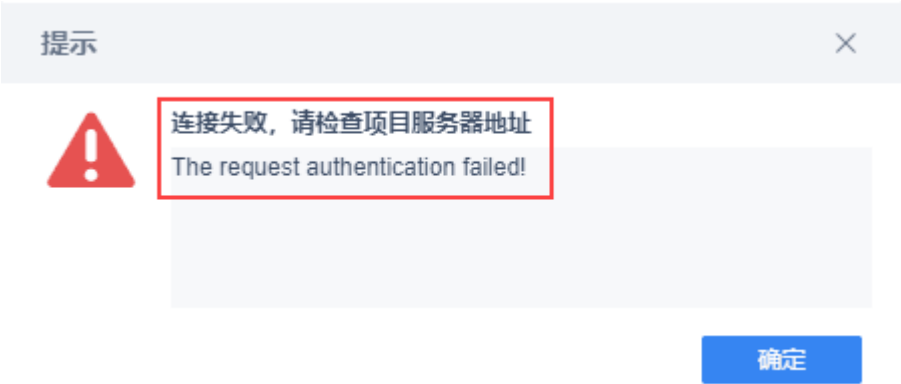
### 已接入项目异常

The request authentication failed

#### 问题描述：

项目已成功接入运维平台，项目可用，但状态突然显示为「不可用」

重新编辑测试接入时，报错：The request authentication failed



#### 排查方向一：

##### 原因分析：

运维接口密钥被修改，或该项目被其他 FineOps 运维平台接入了。

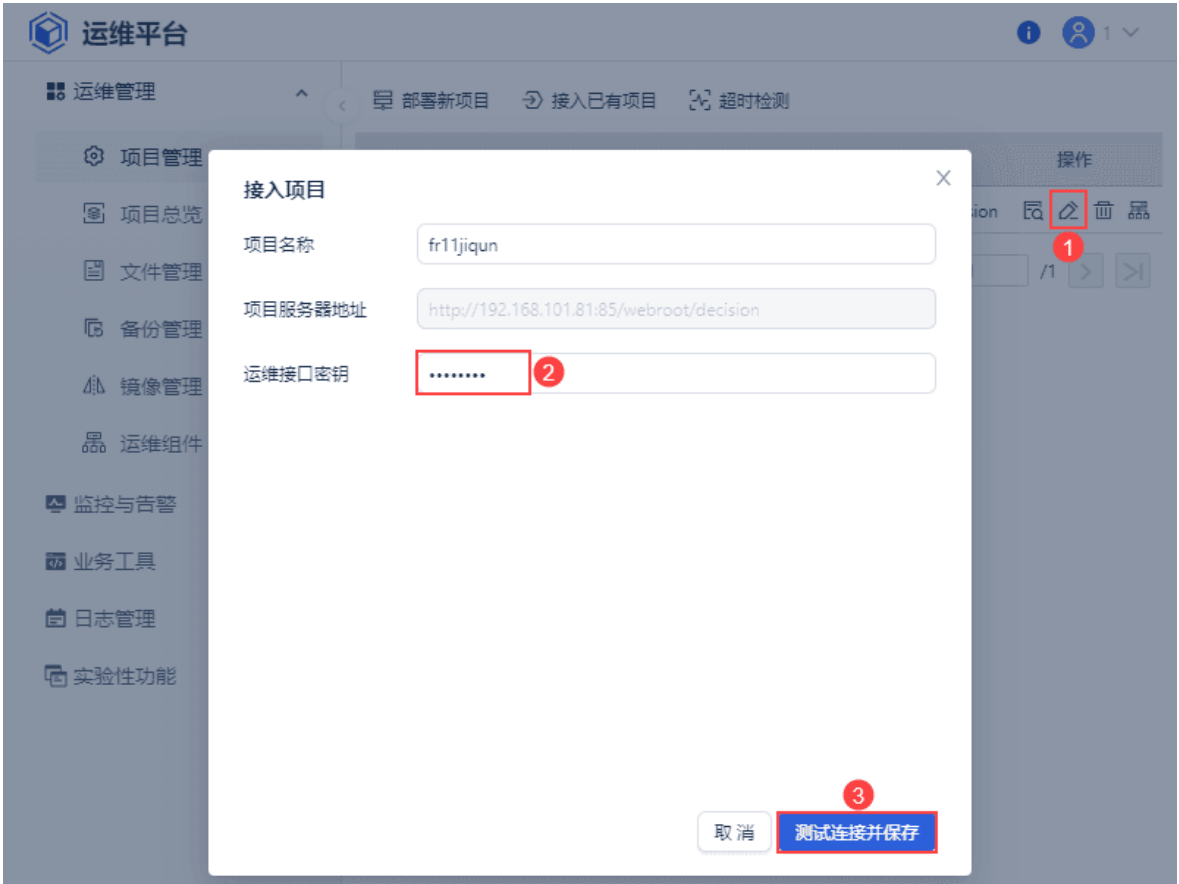
##### 解决方案：

- 1) 管理员登录帆软应用，重置运维接口密钥。





2) 管理员登录 FineOps，编辑运维项目，输入新的运维密钥，再次尝试连接。



排查方向二：

原因分析：帆软应用的配置库出现了变更，在迁移外接库的过程中，导入了其他配置，导致运维平台读取的项目 ID 发生了变更，认定为是其他项目。

解决方案：管理员需要在项目管理中，删除该运维项目，重新接入该项目。

## 4.7.2 MinIO 文件服务器简介

# 5. 运维管理

## 1. 概述

### 1.1 版本

运维平台版本	功能变更
V1.5.5	-
	新增「项目总览」功能
V1.5.8	下架「集群节点管理」功能

### 1.2 功能简介

运维平台提供定期、全面的系统检查功能，帮助用户快速、合理的修复各种系统运维隐患。



## 2. 索引

功能	说明	帮助文档
项目管理	运维平台作为一个运维工具，是对帆软的 FineReport、FineBI 等应用进行运维的。	
	因此在使用运维平台之前，需要将需要被管理的应用绑定到运维平台上。	<a href="#">项目管理</a>
	运维平台提供多项目管理的能力，可以接入多个单机或集群项目进行统一监控运维管理，解决多项目运维管理混乱的痛点	
项目总览	对于接入的运维项目，FineOps 运维平台为管理员提供多角度监	<a href="#">项目总览</a>

控和各类运维操作。

对于每个项目, 运维平台提炼出用户最关心的信息和最常用的操

作, 组合成了「项目总览」界面。

集群工程支持使用 MinIO 作为文件服务器。

文件管理

用户可通过「文件管理」功能, 界面化管理集群文件

[文件管理](#)

运维平台提供「备份管理」功能。

备份管理

对于运维平台管理的项目, 支持一键备份, 支持自动备份, 支持 [备份管理](#)

异机备份, 支持管理备份版本。

对于自有 K8S 平台的用户, 管理员可在「镜像仓库」中配置仓库地址、

镜像管理

用户名及密码, 以方便后续进行 exporter、应用及组件的镜像使用。

[镜像管理](#)

由于企业业务的复杂多变, 一个服务器下可能存在多容器、多应

用。需要给管理员提供一个汇总页面维护所有容器。

运维组件

应用不可用时, 需要给管理员提供快速恢复的方法和手段。

[运维组件](#)

运维平台提供「运维组件」功能。管理员登录运维平台, 点击「运维管理>

运维组件」, 支持对容器化部署项目中的运维组件进行管理。

## 5.1 项目管理

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.5.5	-
V1.5.9	管理项目，支持「项目总览」和「新增节点」功能
V1.6.0	部署方式措辞优化，展示为：非运维平台部署/运维平台部署
	1) 支持编辑项目「访问地址」
V1.8.0	2) 部署新项目、接入已有项目入口位置调整，更醒目
	3) 部署新项目，支持部署新 FineDataLink 集群项目
V1.9.0	非容器化部署的项目，支持「组件管理」操作，用于管理 exporter 组件和新增组件

#### 1.2 功能简介

运维平台作为一个运维工具，是对帆软的 FineReport、FineBI 等应用进行运维的。

管理员可将已有工程接入到运维平台，也可通过运维平台全新部署工程。

管理员可对接入运维平台的项目和其中的组件进行管理。



### 2. 新增项目

运维平台作为一个运维工具，是对帆软的 FineReport、FineBI 等应用进行运维的。

因此在使用运维平台其他功能前，需要先将这些应用项目接入运维平台。

新增项目	功能说明	帮助文档
部署新项目	未部署 FineReport、FineBI、FineDataLink 工程	<a href="#">部署新 FineReport 单机项目</a> <a href="#">部署新 FineReport 集群项目</a> <a href="#">部署新 FineBI 单机项目</a> <a href="#">部署新 FineBI 集群项目</a>
	可通过运维平台快速容器化部署工程并接入	<a href="#">部署新 FineDataLink 单机项目</a> <a href="#">部署新 FineDataLink 集群项目</a>

已容器化部署好 FineReport、FineBI 工程

[接入已有容器化项目](#)

需要将已有容器化工程接入运维平台

接入已有项目 已非容器化部署好 FineReport、FineBI 工程

[接入已有非容器化项目](#)

需要将已有非容器化工程接入运维平台



3. 管理项目

项目管理界面，显示所有已绑定的项目，显示项目的相关信息，包括：项目名称、项目类型、存活节点数、部署方式、状态、agent 版本状态、项目访问地址

支持管理员删除和编辑已有项目。如下图所示：

信息	说明
项目名称	新增项目时，设置的项目名称
项目类型	单机/集群
存活节点数	项目下存活的节点数量
部署方式	运维平台部署/非运维平台部署
状态	可用/不可用
项目访问地址	绑定的 FineReport、FineBI、FineDataLink 工程地址
操作	项目总览/编辑/删除/组件管理/新增节点



4. 修改项目访问地址

应用场景：

运维平台显示的默认访问地址是根据接入时的 IP 生成的。

而管理员可能为工程配置了域名，配置了内外网 IP，会导致自动生成的地址无效，无法访问。

此时管理员需要将自动生成的访问地址修改为实际项目访问地址。

操作步骤：

- 1) 鼠标悬浮到对应运维项目的访问地址上，出现「编辑」按钮，点击该按钮。
- 2) 支持设置项目服务器地址，请确保该地址为实际真实的访问地址。点击「确定」，即可修改成功。

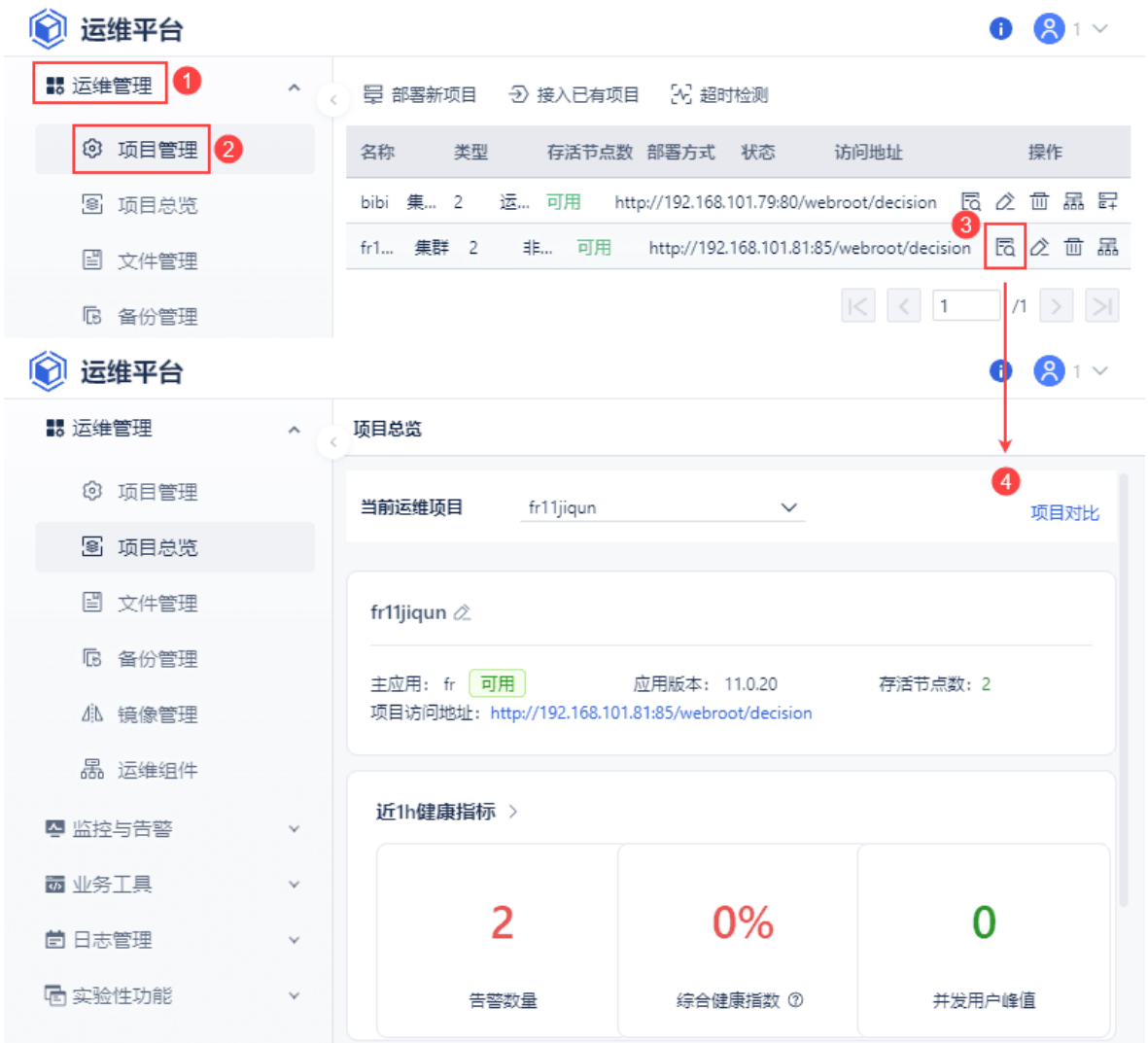


5. 项目总览

对于接入的运维项目，FineOps 运维平台为管理员提供多角度监控和各类运维操作。

对于每个项目，运维平台提炼出用户最关心的信息和最常用的操作，组合成了「项目总览」界面。

详情请参见：[项目总览](#)



## 6. 组件管理

一个项目中，可能存在多节点、多组件。

应用的某个组件出现故障，可能会导致应用整体不可用。需要给管理员提供快速恢复的方法和手段。

运维平台提供「组件管理」功能：

- 1) 支持对容器化部署的项目组件进行启动、停止、重启。
- 2) 1.9.0 及之后版本，非容器化部署的项目，支持「组件管理」操作，用于管理 exporter 组件和新增组件。

详情请参见：[组件管理](#)



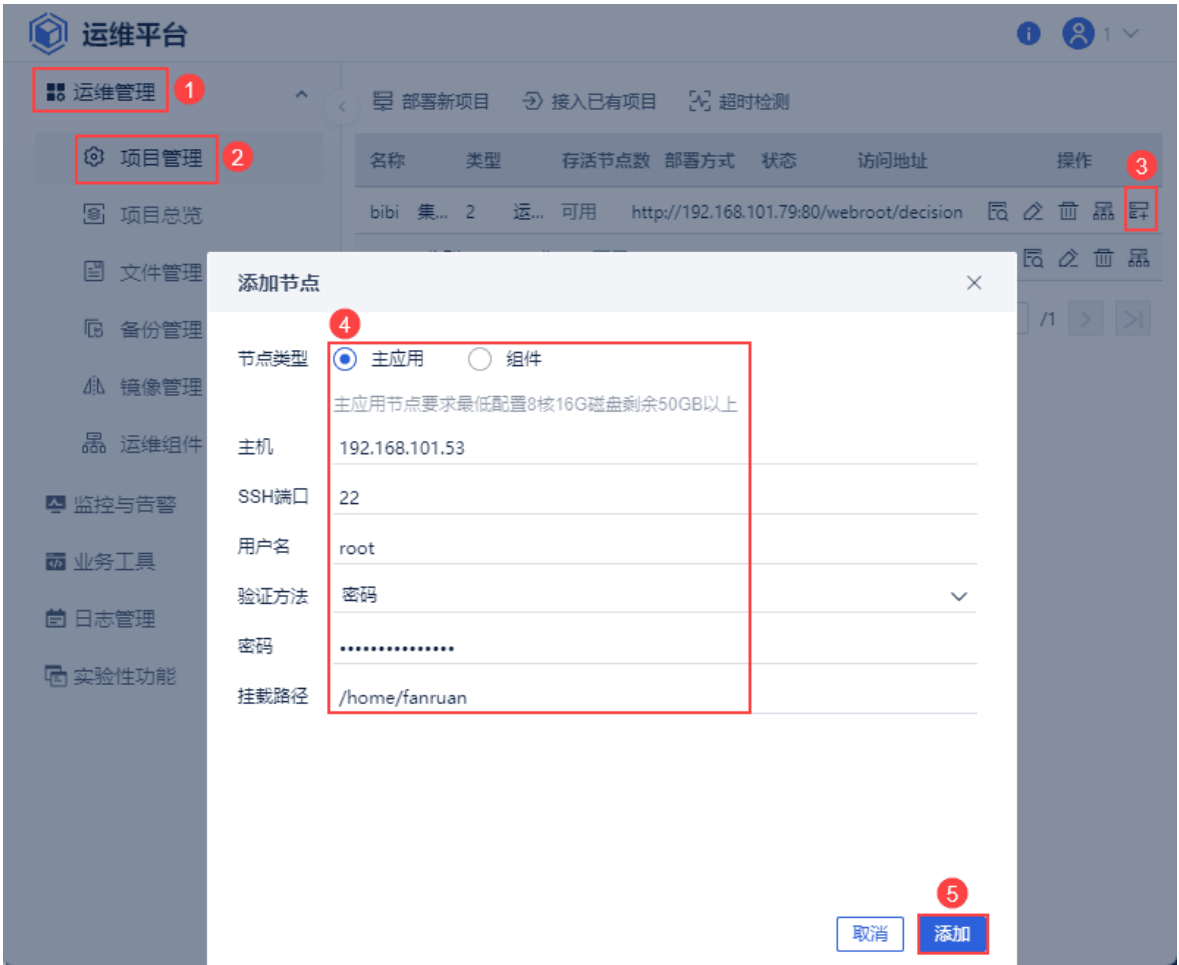


## 7. 新增节点

随着业务数据量和用户访问量的增加，原本部署的集群节点无法满足需求，需要通过添加集群节点，提供更高的性能和可扩展性。

管理员可使用「添加节点」功能，为运维项目接入服务器节点。

详情请参见：[集群运维项目扩容](#)



## 8. 超时检测

运维管理、业务工具、日志管理等功能，在使用之前，都需要对项目进行连接检测，若超时则无法使用。

项目管理页面支持配置「超时检测」时间，时间配置范围为 1-60，默认为 15 秒。  
超时无响应则认为连接异常，超时三次则认为应用状态异常，不再连接，请刷新后再试。



## 5.1.1 组件管理

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.5.5	-
V1.7.0	组件启停重启时，新增二次确认提示框，以防误操作
V1.9.0	非容器化部署的项目，支持「组件管理」操作，用于管理 exporter 组件和新增组件
V1.12.0	新增支持导出部署信息

#### 1.2 应用场景

一个项目中，可能存在多节点、多组件。

应用的某个组件出现故障，可能会导致应用整体不可用。需要给管理员提供快速恢复的方法和手段。

运维平台提供「组件管理」功能，支持对容器化部署的项目组件进行启动、停止、重启。

#### 1.3 使用前提

- 1) 支持对容器化部署的项目中的所有组件进行启动、停止、重启。
- 2) V1.9.0 及之后版本，非容器化部署的项目，支持「组件管理」操作，用于管理 exporter 组件和新增组件。



### 2. 组件管理

管理员登录运维平台，点击「运维管理>项目管理」，点击对应项目后的「组件管理」按钮，即可进入「组件管理」界面。如下图所示：



## 2.1 已有组件信息

组件管理页面，展示当前项目下的所有组件信息。

- 容器化部署的项目，包括部署项目时选择的所有组件
- 非容器化部署的项目，包括接入项目时部署的 exporter 组件，和使用运维平台部署的新组件

展示的组件信息包括：

信息	说明
当前版本	当前容器的安装版本
组件数量	该类型组件数量
可更新版本	是否存在可更新版本
组件配置	-
容器名	容器名称，方便识别用途
容器状态	是否启动，包括： 启动、未启动、启动中
镜像信息	容器的镜像信息
物理节点	容器 IP
运行时间	容器的持续运行时间
操作	包括多种操作：更新、启动、停止、重启、更改配置、终端、添加组件



## 2.2 添加组件

项目中可能存在部分组件初始未安装，在「组件管理」界面，点击「添加组件」，可部

署组件。

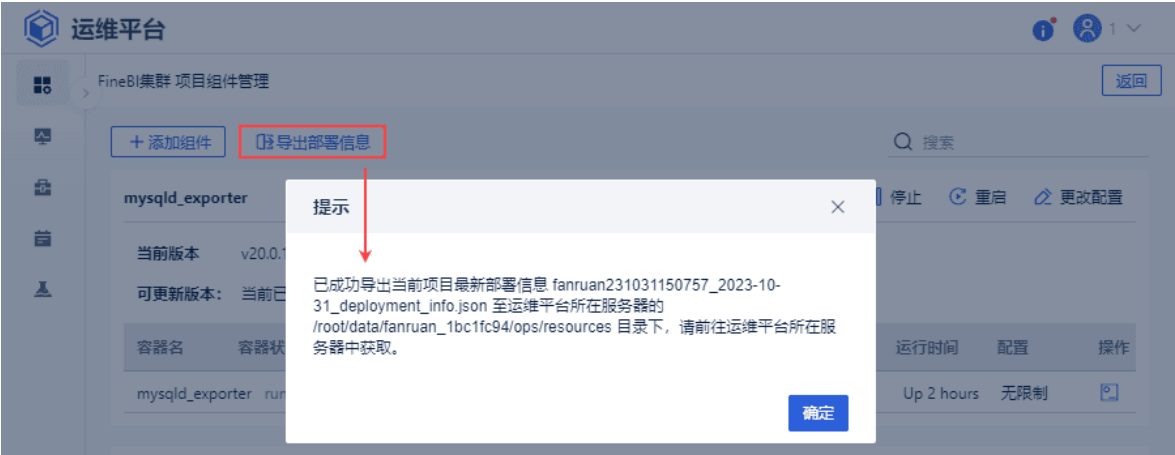
详情请参见：[添加组件](#)



### 2.3 导出部署信息

对于运维平台部署的组件/项目，在后期运维时，可能需要查阅原始部署信息。

可点击「导出部署信息」，导出 json 文件，用户可前往运维平台所在服务器中获取。



## 3. 组件操作

### 3.1 更新组件

若存在「可更新版本」，可点击「更新」按钮，即可对该组件进行更新。如下图所示：

若当前已为最新版本，则「更新」按钮灰化，无法使用。

详情请参见：[外网升级最新版本](#)、[外网升级指定版本/内网升级项目](#)



### 3.2 启停组件

支持对组件进行启动、停止、重启等操作，快速启停组件。

注：部分组件间存在关联关系，重启某一个组件，会触发其他组件的重启。



### 3.3 更改配置

提供「更改配置」功能，方便用户快速对项目进行配置调整修改和命令执行。

详情请参见：[组件配置变更](#)



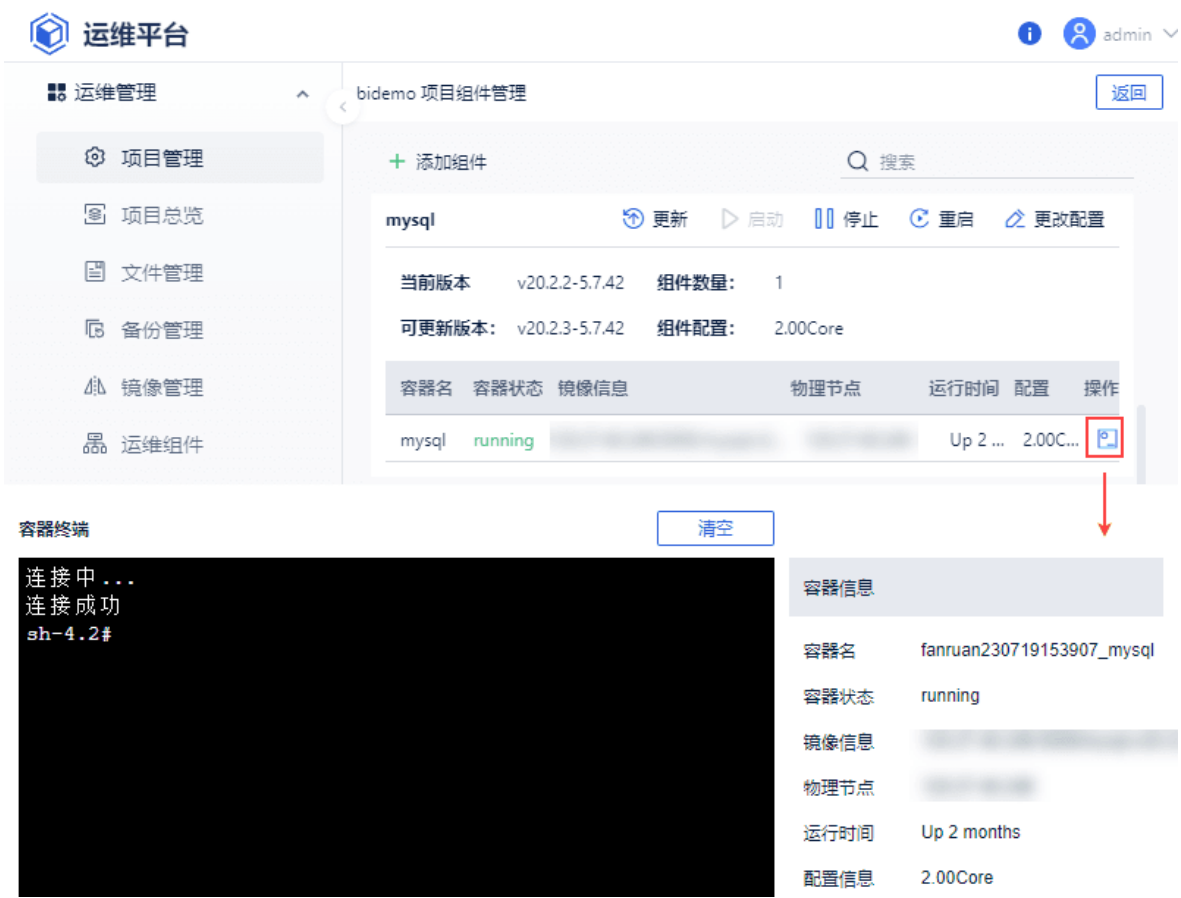
### 3.4 进入终端

尽管运维平台已经提供了很多界面化操作，但管理员仍有可能需要使用终端工具连接组件，对组件进行管理。

运维平台提供终端工具，支持快速进入组件终端。如下图所示：

左侧显示「命令行页面」，用户可输入命令进行查询/执行，支持一键清空命令行页面。

右侧显示该组件的具体容器信息。



注 1：必须在组件启动状态下，方可进入终端，否则提示「当前容器状态下不支持进入终端」。



注 2：单个运维平台，最多支持同时连接 3 个组件终端。超过 3 个时，打开提示「最多可连接 3 个终端，当前已达上限，请关闭其他终端后重试。」





注 3：若连接终端后，持续 10 分钟无命令操作，自动端口连接，提示「终端连接断开，请重新连接」。



## 5.1.2 组件配置变更

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.5.5	-

#### 1.2 应用场景

**场景一：需要合理分配组件占用 CPU 及内存**

小明在服务器上部署了一个容器化项目，但是他发现 elasticsearch 组件运行性能较差。

经过检查，发现是 FR 组件占据了全部 CPU，导致服务器无法给 elasticsearch 组件提供富裕内存。

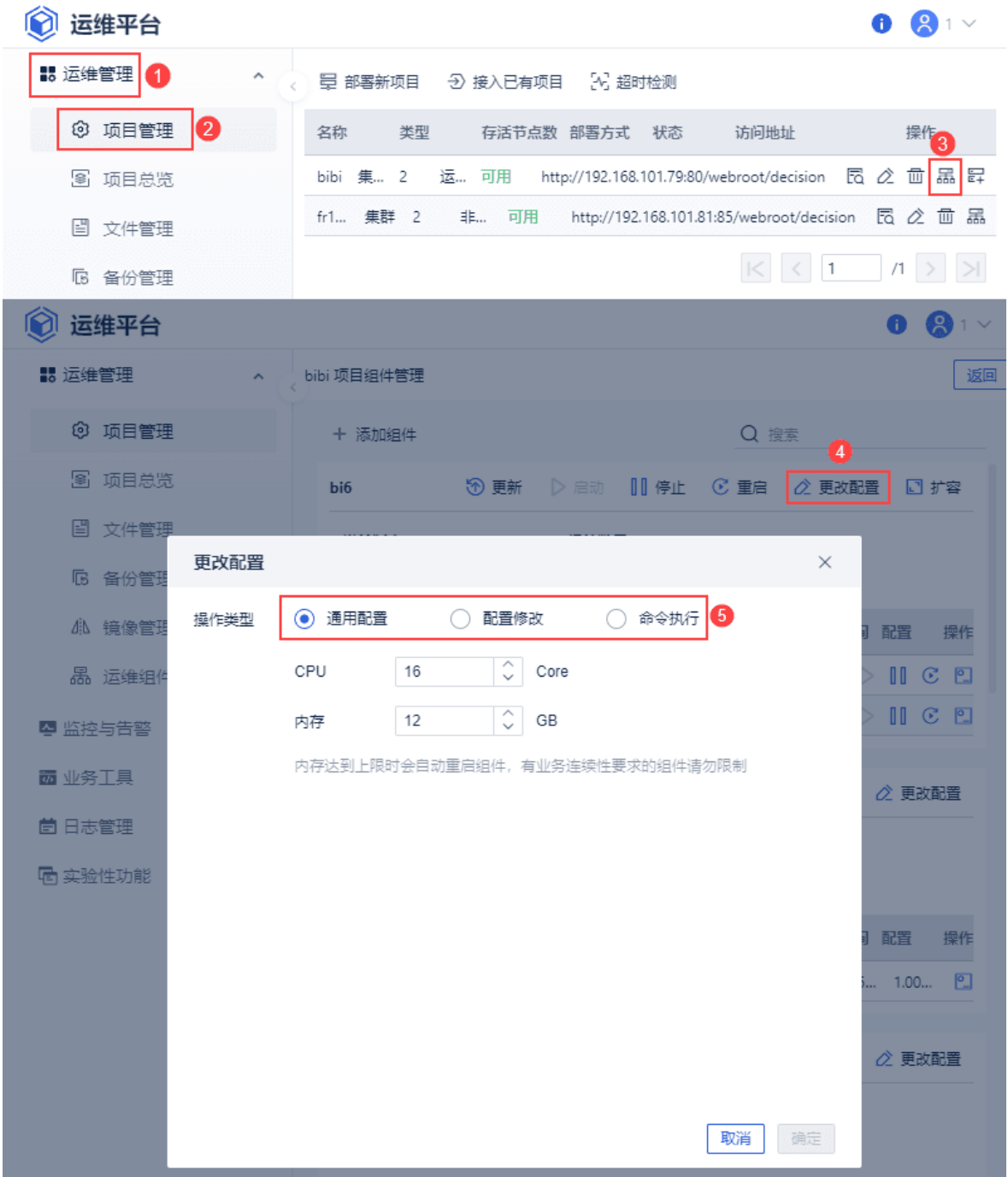
**场景二：需要界面化快捷修改组件配置**

小明部署好项目后，需要定期对 FR/BI 组件、外接数据库 MySQL、负载均衡 Nginx、状态服务器 Redis 进行一些维护操作，比如清理 Redis 缓存，调整 JVM 内存等等。而这些操作都需要使用命令语句执行，或者修改配置文件，操作起来十分麻烦。

#### 1.3 功能简介

FineOps 运维平台提供「更改配置」功能，方便用户快速对项目进行配置调整修改和命令执行。

管理员登录运维平台，点击「运维管理>项目管理」，点击目标项目对应的「组件管理」，点击目标组件对应的「更改配置」，即可进入该组件的更改配置操作界面。



1.4 使用前提

步骤	说明
容器化部署	「更改配置」作为「组件管理」的一个子功能，仅面向容器化部署的 FR/BI 工程，非容器化部署工程不支持本文相关操作。
接入项目	如需使用运维平台对 FR/BI 工程进行配置更新，首先需要完成两者的对接工作。
	详情请参见： <a href="#">接入已有容器化项目</a>

2. 通用配置

- 1) 选中「通用配置」，填入 CPU 或内存限制值。填写值说明：

- 支持对容器化部署项目中的所有类型组件限制 CPU 和内存
- 默认无限制，可输入大于 0 的整数。
- 组件最低运行环境要求请参见 [部署新项目「使用前提」章节](#)，推荐配置在最低运行要求的 2 倍以上。

2) 点击「确定」，自动重启组件，限制生效。重启组件可能影响项目正常运行，请勿在使用期间操作。

3) 限制生效后，内存达到上限时会自动重启组件，有业务连续性要求的组件请勿限制。

更改配置

操作类型

1

通用配置

配置修改

命令执行

CPU

2

15

Core

内存

48

GB

内存达到上限时会自动重启组件，有业务连续性要求的组件请勿限制

取消

3

确定

3. 配置修改

- 1) 对于 FineReport、FineBI、Redis、Mysql、Nginx 类型组件，支持进行「配置修改」。
- 2) 选择需要修改的参数名，输入参数值，点击「确定」，即可完成修改。如下图所示：

更改配置

1

操作类型

通用配置

配置修改

命令执行

参数名(key)

2

set JAVA\_OPTS

参数值(value)

3

-Djava.util.logging.config.file=/usr/local/tomcat/conf/logging.properties -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Duser.country=CN -Duser.language=

参数描述

JVM内存修改（通过修改setenv，如果没有就自己创建），例如 -Xms4g -Xmx4g -XX:MaxDirectMemorySize=1g -Dfineio.read\_mem\_limit=2 -Dfineio.write\_mem\_limit=1 -Dfineio.cache\_mem\_limit=1

4

取消

确定

3) 支持修改的参数如下表所示：

组件类型	参数名	默认参数值	说明
			JVM 内存修改（通过修改 setenv，如果没有就自己创建）
	set JAVA_OPTS		例如 -Xms4g -Xmx4g -XX:MaxDirectMemorySize=1g -Dfineio.read_mem_limit=2 -Dfineio.write_mem_limit=1 -Dfineio.cache_mem_limit=1
fr/bi			容器线程池最大数量：tomcat 配置——maxthreads，默认 500（范围 200 ~ 2000）
	set tomcat_thread_pool_num	acceptCount=100	线程池排队最大数量：tomcat 配置——acceptcount，默认 100（范围 100 ~ 1000）
	set chart_read_pool_num	8	图表线程池最大数量：应用配置——systemoptimizationconfig.chartdatathreadpoolnum，默认为 8（范围 8 ~ 50）
	max_allowed_packet		客户端或者数据库服务器端能够接受的最大数据量，例如 128 M
MySQL	character_set_server		数据库服务器的编码(mysqlId)，例如 utf8mb4，一般需要和 collation-server 一起修改
	collation-server		数据库服务器的默认排序规则，例如 utf8mb4_bin，一般需要和 character_set_server 一起修改

New Help		
Nginx	client_max_body_size	100M 允许用户最大上传数据大小, 根据业务需求调整上传文件大小限制, 默认值为 100M
	proxy_buffer_size	自定义 例如 64k 设置缓冲区的大小为 size
	proxy_buffers	自定义 例如 32 64k 每个连接设置缓冲区的数量和大小, proxy_buffers [number] [size]
	proxy_busy_buffers_size	自定义 例如 64k 当开启缓冲响应的功能以后, 在没有读到全部响应的情况下, 写缓冲到达一定大小时, nginx 一定会向客户端发送响应, 直到缓冲小于此值
	proxy_connect_timeout	自定义 例如 60s nginx 连接到后端服务器的超时时间, 是建立连接的时间, 而不是等待后端返回响应的时间
		读取超时, 默认 60s
		如果在超时时间内服务器未返回任何数据, 视为超时
	proxy_read_timeout	自定义 例如 60s 如果没有大数据量计算或导出的模板, 则建议配置不超过 100s
		如果有大数据量计算或导出的模板, 则根据最长耗时的模板时间进行配置
		写入超时, 默认 60s
Redis		如果在超时时间内服务器未收到数据表示超时, 视为超时
	proxy_send_timeout	自定义 例如 60s 如果没有大数据量计算或导出的模板, 则建议配置不超过 100s
		如果有大数据量计算或导出的模板, 则根据最长耗时的模板时间进行配置
	maxmemory	自定义 例如 2gb redis 内存修改
	maxmemory-policy	自定义 例如 volatile-lru redis 内存策略修改

stop-writes 自定义

-on-bgsave

配置 redis 快照出错时是否停止数据写入

-error 例如 no

4. 命令执行

- 1) 对于 Mysql 类型组件，支持进行「命令执行」。
- 2) 选择需要修改的参数名，输入参数值，点击「确定」，即可完成修改。如下图所示：

更改配置

操作类型

通用配置

配置修改

1

命令执行

命令名

2

set GLOBAL read\_only = {}

参数值(value)

3

false

命令描述

开启关闭只读模式GLOBAL read\_only，例如false

取消

4

确定

- 3) 支持修改的参数/配置如下表所示：

组件类型	命令名	参数值	说明
Mysql	flush hosts	无	清理 hosts 文件，控制台执行 flush hosts
	set GLOBAL net_write_timeout = {}	自定义 例如 1000	修改写入超时时间 net_write_timeout
	set GLOBAL read_only = {}	自定义 例如 false	开启关闭只读模式 GLOBAL read_only
	set global max_connect_errors = {}	自定义 例如 1000	修改最大连接超时个数 global max_connect_errors

## 5.1.3 添加组件

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.5.5	-

#### 1.2 应用场景

单机转集群工程时，需要准备很多组件，例如文件服务器、负载均衡、状态服务器等

#### 1.3 功能简介

运维平台支持为容器化部署的单机工程，添加此类集群组件，帮助用户快速升级集群。



### 2. 使用前提

#### 2.1 接入项目

如需使用运维平台对项目组件进行更新，首先需要完成两者的对接工作。

- 详情请参见：[接入已有容器化项目](#)

#### 2.2 配置镜像仓库

如需使用运维平台添加项目组件，需要有组件镜像包用于安装。

因此需要先为运维平台配置一个镜像仓库，用于存放镜像包。

- 1) 使用自带镜像仓库：在部署运维平台时，会默认安装一个仓库组件 registry。镜像仓库地址为运维平台 IP:5000，无需填写用户名、密码、SSL，点击「保存」即可。
- 2) 使用用户自有仓库：用户需要自行提供仓库地址、用户名、密码、SSL，点击「保存」即可。





## 2.3 存在未部署组件

若该项目中所有组件均已完备部署，则不会出现「添加组件」的按钮。

## 3. 操作步骤

### 3.1 添加组件

管理员登录运维平台，点击「运维管理>项目管理」，点击目标项目对应的「组件管理」。



点击「添加组件」，选择组件安装所在的节点，选择需要部署的组件，设置组件的端

口、用户名和密码等，点击「开始部署」

注：支持安装的组件环境要求和默认端口等信息，请参考：[部署新项目](#)。



### 2.3 部署完成

开始部署后，界面化展示部署进程。此时可以点击「隐藏至后台进行」，管理员可以去进行其他工作。

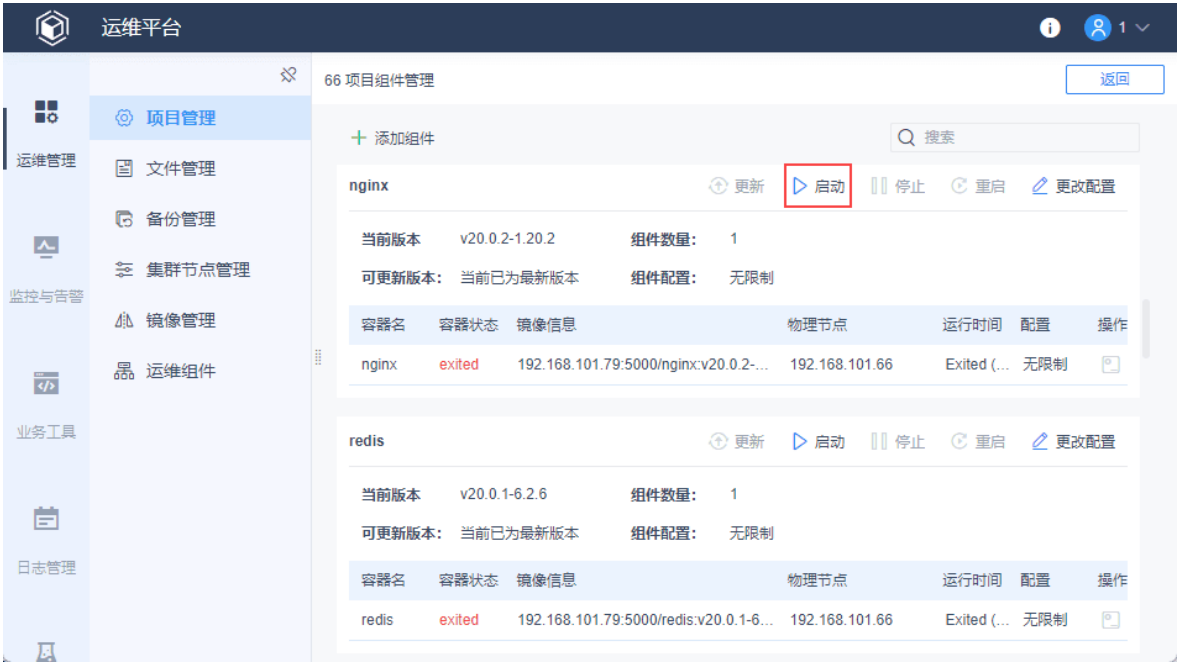


部署完成后，出现提示「组件部署成功」，如下图所示：



## 2.4 启动组件

此时回到「组件管理」界面，出现新部署的组件。这些组件会默认不启动，需要手动启用。



## 5.1.4 为运维项目开启 SSL

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.13.0	-

#### 1.2 应用场景

要启用 HTTPS 协议，我们需要使用有效的 SSL 证书。SSL 证书中包含身份验证信息，帮助保障用户的通信加密和安全。

在构建一个安全的工程中，证书起到关键作用，它保证了客户端与服务器之间的交互安全性。

本文将介绍如何通过简单的前端界面操作，为容器化部署的 FineReport、FineBI 运维项目配置 SSL 证书，以满足使用 HTTPS 访问帆软应用的需求。

这样做可以确保 FineReport、FineBI 工程使用安全的通信协议，并给用户提供加密保护。

### 2. 使用前提

#### 2.1 确认工程部署方式

运维平台前端开启运维项目的 SSL，是要对运维项目中的 Nginx 组件进行配置。因此需要满足以下两个条件：

- 1) 运维项目的部署方式必须是「运维平台部署」，「非运维平台部署」的项目不支持本文操作。
- 2) 运维平台部署的单机项目，一般不配置 Nginx 组件。

管理员需在「组件管理」中添加 Nginx 组件，自行修改 nginx.conf，使其成功对接工程节点，方可进行 SSL 配置。



#### 2.2 准备证书

在配置证书前，首先需要申请证书。

请自行向 CA 厂商购买证书，证书认证一般都是由 VeriSign、GlobalSign 等国际公  
认的 CA 机构认证。

需要准备的证书相关文件和要求如下：

内容	要求
证书文件	请准备好你的证书 证书格式为 .crt 或 .pem（nginx 仅支持这两种）
密钥文件	请准备好你的证书对应的密钥文件 证书格式为 .key

2.3 确认证书、域名、IP 一一对应

1) 确认证书对应的域名

- 将证书上传到拥有 JDK 环境的计算机上，进入证书所在目录。
- 使用语句打印查询证书信息：keytool -printcert -file name.pem（请将 name.pem 换成你的证书名称）。
- 确认证书绑定的域名：在返回值中，「Owner: CN」/「所有者: CN」后的值，即为该证书对应的域名。

```
[root@ecs-new-0017 ~]# cd /home/ssl
[root@ecs-new-0017 ssl]# keytool -printcert -file demo.fanruan.com.pem
Certificate[1]:
Owner: CN= n.com
```

2) 确认域名绑定 IP

请将证书对应的域名，与运维项目中的 Nginx 组件所在服务器的 IP，互相绑定。

由于不同服务器的操作不尽相同，帆软不提供此操作的指导文档，请自行联系你的服务器厂商咨询。

绑定成功后，管理员可 ping 域名，确认返回值中的 IP 是否正确。

```
[root@ecs-new-0017 ssl]#
[root@ecs-new-0017 ssl]# ping opsdemo.fanruan.com
PING opsdemo.fanruan.com (106.11.122) 56(84) bytes of data.
64 bytes from : icmp_seq=1 ttl=50 time=17.9 ms
64 bytes from : icmp_seq=2 ttl=50 time=17.8 ms
64 bytes from : icmp_seq=3 ttl=50 time=17.8 ms
64 bytes from : icmp_seq=4 ttl=50 time=17.8 ms
64 bytes from : icmp_seq=5 ttl=50 time=17.8 ms
64 bytes from : icmp_seq=6 ttl=50 time=17.7 ms
```

2.4 准备端口

请准备一个运维项目中的 Nginx 组件所在服务器的空闲端口，用于后续访问运维项

目。

请确保端口防火墙已开放，可正常访问。

如希望在访问时不添加端口，直接通过域名访问，请使用 HTTPS 的默认端口 443 。

本文示例使用 443 端口。

### 3. 操作步骤

#### 3.1 配置证书

- 1) 管理员登录 FineOps 运维平台，点击「运维管理>项目管理」，点击指定项目后的「组件管理」。
- 2) 在「组件管理」界面，找到 Nginx 组件，点击「更改配置」。



- 3) 切换到 SSL 配置界面，依次上传第二章准备的证书和密钥文件，填入域名和端口。
- 4) 点击「确定」。



#### 3.2 校验配置

1) 点击保存后，会弹出提示「开启 SSL 将重启项目的 Nginx 组件，期间项目将持续几分钟无法访问，重启完成后 https 访问地址为 XXX（对应域名后的 url），是否继续？」

2) 点击「继续并重启」，运维平台会自动进行以下检测和配置，请耐心等待：

- 检测所填端口是否可用，如不可用，配置失败。
- 将证书和密钥文件上传到运维项目 Nginx 组件的/nginx/conf/ssl 目录下
- 使用原命令新建替换 Nginx，并新增 443 端口，环境变量新增 SSL 相关配置。
- 自动重启 Nginx，启动完毕后即可实现 HTTPS 访问。

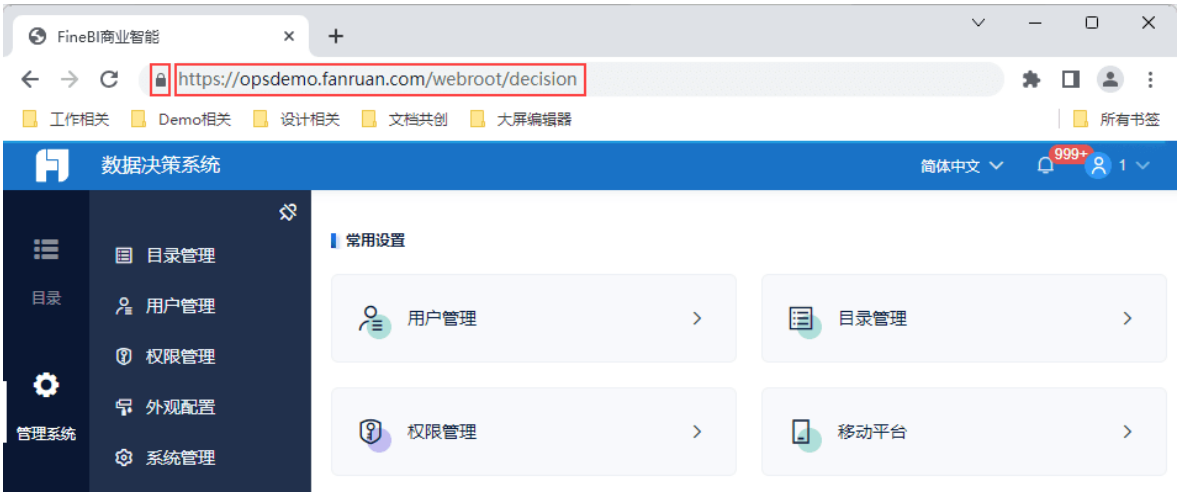


### 3.3 效果预览

在浏览器输入上文提示中的地址，即 `https://your_domain:port/webroot/decision`，确保能够成功访问，如果网页地址栏出现小锁标志，表示 SSL 配置成功。

- `your_domain` 是你的域名。
- 端口为上文配置的端口，如果端口号为 443，可省略不写。

注：原 http 请求会自动跳转 https。





## 5.2 项目总览

## 5.3 文件管理

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.5.5	-
V1.12.0	1) 支持新建文件夹
	2) 支持批量上传文件
	3) 支持上传.fvs 和.mfvs 格式的文件
	4) 上传同名文件时，新增覆盖提示

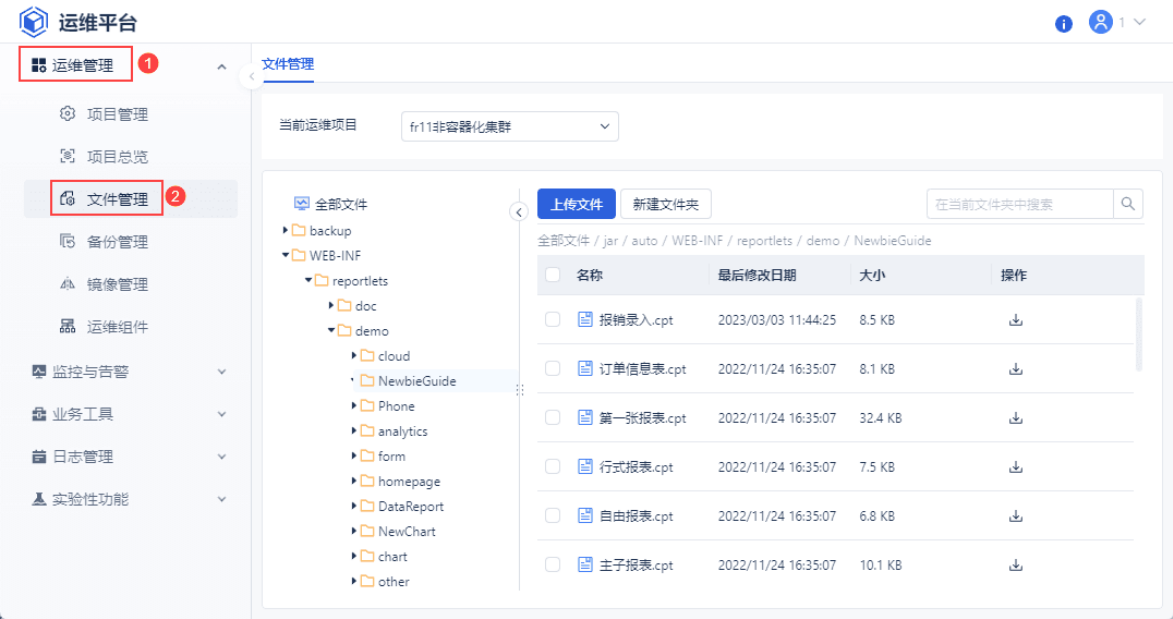
#### 1.2 应用场景

对于一个项目，管理员需要对项目中的文件进行管理。例如：

- 将模板、Excel 文件上传到文件服务器。
- 单机切换集群时，需要将单机工程文件上传到集群文件服务器上。
- 将定时调度结果下载到本地，将工程模板文件下载备份。
- 对于复杂目录结构，根据文件名称快速搜索查找。

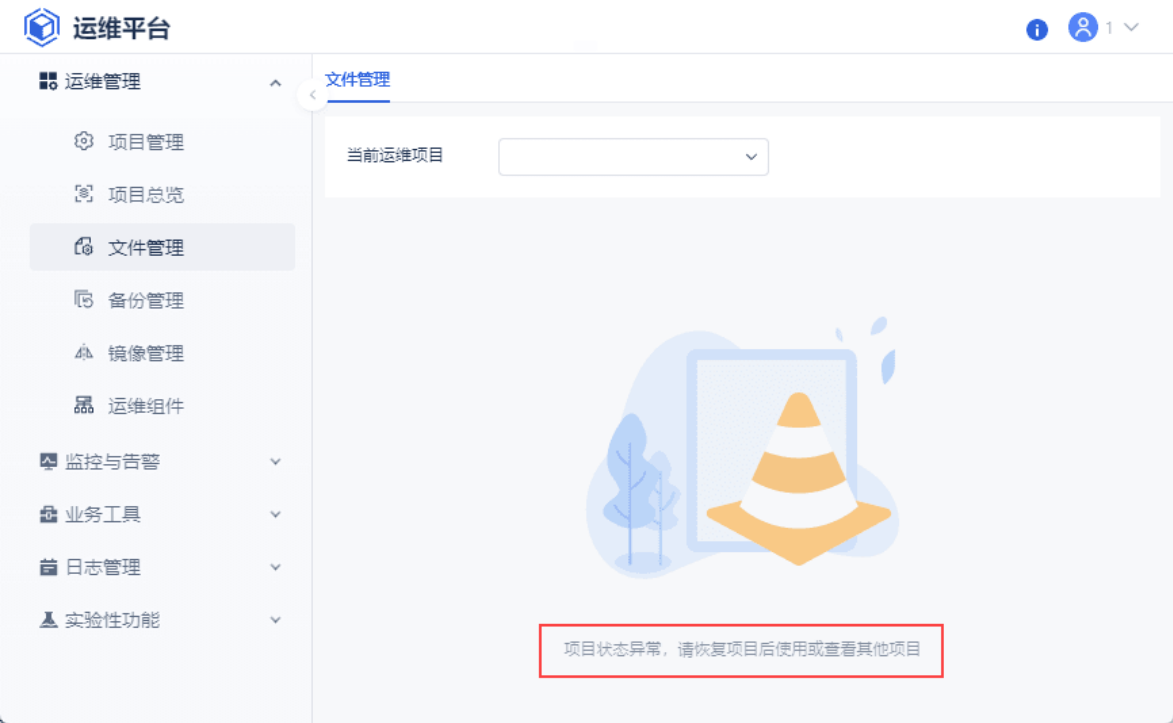
#### 1.3 功能简介

运维平台提供「文件管理」功能，让用户可通过前端界面化对文件服务器中的文件进行查找、下载、上传等操作。



1.4 使用前提

若项目状态不可用，提示「项目状态异常，请恢复项目后使用或查看其他项目」，无法使用文件管理功能。



2. 界面概览

管理员登录运维平台，点击「运维管理>文件管理」。在「文件管理」页面，选择运维项目。

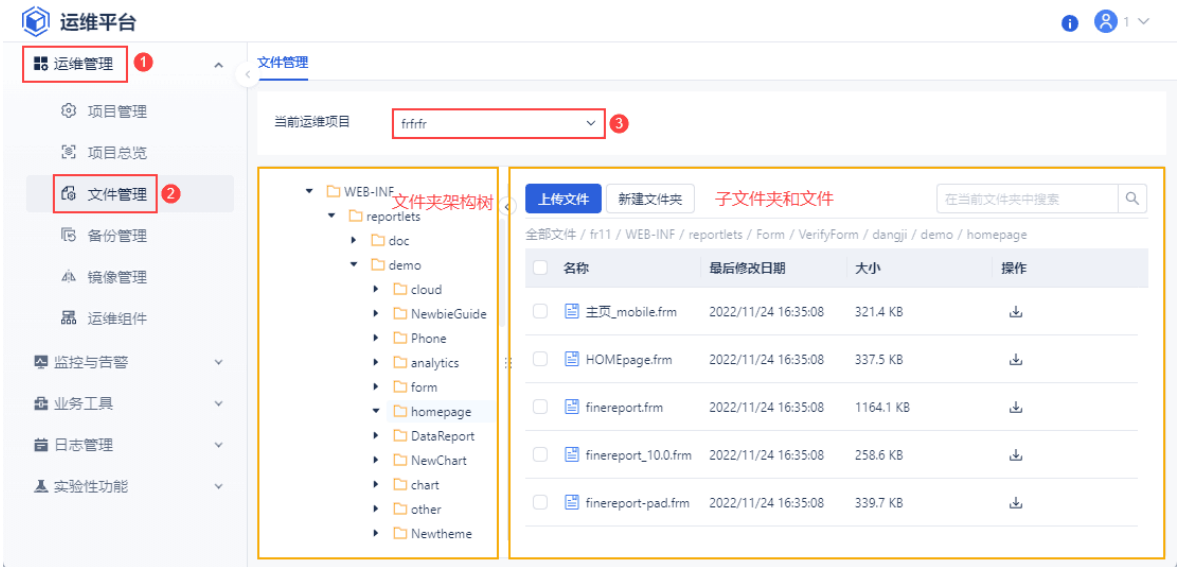
左侧显示文件夹架构树，右侧显示选中的文件夹下的子文件夹和文件。

在选中的文件夹中，可输入文件名称搜索查找文件。

对于单个文件，显示每个文件的名称、最后修改日期、大小。

支持显示的文件包括：

项目类型	显示的文件
容器化部署的项目	显示帆软项目外挂目录中的文件，具体可查看 <a href="#">部署新项目</a> 中「设置项目挂载目录」章节的目录内容清单
非容器化部署的项目	显示帆软工程%Tomcat_HOME%\webapps\webroot 文件夹中的内容
集群 -	显示帆软集群文件服务器中存放的文件
	手动拷贝文件到文件服务器后，需要清除浏览器缓存，方可在文件管理中看到
	存放在 MinIO 文件服务器中的文件，不显示最后修改日期



### 3. 新建文件夹

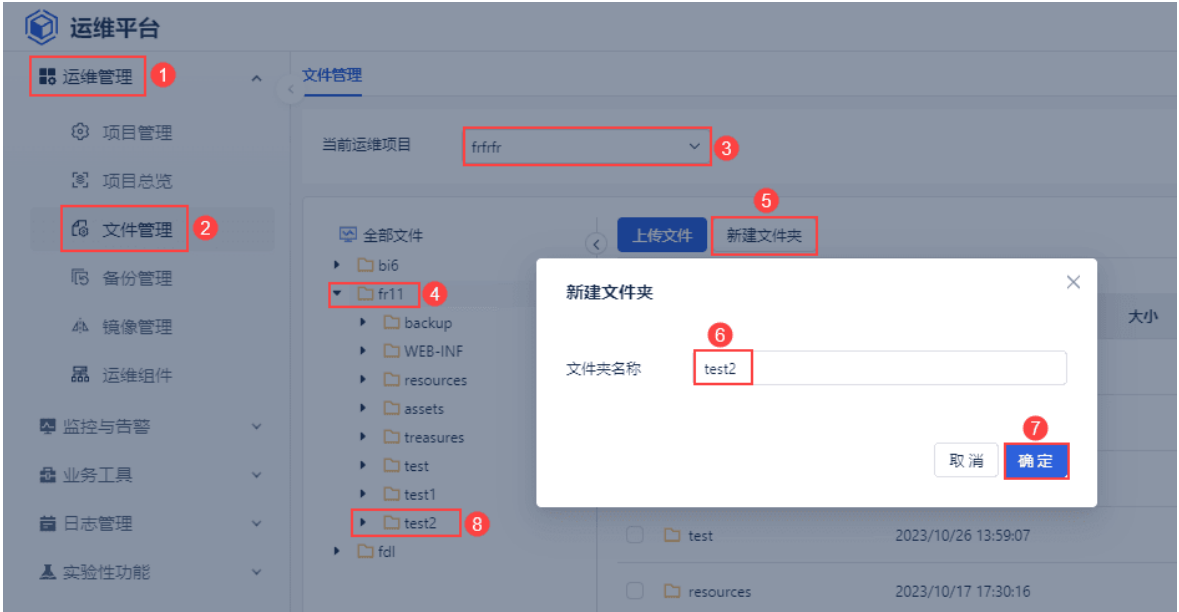
管理员可在文件管理中新建文件夹。

- 1) 管理员登录运维平台，点击「运维管理>文件管理」。在「文件管理」页面，选择运维项目。
- 2) 选中父目录，新文件夹将新建在该选中目录下。

注：容器化部署的单机项目，请勿在「全部文件」中新建文件夹，升级后会导致该文件夹丢失，文件无法找回。

- 3) 点击「新建文件夹」，设置文件夹名称，点击「确定」，即可新建文件夹。

注：文件夹名称仅支持输入英文字母和数字，不允许和当前目录其他文件夹重名。



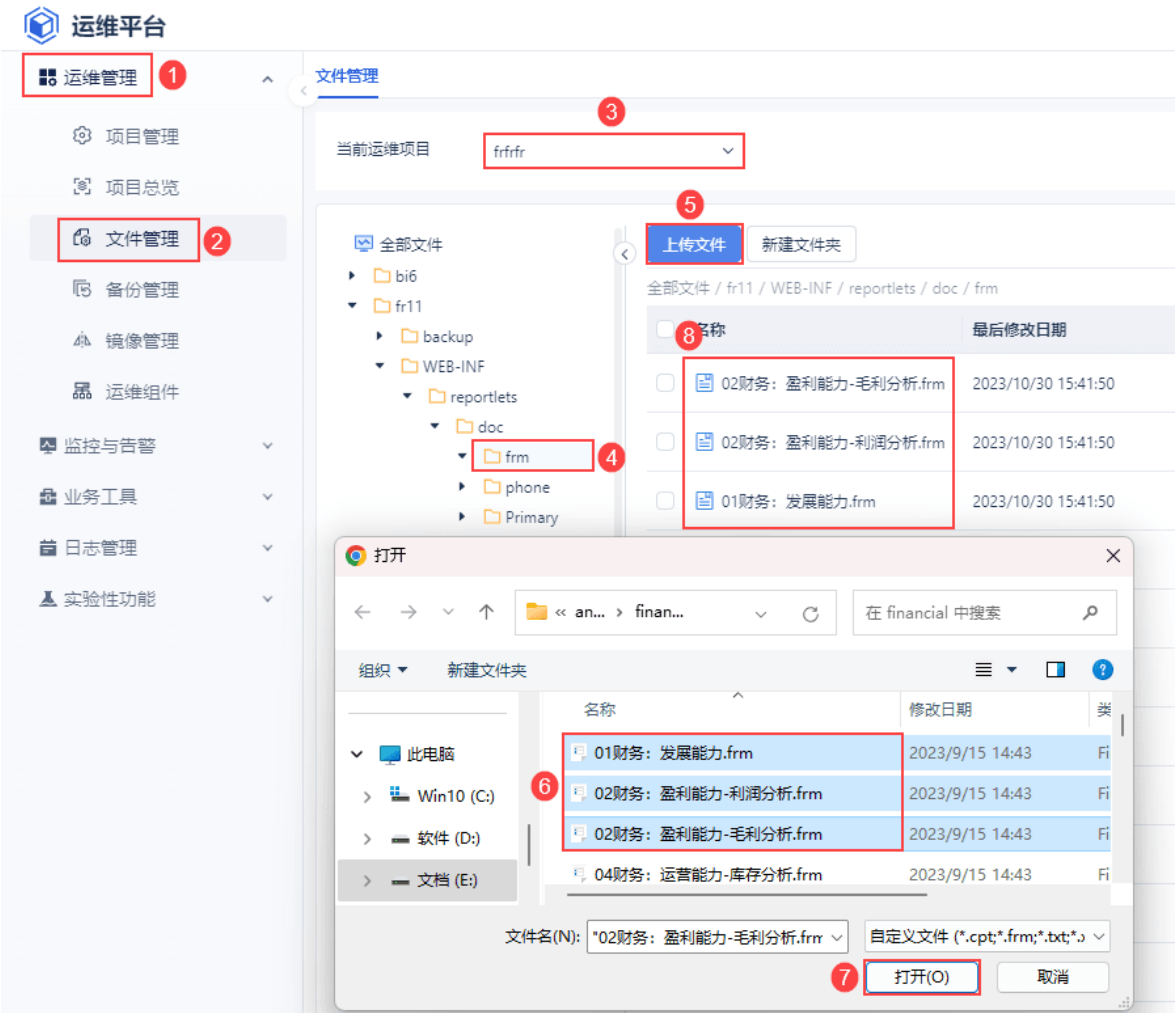
### 4. 上传文件

管理员可将本地文件上传到文件服务器的文件夹中。

- 1) 管理员登录运维平台，点击「运维管理>文件管理」。在「文件管理」页面，选择运维项目。

2) 选中指定的文件夹，点击「上传文件」，选择本地文件，即可将文件上传到该文件夹下。

- 支持上传的文件类型包括：模板文件（.cpt/.frm/.fvs）、文档文件（.txt/.xml/.xlsx/.xls）、资源文件（.mfvs）。
- 支持多选文件进行上传。
- 上传的文件大小不得超过 500M。
- 若存在同名文件，可选择覆盖/跳过/取消上传，若选择覆盖，被覆盖的文件将无法恢复。



## 5. 下载文件

管理员可将文件服务器中的文件下载到本地，支持单个下载/批量下载。

勾选指定的文件/文件夹，点击「下载」，即可将文件打包下载到本地。如下图所示：

运维平台

运维管理

项目管理

项目总览

文件管理

备份管理

镜像管理

运维组件

监控与告警

业务工具

日志管理

实验性功能

文件管理

当前运维项目friffr

全部文件

bi6

fr11

backup

WEB-INF

reportlets

doc

frm

phone

Primary

JS

Form

Parameter

SpecialSubject

上传文件

新建文件夹

下载

在当前文件夹中搜索

全部文件 / fr11 / WEB-INF / reportlets / doc / frm

名称	最后修改日期	大小	操作
<input checked="" type="checkbox"/> 02财务: 盈利能力-毛利分析.frm	2023/10/30 15:41:50	1137.4 KB	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> 02财务: 盈利能力-利润分析.frm	2023/10/30 15:41:50	718 KB	<input type="checkbox"/>
<input checked="" type="checkbox"/> 01财务: 发展能力.frm	2023/10/30 15:41:50	827.1 KB	<input type="checkbox"/>
<input type="checkbox"/> 决策报表自动查询.frm	2022/11/24 16:35:07	31.1 KB	<input type="checkbox"/>

## 5.4 备份管理

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.5.5	-
	提高备份前磁盘空间检查要求，防止备份导致磁盘爆
	满
V1.12.0	调整运维项目备份还原要求，存在 24 小时内备份即
	可还原

#### 1.2 应用场景

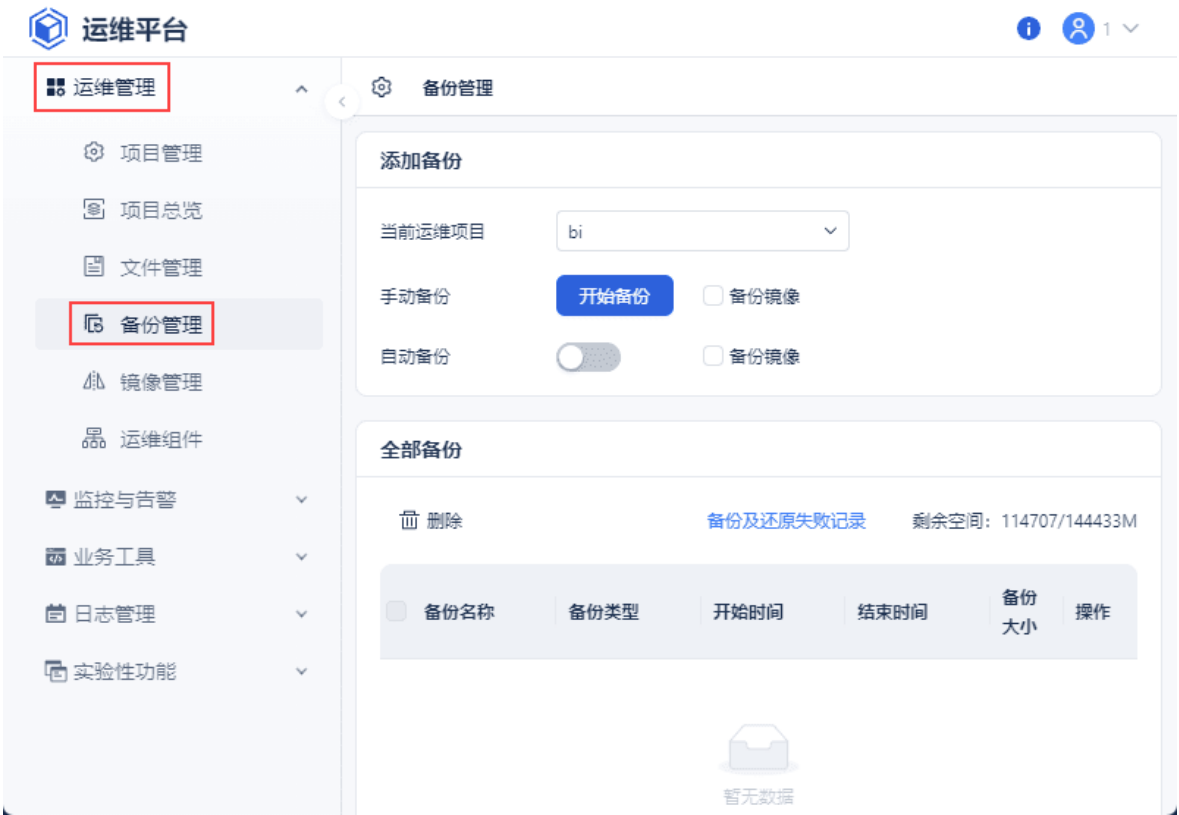
FineReport、FineBI 应用本身存在备份还原功能，但是存在一些短板：

- 1) 应用备份文件，只能备份至工程目录下，无法异地备份，误操作时若删除工程，可能会将备份也删除掉。
- 2) 应用中的模板、数据越多，应用备份的速度越慢。
- 3) 应用备份时，jar 包、模板、配置都是分开备份还原的，但是大多时候需要将这些内容按照时间点统一备份还原。

#### 1.3 功能简介

运维平台提供「备份管理」功能。

- 支持一键备份，支持自动备份，支持异机备份。
- 支持还原备份，支持管理备份版本，支持查看备份失败记录。



2. 添加备份

2.1 使用前提

若运维平台未配置镜像仓库，不支持使用备份功能。选中该项目后，提示「请配置镜像仓库后使用此功能」。



2.2 设置备份限制

由于备份会占用服务器内存，因此需要对备份进行限制。

管理员登录运维平台， 点击「运维管理>备份管理>全局设置」， 设置后点击「保存」生效。

设置内容	说明
备份上限	当达到备份上限时，会自动清理最早的一份手动备份文件/自动备份文件 默认支持分别保留 3 份手动备份文件/自动备份文件



当备份保存目录剩余可用空间 < （最近一次备份大小+设置值）时， 不再进行备份

磁盘限制 首次备份时， 备份保存目录剩余可用空间 < 2\*设置值， 则不进行备份

默认设置值为 5G



2.3 添加备份

支持手动备份和自动备份两种方式。

2.3.1 手动备份

管理员登录运维平台， 点击「运维工具>备份管理」， 选择运维项目。

1) 选择备份内容：

项目类型	备份内容
容器化部署的项目	1) 如不勾选「备份镜像」：  会备份项目当前的配置文件， 包括： config、lib、reportlets、plugins、assets、classes、customlib、dpworks、resources 文件
	2) 如勾选「备份镜像」：  会备份项目当前使用版本的镜像文件  会备份项目当前的配置文件， 包括： config、lib、reportlets、plugins、assets、classes、customlib、dpworks、resources 文件
非容器化部署的项目	1) 如不勾选「备份 jar 包」  会备份项目当前的配置文件， 包括： config、reportlets、plugins、embed 文件
	2) 如勾选「备份 jar 包」：  会备份项目当前使用版本的 JAR 包文件

会备份项目当前的配置文件，包括：config、reportlets、plugins、embed 文件

2) 开始备份：

点击「开始备份」，设置备份名称，点击「确定」，即可开始手动备份该工程。

开始备份后，用户可切换到其他页面进行操作，备份可在后台进行。

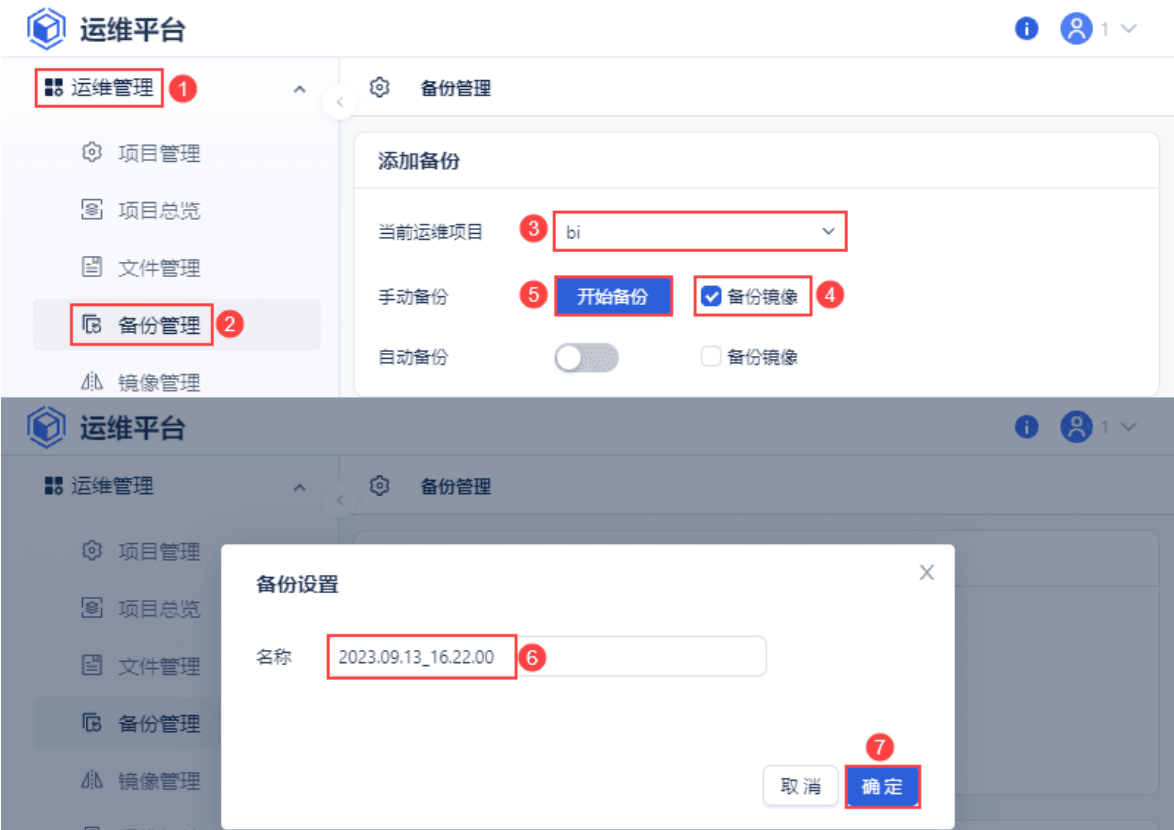
注 1：手动备份超出备份上限时，提醒「此项目已存在 X 个手动备份，此次备份将覆盖最早一次手动备份，是否继续？」。

注 2：若备份保存目录剩余可用空间小于 2.2 节的限制，则备份失败。

注 3：备份文件禁止命名重复。

3) 备份生成路径：

工程备份生成的内容，统一存放在运维平台的挂载目录中。用户可将运维平台和项目的服务器分开，实现异机备份。



2.3.2 自动备份

管理员登录运维平台，点击「运维工具>备份管理」，选择运维项目。

1) 选择备份内容：

容器化部署的项目，默认备份 config、lib、reportlets、plugins、assets、classes、customlib、dpworks、resources、schedule 文件。可自行选择是否「备份镜像」。

非容器化部署的项目，默认备份 config、reportlets 文件。可自行选择是否「备份 jar

包」。

2) 开始备份：

开启「自动备份」按钮，设置好备份频率。在指定时间，运维平台会自动对该项目进行备份。

自动生成的备份名称为备份时间。

注 1：自动备份超出备份上限时，自动删除最早 1 份。

注 2：若备份保存目录剩余可用空间小于 2.2 节的限制，则不再进行自动备份。

3) 备份生成路径：

工程备份生成的内容，统一存放在运维平台的挂载目录中。用户可将运维平台和项目的服务器分开，实现异机备份。



注：默认备份，一般存储在运维平台 ops/resource/backup 文件夹下，如下图所示：

```
[root@iZvts68dnh67t9Z backup]# pwd
/opspt/data/fanruan_2a5a2ff3/ops/resources/backup
[root@iZvts68dnh67t9Z backup]# ll
total 32
drwxr-x--- 2 root root 4096 Dec 10 01:00 2f9178f5-d368-481a-83e9-2864648b1634
drwxr-x--- 2 root root 4096 Dec 12 01:00 57f05e58-61aa-4f2b-bb9f-c06640132756
drwxr-x--- 2 root root 4096 Dec 5 19:32 6eadaa5e-4e4b-4f20-8445-38c33b4a936f
drwxr-x--- 2 root root 4096 Dec 9 01:00 8fe3c904-3f37-4b66-b4c5-9f484acda7ce
drwxr-x--- 2 root root 4096 Dec 11 01:00 ad5ef30c-aabc-4d85-a305-4fb1e2c16584
drwxr-x--- 2 root root 4096 Dec 7 01:00 c58eb613-b940-401f-8c1f-0cb36fef77a5
drwxr-x--- 2 root root 4096 Dec 6 01:00 dff6d490-cf4a-49d2-837b-78015cba7d8d
drwxr-x--- 2 root root 4096 Dec 8 01:00 e344398b-4d08-46d6-8504-313053c5c3c2
```

3. 还原备份

对于生成的备份，支持勾选还原。

3.1 使用前提

1) 容器化部署的项目

仅当运维项目为容器化部署时，支持还原功能。

非容器化部署的项目，不支持使用还原功能。选中该项目的备份后，还原按钮灰化，

提示「当前项目部署方式暂不支持还原备份」。

2) 存在 24 小时内的备份

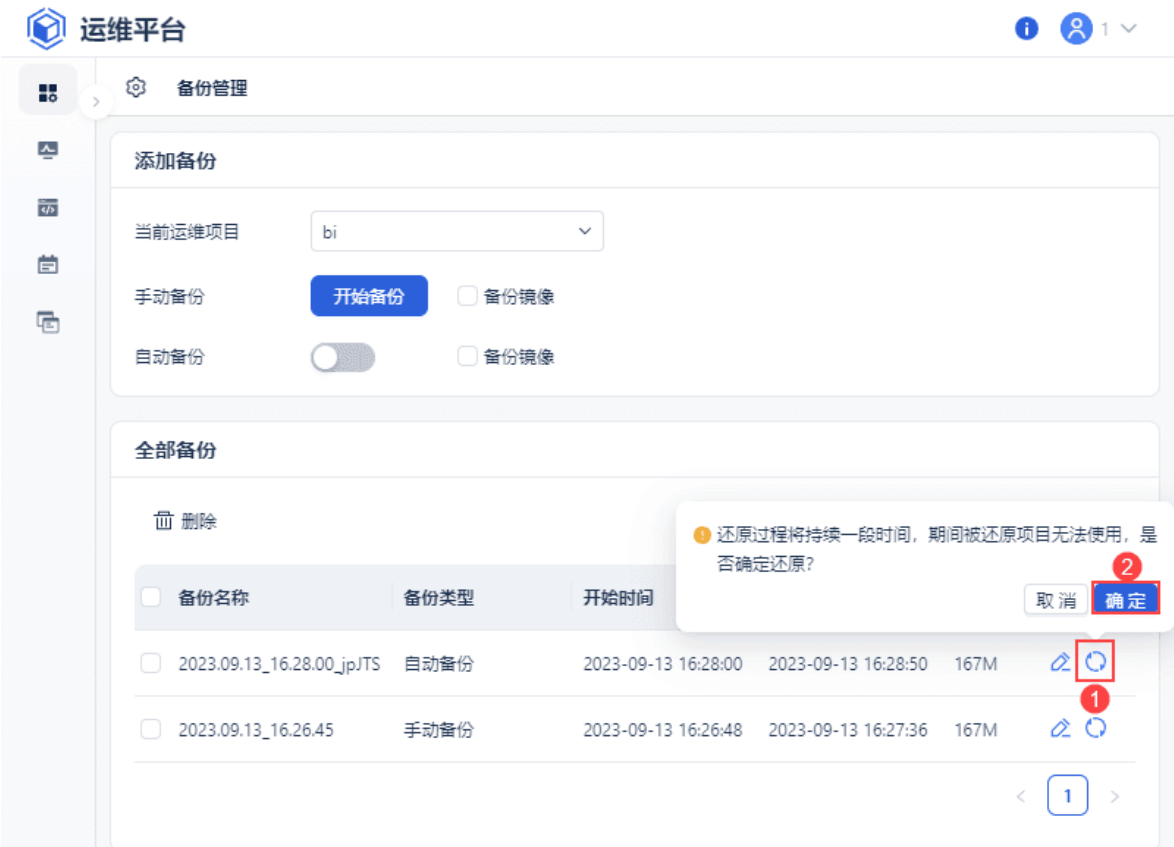
仅当存在 24 小时内的备份时，方可使用其他备份进行还原，避免还原出现问题。

3.2 还原备份

点击手动备份/自动备份的文件，点击「还原按钮」，提示「还原过程将持续一段时间，期间被还原项目无法使用，是否确定还原？」

点击「确认」按钮后开始还原，如下图所示：

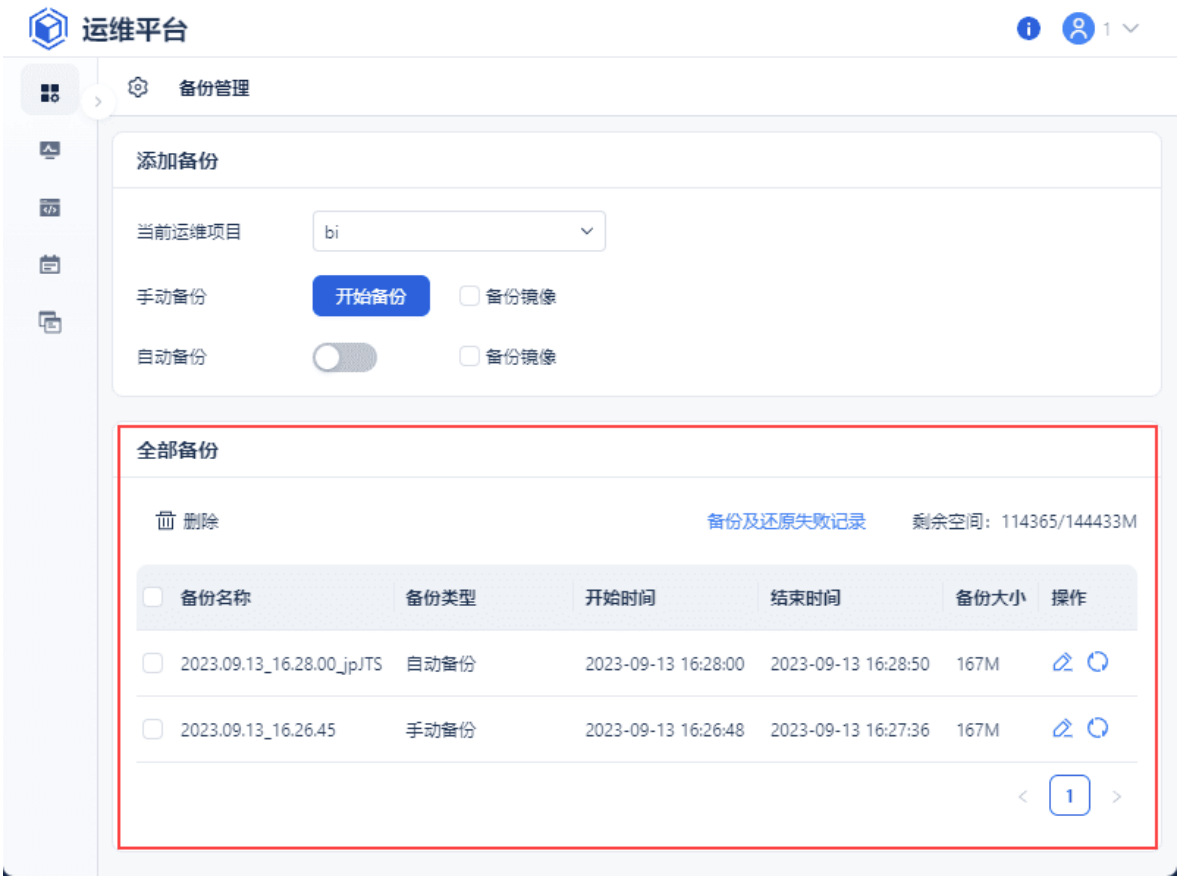
注：还原时，若检测发现不存在对应版本的镜像，则还原失败，提示「未找到所需要 xxx 镜像，请获取对应镜像并推送至镜像仓库后重试」。



4. 管理备份

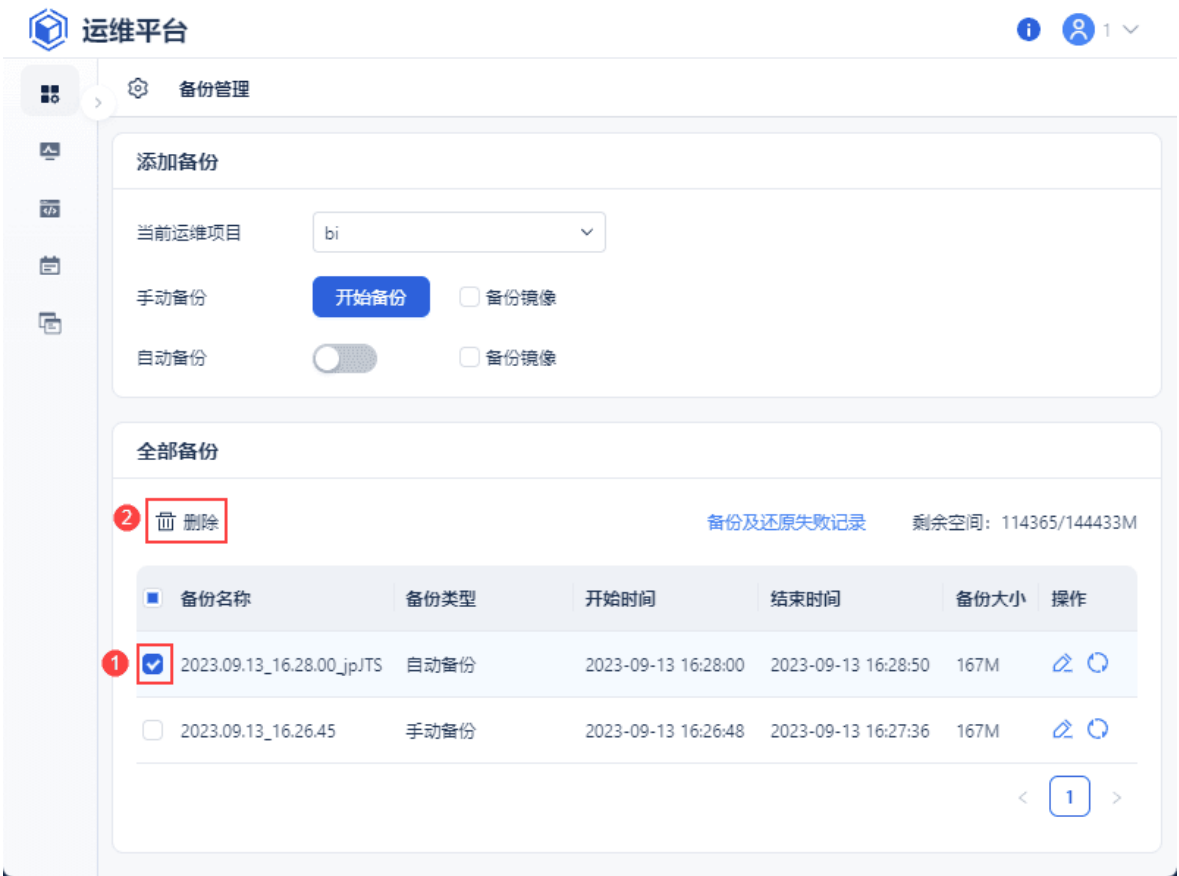
在「全部备份」中，以明细表格式展示每一个备份的详细情况。右上角展示备份保存目录剩余可用空间。

展示项	说明
备份名称	备份文件的名称
备份类型	备份的类型，包括：手动备份、自动备份
开始时间	备份开始的时间，精确到秒
结束时间	备份结束的时间，精确到秒
备份大小	备份文件的大小



4.1 删除备份

勾选需要删除的备份，点击「删除」，即可删除备份。如下图所示：



4.2 重命名备份

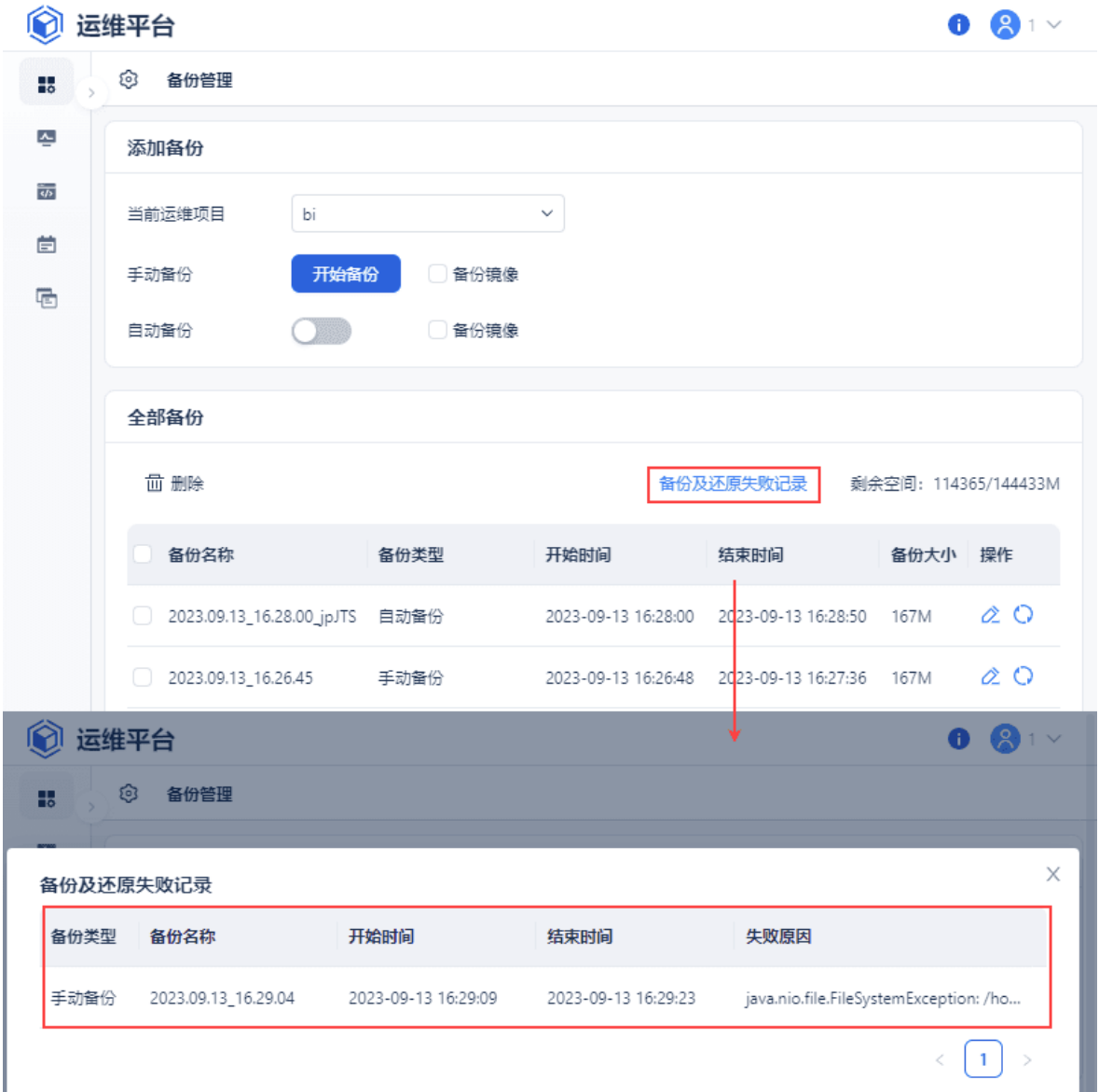
点击备份后方的「重命名」按钮，对备份进行重命名，点击「确定」，重命名生效。

注：备份文件禁止命名重复。



4.3 备份及还原失败记录

点击「备份及还原失败记录」，可查看备份及还原失败的记录明细表。如下图所示：



### 4.4 查看备份目录名

非容器化部署的工程，手动生成的备份，在运维平台的备份目录中显示的文件名均为ID。用户很难判断每个备份文件对应的前端显示内容，在进行还原时十分不方便。

对于非容器化部署的工程，手动生成的备份，鼠标悬浮即可查看备份目录名，点击按钮时可复制名称到剪贴板。



## 5.5 镜像管理

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.5.5	-

#### 1.2 功能简介

运维平台管理的项目中，存在多种组件。随着时间的推移，会不断推出更新版本的组件。

因此需要准备一个镜像仓库，用于获取和存放各类组件的最新镜像，方便对组件进行更新。

本文简单介绍外网环境/内网但可开通外网端口的环境，如何获取最新镜像到仓库。

### 2. 配置仓库地址

#### 2.1 默认仓库地址

在安装 FineOps 运维平台时，工具解压目录下 conf 文件夹里的 finekey.yaml 中，配置了 FineOps 使用的镜像仓库地址。

- 1) 默认会在 FineOps 服务器全新安装一个 registry 镜像仓库，仓库端口默认 5000。
- 2) 用户也可输入已有仓库的 url、username、password、ssl，则会启用用户自己的镜像仓库。

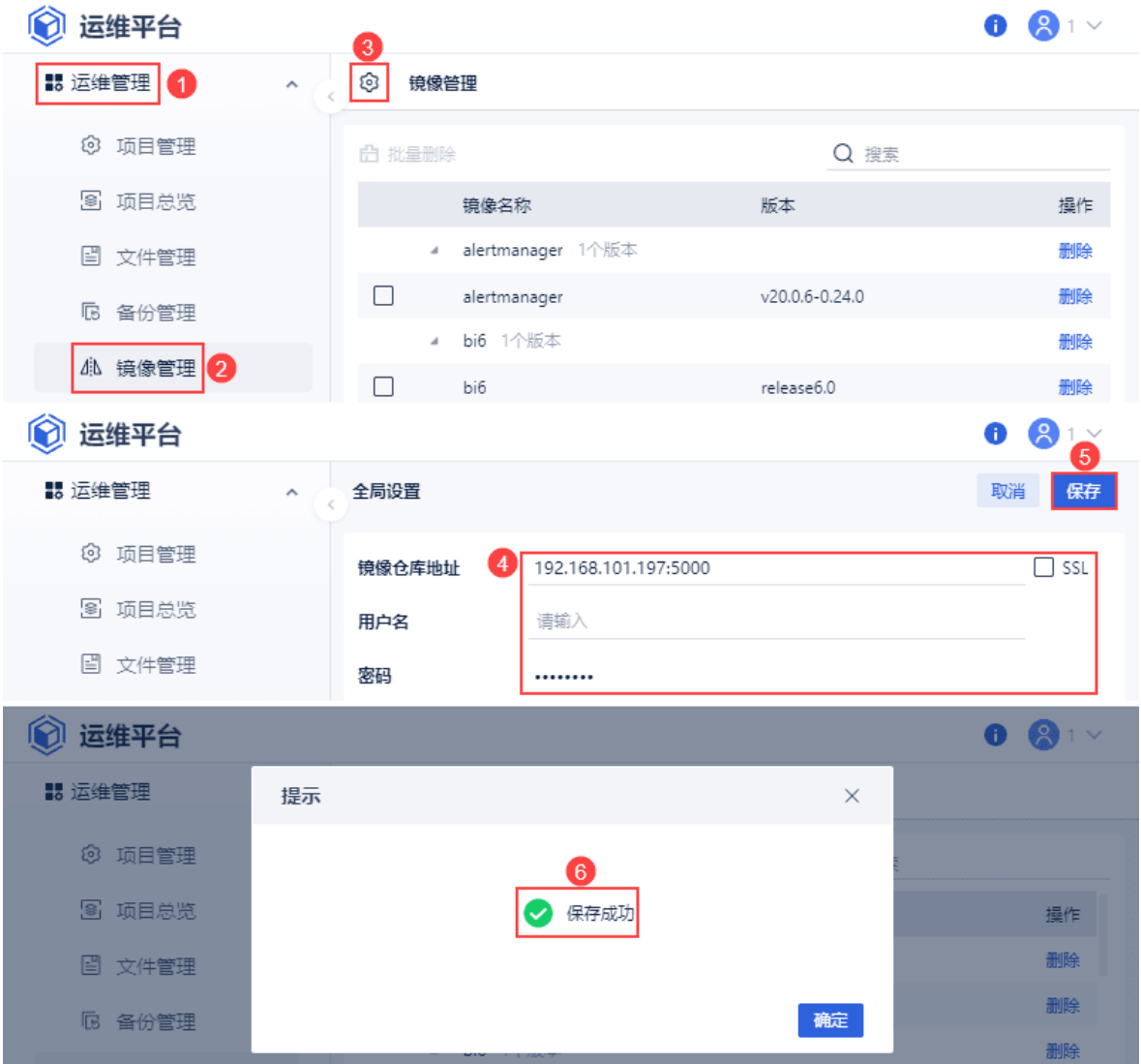
```
2 node:
3   ip: localhost      #机器ip
4   port: 22           #机器ssh端口
5   user:              #节点ssh用户
6   password:          #节点ssh密码
7   dataRootPath: ~/data
8 repo:
9   port: 5000         #待安装的registry仓库端口
10  #指定已有repo
11  url:               #已有repo的url
12  username:          #已有repo的用户名
13  password:          #已有repo的密码
14  ssl: true          #已有repo是否安全仓库
15
16 online: false       #在线安装
```

#### 2.2 修改仓库地址

FineOps 部署成功后，用户仍可前端修改仓库地址。



- 1) 管理员登录运维平台，点击「运维管理>镜像管理>全局配置」。
- 2) 配置仓库地址（IP:端口）、用户名、密码和是否开启 SSL，点击「保存」。
- 3) 当出现提示「保存成功」，运维平台的镜像仓库即修改成功。如下图所示：



3. 拉取镜像

3.1 确保外网联通

如需确保镜像仓库成功获取最新镜像，请确保镜像仓库所在服务器支持访问以下外网地址：

功能	网址
获取镜像/更新运维平台/ 更新项目组件	域名： registry.cn-hangzhou.aliyuncs.com  端口： 443
获取更新信息	四选一，有一个能联通即可：  https://cloud.fanruan.com/config https://cloud.fanruan.com:8443/config http://cloud.fanruan.com/config http://cloud.fanruan.com:8090/config

3.2 首次拉取镜像

帆软提供的 FineOps 运维平台安装包，包括三种类型。

1) 全量包离线安装包

在安装运维平台时，安装包中内置了全部镜像，均推送到镜像仓库。

包括：运维平台镜像、运维组件镜像、frbi 项目镜像

2) 在线版安装包和纯净版离线安装包

在安装运维平台时，安装包中内置了部分镜像，推送到镜像仓库。包括：运维平台镜像、运维组件镜像

在安装 frbi 项目时，会自动从云端拉取 frbi 项目镜像进行使用，并推送到镜像仓库中。

未使用到的 frbi 项目镜像，不会出现在镜像仓库中。



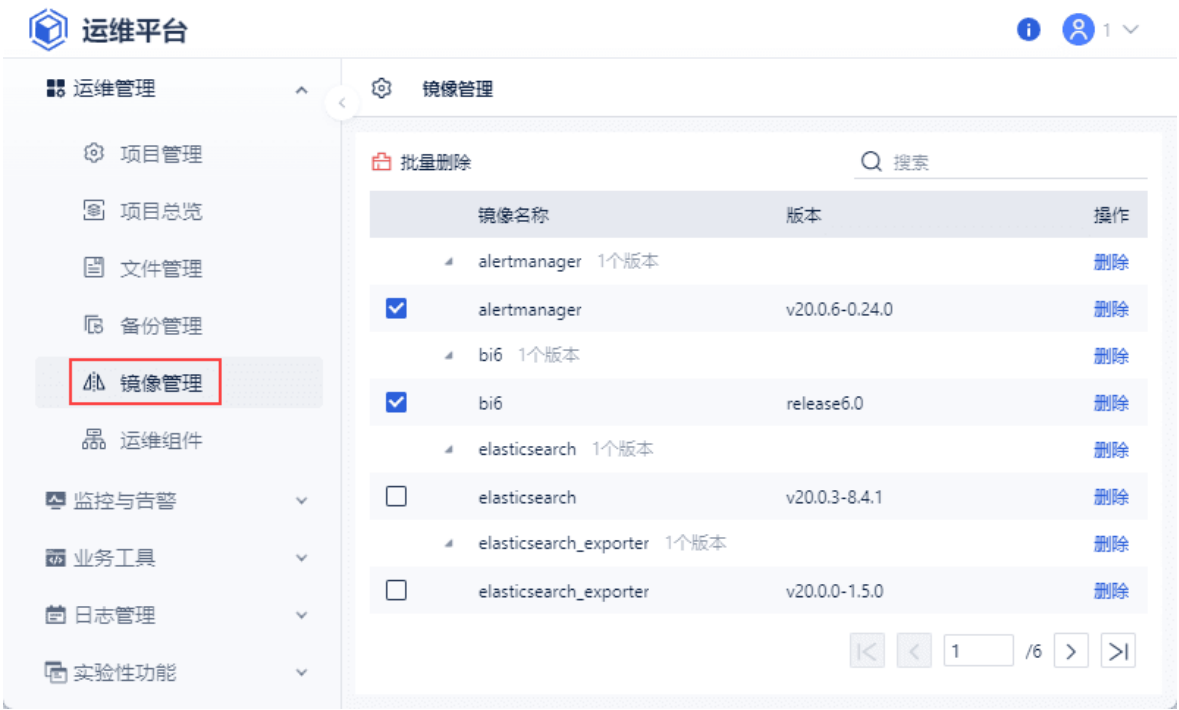
3.3 新镜像自动推送

当镜像被首次拉取到镜像仓库后，会出现在镜像管理界面中。

出现新版本组件镜像后，会自动推送到镜像仓库中，并展示在「镜像管理」页面中。

如下图所示：

支持删除/批量删除镜像版本。



#### 4. 手动推送镜像包

**应用场景：**运维平台中默认不内置 FineDataLink 相关镜像包，用户从技术支持处获取镜像包后，需要手动上传到镜像仓库中。

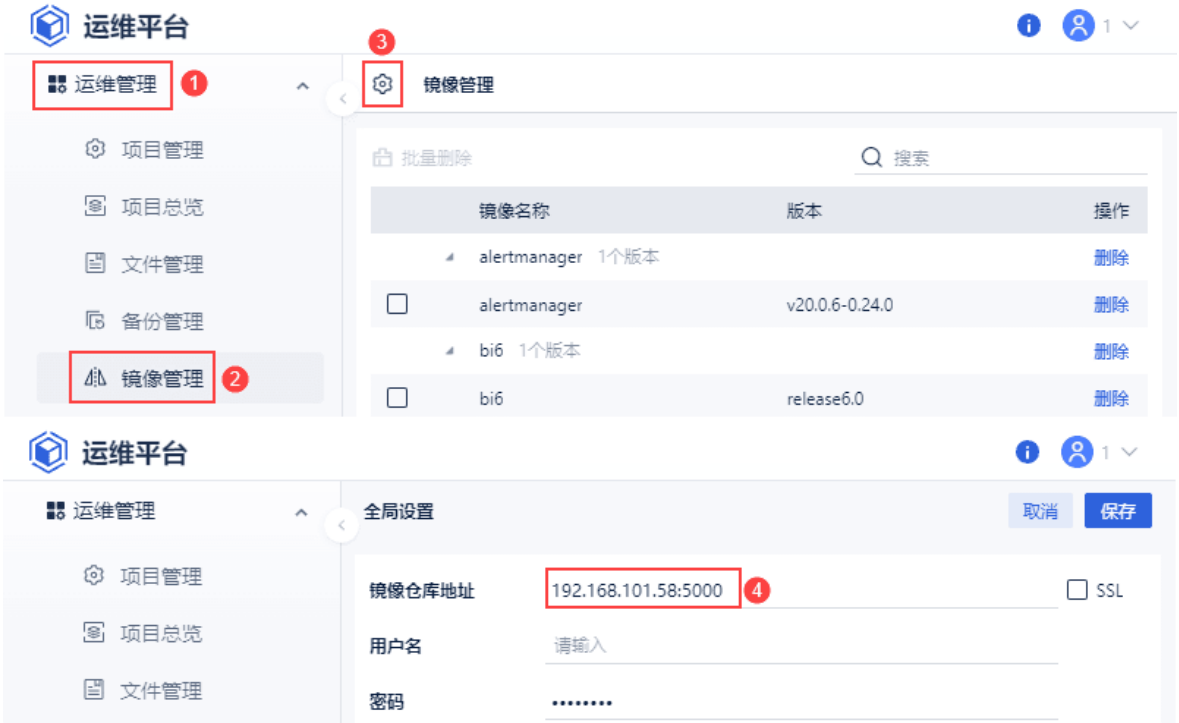
**示例：**本文演示如何上传 Kafka 镜像到镜像仓库中。

##### 4.1 确认镜像仓库地址

**目的：**确定你要上传的镜像的仓库地址。

**操作：**管理员可通过登录运维平台，点击「运维管理>镜像管理>全局配置」，查看运维平台所用镜像仓库的地址。

本文示例地址为：192.168.101.58:5000



## 4.2 确认仓库前缀

**目的：**对于私有仓库，需要确认仓库的前缀。这是在推送镜像时所需的完整镜像标签。

**操作：**确认仓库前缀的方法会根据你使用的具体的镜像仓库而有所不同。如果是通过 finekey 安装的 registry 仓库，前缀与仓库地址相同。

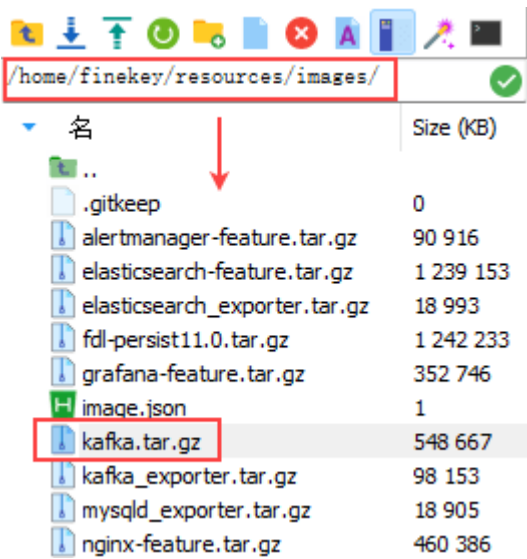
本文示例仓库前缀为：192.168.101.58:5000

## 4.3 获取镜像包

1) 付费用户请联系帆软技术支持，获取镜像包。技术支持联系方式：前往「[服务](#)」，选择「在线支持」、电话「400-811-8890」。

2) 将示例的 kafka.tar.gz 镜像包上传到仓库所在服务器。

本文上传的示例位置为：/home/finekey/resources/images



## 4.3 加载镜像

**目的：**需要使用 docker load 命令将镜像包加载到本地 Docker 引擎中。

**语句：**docker load -i /path/to/image.tar

**示例：**docker load -i /home/finekey/resources/images/kafka.tar.gz

```
[root@192 images]# docker load -i /home/finekey/resources/images/kafka.tar.gz
85b3e0196d07: Loading layer 561.8MB/561.8MB
Loaded image: kafka:v20.0.0-3.4.0
[root@192 images]#
```

## 4.4 标记镜像

**目的：**需要为镜像添加标签，以便将其与正确的仓库地址关联起来。

**语句：**docker tag <镜像 ID> <仓库前缀>/<镜像名称>:<标签>

**示例：**docker tag kafka:v20.0.0-3.4.0 192.168.101.58:5000/kafka:v20.0.0-3.4.0

```
[root@192 images]# docker load -i /home/finekey/resources/images/kafka.tar.gz
85b3e0196d07: Loading layer 561.8MB/561.8MB
Loaded image: kafka:v20.0.0-3.4.0
[root@192 images]#
[root@192 images]#
[root@192 images]# docker tag kafka:v20.0.0-3.4.0 192.168.101.58:5000/kafka:v20.0.0-3.4.0
[root@192 images]#
[root@192 images]#
```

4.5 推送镜像

目的：将镜像推送到仓库中。

语句：docker push <仓库前缀>/<镜像名称>:<标签>

示例：docker push 192.168.101.58:5000/kafka:v20.0.0-3.4.0

```
[root@192 images]#
[root@192 images]# docker push 192.168.101.58:5000/kafka:v20.0.0-3.4.0
The push refers to repository [192.168.101.58:5000/kafka]
85b3e0196d07: Pushed
v20.0.0-3.4.0: digest: sha256:daf82ed17f620f01ce1010efd31765a0604a20990ea1459b84a7e24ea5e6cc31 size: 530
[root@192 images]#
```

4.6 查看效果

管理员登录运维平台，在「运维管理>镜像管理」中，即可查看到刚刚手动推送的镜像包。

注：如参照上文推送镜像后，不能正常显示，付费用户请联系帆软技术支持协助更新version.json。

技术支持联系方式：前往「[服务](#)」，选择「在线支持」、电话「400-811-8890」。



## 5.6 运维组件

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.5.5	-
V1.5.7	elasticsearch、skywalking_oap 组件默认启动
V1.7.0	组件启停重启时，新增二次确认提示框，以防误操作

#### 1.2 应用场景

运维平台存在多个组件，需要给管理员提供一个汇总页面维护所有容器。

组件不可用时，需要给管理员提供快速恢复的方法和手段。

#### 1.3 功能简介

运维平台提供「运维组件」功能。管理员登录运维平台，点击「运维管理>运维组件」，支持对运维平台中的运维组件进行管理。



### 2. 展示信息

运维组件页面，展示运维平台中的所有运维组件信息。如下图所示：



支持展示的运维组件类型包括：

类型	组件	组件说明
运维平台相关	ops	运维平台容器，对 FineReport、FineBI 等项目进行运维
	ops-agent	与物理节点绑定，用于管理组件，容器化升级也是由该组件完成
	nginx	提供负载转发服务，所有服务的入口、使用都从这个入口访问
	grafana	用于展示监控仪表板
监控告警	prometheus	收集 exporter 的数据，提供给监控仪表板
	alertmanager	告警容器，用于告警规则的修改、触发等
链路追踪	elasticsearch	采集运维项目中 FineReport、FineBI、外接数据库、业务库和其他外置组件的请求
	skywalking_oap	用于链路追踪的数据上报和分析整理
指标收集	node_exporter	服务器物理节点的 exporter，收集运维平台所在服务器的各种指标信息
	nginx_exporter	nginx 服务的 exporter，收集 nginx 的指标信息
	elasticsearch_exporter	elasticSearch 的 exporter，收集 elasticSearch 的指标

镜像仓库

registry

存放 docker 镜像的仓库

支持展示的组件各类信息包括：

信息	说明
当前版本	当前组件安装的镜像版本
组件数量	该类型组件数量
	是否存在可更新版本
可更新版本	如已是最新版本，则显示：当前已为最新版本
	如存在更新版本组件，显示最新镜像版本号，界面右上角支持「一键更新」
组件配置	展示该类组件下所有容器在下文 3.3 节的配置（如未配置，显示无限制）
容器名	容器名称，方便识别用途
	容器是否启动
容器状态	running：启动
	exited：未启动
镜像信息	容器所使用的镜像信息
物理节点	容器所在服务器的 IP 地址
	容器的持续运行时间，ops 组件会额外显示健康状况
运行时间	healthy：容器健康运行
	unhealthy：容器无法正常运行，可能正在重启，请耐心等待变为健康
配置	展示下文 3.3 节的配置（如未配置，显示无限制）
操作	包括多种操作：更新、启动、停止、重启、更改配置、终端

### 3. 组件操作

#### 3.1 更新组件

1) 若存在「可更新版本」，可手动点击「一键更新」按钮，即可对该组件进行更新。

详情请参见：[内网升级运维平台](#)

2) 在线升级运维平台后，管理员登录运维平台时，运维平台会自动检测并升级运维组件。详情请参见：[外网升级运维平台](#)

3) 若当前已为最新版本，则「一键更新」按钮灰化，无法使用，提示「已为最新版本」。





### 3.2 组件启停

支持对组件进行启动、停止、重启等操作，快速启停组件。



注 1：部分组件间存在关联关系，重启某一个组件，会触发其他组件的重启。

注 2：组件关停，可能会影响运维功能的使用，具体如下表所示：

类型	组件	组件说明
运维平台相关	ops	必须启用该组件，如不启用该组件，运维平台直接宕机无法登录 因此运维组件界面不支持停止 ops 组件

	ops-agent	必须启用该组件，用于管理组件 因此运维组件界面不支持停止 ops-agent 组件
	nginx	必须启用该组件，所有服务的入口、使用都从这个入口访问，如不启用该组件，无法进入运维平台 因此运维组件界面不支持停止 nginx 组件
监 控 告 警	grafana	如不启用该组件，「监控与告警」功能中的仪表板无法查看
	prometheus	如不启用该组件，「监控与告警」功能中的仪表板无数据， 「监控与告警」功能中的告警规则无法使用
	alertmanager	如不启用该组件，「监控与告警」功能中的告警功能无法使用
链 路 追 踪	elasticsearch	如不启用该组件，「链路追踪」功能无法使用
	skywalking_oap	如不启用该组件，「链路追踪」功能无法使用
指 标 收 集	node_exporter	如不启用该组件，「监控与告警」功能中的仪表板无数据
	nginx_exporter	如不启用该组件，「监控与告警」功能中的仪表板无数据
	elasticsearch_exporter	如不启用该组件，「监控与告警」功能中的仪表板无数据
镜 像 仓 库		若用户自有仓库，无需启用该组件
	registry	若用户无镜像仓库，可启用该组件，作为「镜像管理」的仓库

### 3.3 更改配置

可输入大于 0 的正数。

内存达到上限时会自动重启组件，有业务连续性要求的组件请勿限制。

更改通用配置将自动重启组件，可能影响项目正常运行，请勿在使用期间操作。



注：下表列出的为组件正常运行的最低配置要求。建议用户进行配置时，使用最低配置的两倍，请根据服务器实际情况自行调整。

类型	组件	CPU 使用上限	内存使用上限	内存配置 xmx
运维平台 相关	ops	2Core	-	-
	ops-agent	1Core	-	1G
	nginx	1Core	-	-
监控告警	grafana	1Core	1G	-
	prometheus	1Core	1G	-
	alertmanager	1Core	1G	-
链路追踪	elasticsearch	1Core	-	2G
	skywalking_oap	1Core	-	2G
	node_exporter	0.1Core	0.1G	-
指标收集	nginx_exporter	0.1Core	0.1G	0.1G
	elasticsearch_exporter	0.1Core	0.1G	-
镜像仓库	registry	1Core	-	-

### 3.4 进入终端

尽管运维平台已经提供了很多界面化操作，但管理员仍有可能需要使用终端工具连接组件，对组件进行管理。

运维平台提供终端工具，支持快速进入组件终端。如下图所示：

左侧显示「命令行页面」，用户可输入命令进行查询/执行，支持一键清空命令行页面。

右侧显示该组件的具体容器信息。

运维平台

1

运维组件

已为最新版本

ops

启动 停止 重启 更改配置

当前版本

v1.10.0.11-feature

组件数量:

1

可更新版本:

当前已为最新版本

组件配置:

无限制

容器名	容器状态	镜像信息	物理节点	运行时间	配置	操作
ops	running	ops:v1.10.0.11-feature	192.168.101.197	Up 31 mi...	无限制	

容器终端

清空

连接中...

连接成功

# docker ps | grep ops

2cf0ed2da4b1 ops:v1.10.0.11-feature "/d

r-entrypoint..." 32 minutes ago Up 32 minutes (healthy)

.168.101.197:8081->8080/tcp

fanruan230911152433\_ops

9ba8def7190f ops\_agent:v1.10.0.1-feature "/d

r-entrypoint..." 44 hours ago Up 44 hours

.168.101.197:9070->9070/tcp

fanruan230911152433\_ops\_agent

#

容器信息

容器名

fanruan230911152433\_ops

容器状态

running

镜像信息

ops:v1.10.0.11-feature

物理节点

192.168.101.197

运行时间

Up 31 minutes (healthy)

配置信息

无限制

注 1：组件必须在启动 running 状态，方可进入组件终端。否则提示「当前容器状态下不支持进入终端」。

运维平台

1

运维组件

已为最新版本

prometheus

启动 停止 重启 更改配置

当前版本

v20.0.8-2.34.0

组件数量:

1

可更新版本:

当前已为最新版本

组件配置:

无限制

容器名	容器状态	镜像信息	物理节点	运行时间	配置	操作
prometheus	exited	prometheus:v20.0.8-2.34.0	192.168.101.197	Exited (0)...	无限制	

容器终端

清空

连接中...

连接成功

# docker ps | grep ops

2cf0ed2da4b1 ops:v1.10.0.11-feature "/d

r-entrypoint..." 32 minutes ago Up 32 minutes (healthy)

.168.101.197:8081->8080/tcp

fanruan230911152433\_ops

9ba8def7190f ops\_agent:v1.10.0.1-feature "/d

r-entrypoint..." 44 hours ago Up 44 hours

.168.101.197:9070->9070/tcp

fanruan230911152433\_ops\_agent

#

容器信息

容器名

fanruan230911152433\_ops

容器状态

running

镜像信息

ops:v1.10.0.11-feature

物理节点

192.168.101.197

运行时间

Up 31 minutes (healthy)

配置信息

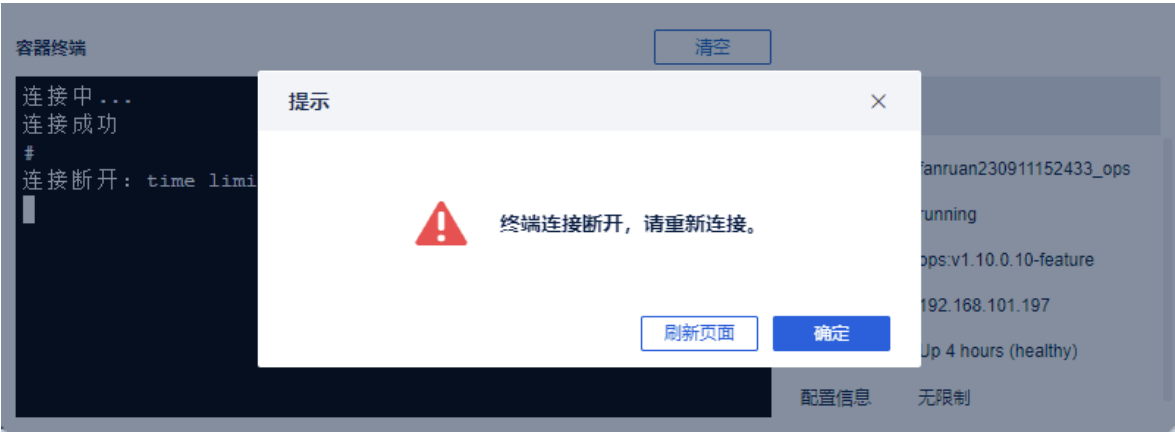
无限制

当前容器状态下不支持进入终端

注 2：单个运维平台，最多支持同时连接 3 个组件终端。超过 3 个时，打开提示「最多可连接 3 个终端，当前已达上限，请关闭其他终端后重试。」



注 3：若连接终端后，持续 10 分钟无命令操作，自动端口连接，提示「终端连接断开，请重新连接」。如下图所示：



# 5.7 为运维平台开启 SSL

## 1. 概述

### 1.1 版本

运维平台版本	功能变更
V1.6.0	-
V1.13.0	配置证书入口位置调整

### 1.2 应用场景

要启用 HTTPS 协议，我们需要使用有效的 SSL 证书。SSL 证书中包含身份验证信息，帮助保障用户的通信加密和安全。

在构建一个安全的工程中，证书起到关键作用，它保证了客户端与服务器之间的交互安全性。

本文将介绍如何通过简单的前端界面操作，为运维平台配置 SSL 证书，以满足使用 HTTPS 访问 FineOps 运维平台的需求。

这样做可以确保 FineOps 运维平台使用安全的通信协议，并给用户提供加密保护。

## 2. 使用前提

### 2.1 准备证书

在配置证书前，首先需要申请证书。

请自行向 CA 厂商购买证书，证书认证一般都是由 VeriSign、GlobalSign 等国际公认的 CA 机构认证。

需要准备的证书相关文件和要求如下：

内容	要求
证书文件	请准备好你的证书 证书格式为 .crt 或 .pem（nginx 仅支持这两种）
密钥文件	请准备好你的证书对应的密钥文件 证书格式为 .key

### 2.2 确认证书、域名、IP 一一对应

#### 1) 确认证书对应的域名

- 将证书上传到拥有 JDK 环境的计算机上，进入证书所在目录。

- 使用语句打印查询证书信息：keytool -printcert -file name.pem（请将 name.pem 换成你的证书名称）。
- 确认证书绑定的域名：在返回值中，「Owner: CN」/「所有者: CN」后的值，即为该证书对应的域名。

```
[root@ecs-new-0017 ~]# cd /home/ssl
[root@ecs-new-0017 ssl]# keytool -printcert -file demo.fanruan.com.pem
Certificate[1]:
Owner: CN= n.com
```

2) 确认域名绑定 IP

请将证书对应的域名，与 FineOps 运维平台所在服务器的 IP，互相绑定。

由于不同服务器的操作不尽相同，帆软不提供此操作的指导文档，请自行联系你的服务器厂商咨询。

绑定成功后，管理员可 ping 域名，确认返回值中的 IP 是否正确。

```
[root@ecs-new-0017 ssl]#
[root@ecs-new-0017 ssl]# ping opsdemo.fanruan.com
PING opsdemo.fanruan.com (106. .122) 56(84) bytes of data.
64 bytes from : icmp_seq=1 ttl=50 time=17.9 ms
64 bytes from : icmp_seq=2 ttl=50 time=17.8 ms
64 bytes from : icmp_seq=3 ttl=50 time=17.8 ms
64 bytes from : icmp_seq=4 ttl=50 time=17.8 ms
64 bytes from : icmp_seq=5 ttl=50 time=17.8 ms
64 bytes from : icmp_seq=6 ttl=50 time=17.7 ms
```

2.3 准备端口

请准备一个 FineOps 运维平台所在服务器的空闲端口，用于后续访问运维平台。

请确保端口防火墙已开放，可正常访问。

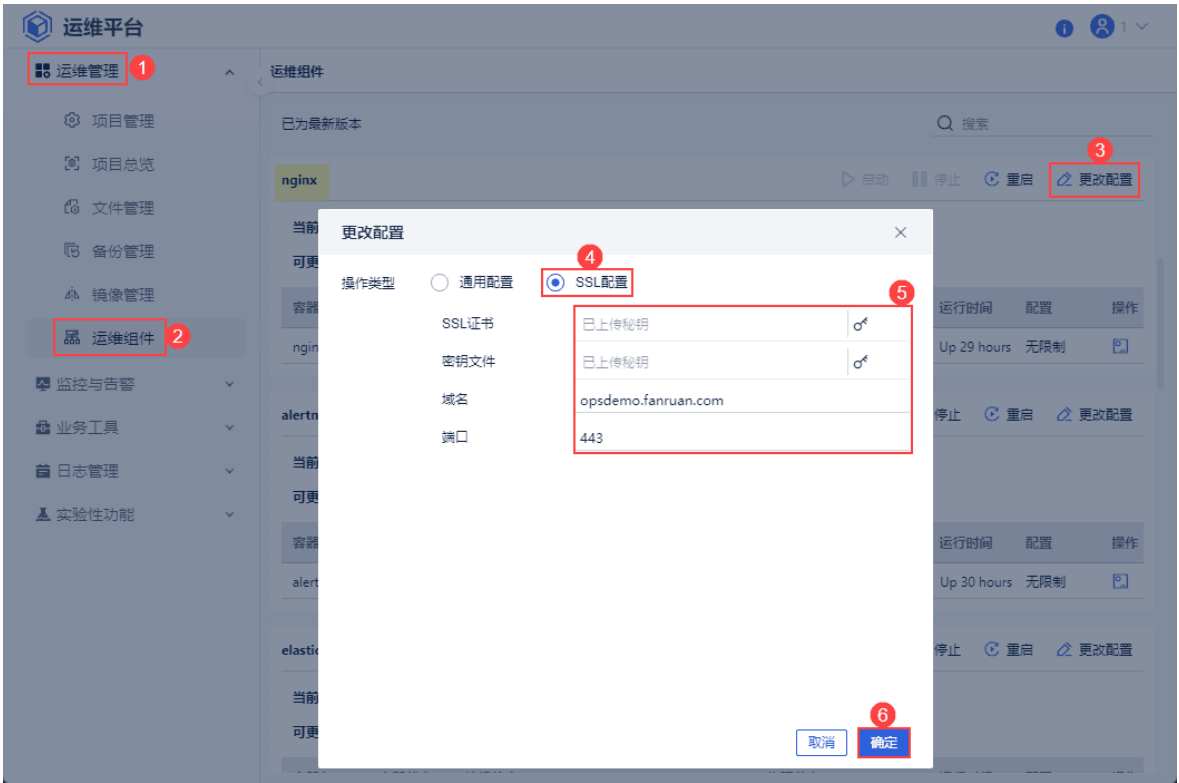
如希望在访问时不添加端口，直接通过域名访问，请使用 HTTPS 的默认端口 443 。

本文示例使用 443 端口。

3. 操作步骤

3.1 配置证书

- 1) 管理员登录 FineOps 运维平台，点击「运维管理>运维组件」。
- 2) 找到 Nginx 组件，点击「更改配置」。进入 SSL 配置界面。
- 3) 依次上传第二章准备的证书和密钥文件，填入域名和端口。
- 4) 点击「确定」。



### 3.2 校验配置

1) 点击保存后，会弹出提示「开启 SSL 将重启运维平台的 Nginx 组件，期间运维平台将持续几分钟无法访问，重启完成后 https 访问地址为 XXX（对应域名后的 url），是否继续？」

2) 点击「继续并重启」，运维平台会自动进行以下检测和配置，请耐心等待：

- 检测所填端口是否可用，如不可用，配置失败。
- 将证书和密钥文件上传到/nginx/conf/ssl 目录下
- 使用原命令新建 Nginx，并新增 443 端口，环境变量新增 SSL 相关配置。
- 自动重启 Nginx，启动完毕后即可实现 HTTPS 访问。



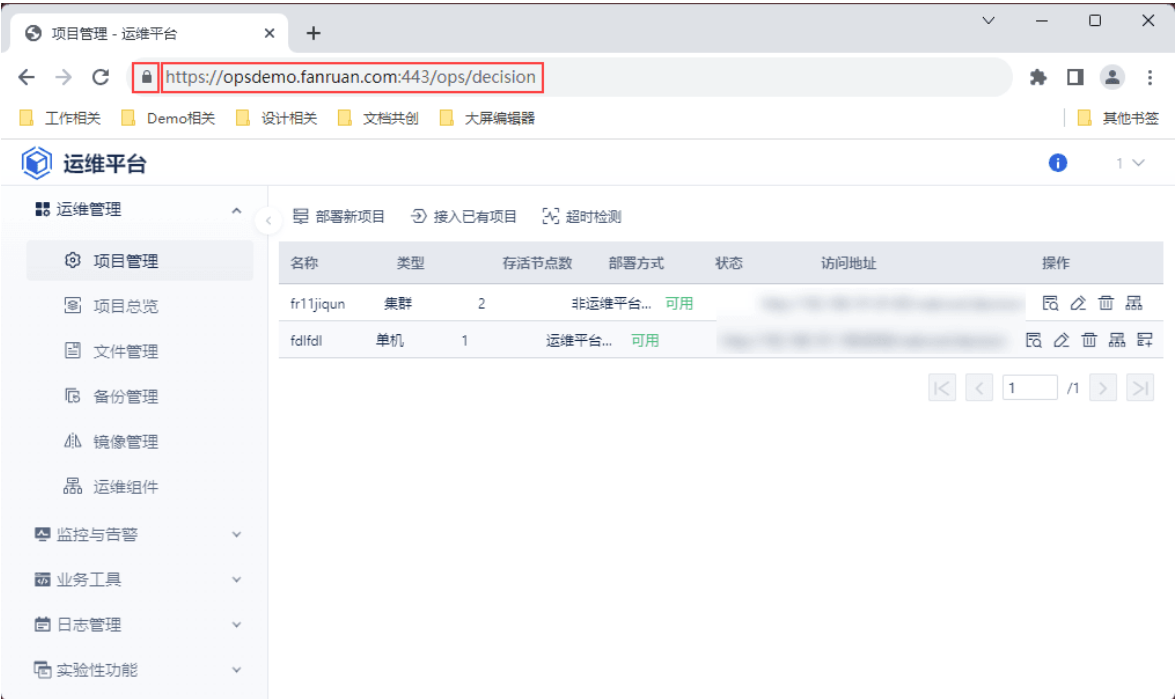
### 3.3 效果预览

在浏览器输入上文提示中的地址，即 https://your\_domain:port/ops/decision，确保能够成功访问，如果网页地址栏出现小锁标志，表示 SSL 配置成功。



- your\_domain 是你的域名。
- 端口为上文配置的端口，如果端口号为 443，可省略不写。

注：原 http 请求会自动跳转 https。



## 6. 监控与告警

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.5.5	-

#### 1.2 应用场景

##### 问题描述：

小明是公司的运维人员，负责管理着多个应用平台。随着应用长时间运行、人员增加，应用磁盘空间不足，长期高负载。

尽管他定期会对应用进行检查，仍然被业务人员反复质问：怎么又出现宕机事故了？怎么又没法访问平台了？

小明很困扰，怎么才能提前感知到这些平台的问题，并且解决这些隐患呢？

##### 解决方案：

小明将这些应用接入到运维平台，通过运维平台的「监控与预警」功能：

- 1) 前端可视化仪表盘，方便快速查看哪些配置飘红有隐患。
- 2) 设置告警规则和告警方式，运维平台检测到触发告警规则（例如某节点负载过高）时，直接发送邮件提醒小明。

小明终于可以在业务用户感知之前，解决这些应用平台的隐患。

### 2. 索引

功能	说明	帮助文档
系统资源监控	用于关注应用的全局状态情况概览，包括了各组件与应用	<a href="#">系统资源监控</a>
	的连通性，应用整体的运行状况，使用情况等	
系统健康观测仪表盘	对系统的健康状态和运行状态进行智能化监测	<a href="#">系统健康观测</a>
	服务器监控仪表盘主要用于监控帆软应用及相关组件运行的服务	
服务器监控	器的状态，方便及时关注服务器的各项指标情况，避免影响应用的	<a href="#">服务器监控</a>
	正常运行。	

运 维 平 台	运维平台监控仪表盘用于监控运维平台中各个运维组件的运行情况。	<a href="#">运维平台监控</a>
应 用 监 控	应用监控仪表盘用于监控帆软应用的运行情况，不同于主仪表盘，应用监控仪表盘可以针对单个节点的运行状态的详细数据进行查询。	<a href="#">应用监控</a>
组 件 监 控	提供多个仪表盘，包括：外接数据库监控、Redis 监控、Nginx 监控、Elasticsearch 监控、Minio 监控。	<a href="#">组件监控</a>
告 警 任 务	提供内置的默认告警规则。用户也可自定义异常触发条件和达成告警的持续时间	<a href="#">告警任务</a>
告 警 管 理	告警记录是指在告警管理中，针对运维项目出现异常情况时所触发的告警任务的记录	<a href="#">告警记录</a>
	通过查看告警记录，管理员可以获取关键指标和详细信息，如告警发生时间、告警类型、触发项目等	
告 警 通 知 设置	当触发告警时，支持通过三种方式通知管理员：邮箱、webhook 和企业微信。	<a href="#">告警通知设置</a>

## 6.1 监控

## 6.1.1 系统资源监控

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.5.5	-
	1) 功能更名
	「系统运行状态」更名为「系统资源监控」
	「当前系统状态」更名为「基础资源监控」
	2) 展示内容优化
	精选新增高频关注指标，剔除部分冗余指标
V1.5.9	仪表盘重新划分为：全项目状态总览、基础资源监控、线程池监控
	3) 联动交互优化
	顶部运维项目筛选按钮联动，基础资源监控和线程池监控中，默认仅展示所选项目的信息
	部分组件支持联动跳转「日志管理>性能监控>性能堆栈」，进行必要的堆栈分析

#### 1.2 功能简介

**应用场景：**

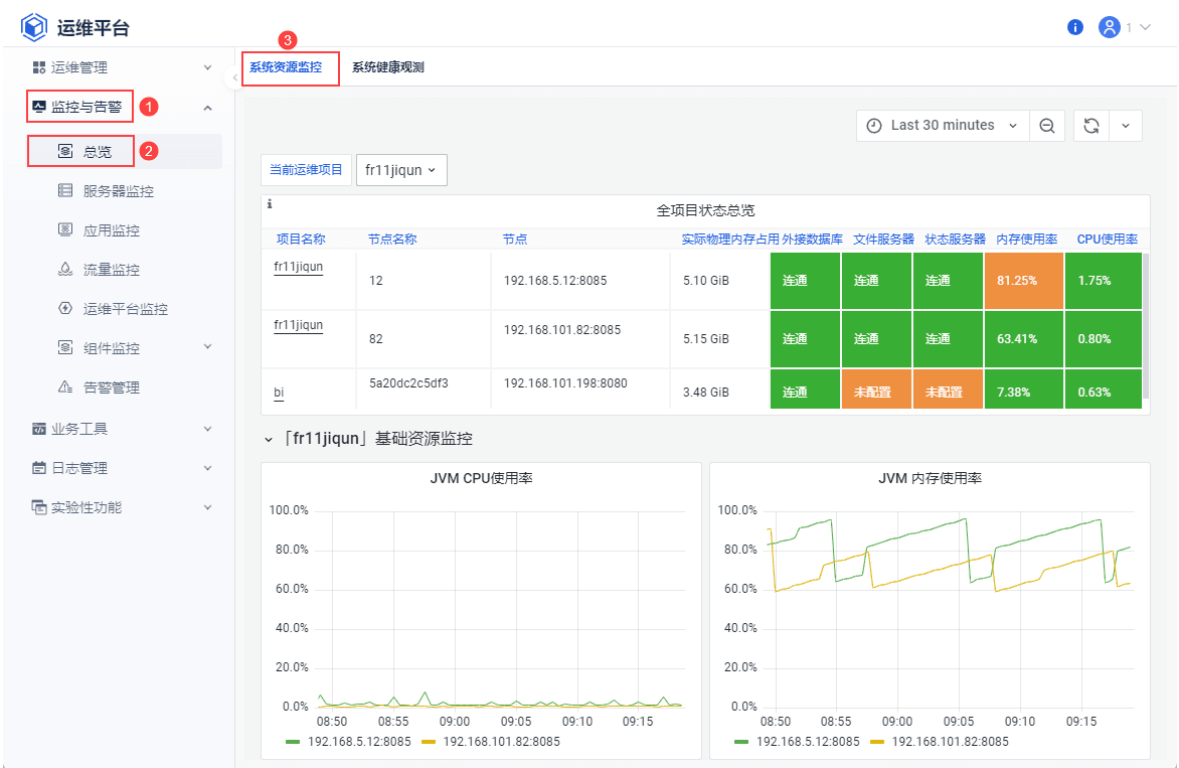
系统资源监控仪表盘，主要用于关注应用的全局状态情况概览，包括了各组件与应用  
的连通性，应用整体的运行状况，使用情况等。

**功能入口：**

管理员登录 FineOps 运维平台，点击「监控与告警>总览」，即可查看「系统资源监控」。  
仪表盘分为三个部分：全项目状态总览、基础资源监控、线程池监控。

**注意事项：**

右上角支持切换查看不同时间段的监控数据，默认时间段为前 30 分钟，支持查看时  
间范围为前 15 天。  
左上角支持切换查看不同运维项目的监控数据。



## 2. 使用前提

### 2.1 运维项目要求

仪表盘中的数据，由相关 exporter 指标收集组件提供。容器化部署的帆软项目，默认安装了相关组件。

非容器化部署的帆软项目，在接入运维平台时，需要安装相关组件，否则无法正常使用本文功能。

请确保参考「[接入已有非容器化项目](#)」文档，安装了相关日志信息采集和服务器信息采集组件。

### 2.2 运维平台要求

仪表盘功能，由这些运维组件提供：grafana、prometheus

因此如需正常查看相关仪表盘，需要确保这些组件正常运行。

管理员登录运维平台，点击「运维管理>运维组件」，可启动相关组件。



### 3. 全项目状态总览

#### 1) 仪表盘示例：

展示运维平台对接的所有运维项目的总览信息，如果出现异常指标将标红/标黄显示，通过点击项目名称可以联动下方明细指标。

全项目状态总览								
项目名称	节点名称	节点	实际物理内存占用	外接数据库	文件服务器	状态服务器	内存使用率	CPU使用率
fr11jiqun	12	192.168.5.12:8085	5.10 GiB	连通	连通	连通	83.74%	1.75%
fr11jiqun	82	192.168.101.82:8085	5.16 GiB	连通	连通	连通	66.26%	0.72%
bi	5a20dc2c5df3	192.168.101.198:8080	3.48 GiB	连通	未配置	未配置	12.76%	0.63%

#### 2) 仪表盘指标说明：

指标	具体说明
项目名称	当前查看的运维项目的名称 点击可联动下方基础资源监控、线程池监控的明细指标
节点名称	当前查看的应用的节点名称
节点	当前查看的应用节点的 IP 和端口号
实际物理内存占用	当前查看的应用，实际占用的物理内存大小 单位：GiB
外接数据库	应用与外接数据库的连接状态 结果可能为：连通、断开、未设置（未使用外接）和部分节点异常（集群）
文件服务器	应用与文件服务器的连接状态 结果可能为：连通、断开、未设置（未使用文件服务器）和部分节点异常（集群）
状态服务器	应用与状态服务器的连接状态

结果可能为：连通、断开、未设置（未使用状态服务器）和部分节点异常（集群）

内存使用率 = 当前使用的内存 / 可以使用的最大内存

当前使用的内存：当前使用的内存大小（字节）

内存使用率  
最大内存：可以使用的最大内存（字节），分配给 JVM 的最大内存

CPU 使用率  
CPU 使用率 = 采集时刻 CPU 使用率

4. 基础资源监控

1) 仪表盘示例：

与「应用状态总览」联动，展示选中的运维项目的基础资源信息。



2) 仪表盘指标说明：

指标	具体说明
JVMCPU 使用率	JVM 的 CPU 使用率变化情况，反映应用的 CPU 占用情况 点击指标点，支持联动跳转「日志管理>性能监控>性能堆栈」，进行必要的堆栈分析



JVM 内存使用率	<p>JVM 的内存使用率变化情况，反映应用的客观内存占用情况，可能包含可被 gc 回收的内存占用</p> <p>点击指标点，支持联动跳转「日志管理&gt;性能监控&gt;性能堆栈」，进行必要的堆栈分析</p>
GC 次数	<p>随时间的 gc 次数变化情况，gc 次数越多通常 gc 负荷越大</p> <p>点击指标点，支持联动跳转「日志管理&gt;性能监控&gt;性能堆栈」，进行必要的堆栈分析</p>
GC 时间	<p>应用每秒花在 gc 上的时间变化情况，反应 gc 的负荷和应用的吞吐量，gc 时间越长，应用吞吐量越小</p> <p>点击指标点，支持联动跳转「日志管理&gt;性能监控&gt;性能堆栈」，进行必要的堆栈分析</p>
应用进程实际占用物理内存	<p>随时间变化，应用实际占用的物理内存大小</p> <p>单位：GiB</p>
工程目录磁盘使用情况	<p>展示应用各个节点：</p> <p>已使用：工程目录已使用磁盘空间</p> <p>磁盘余量：工程目录剩余可用磁盘空间</p> <p>日均增量：过去 10 天内，工程目录磁盘日均增长占用空间</p>
tempDir 目录磁盘使用情况	<p>展示应用相关的 root 目录、工程目录、备份目录、temp 目录、schedule 目录和 logs 目录所在的磁盘空间使用率情况，当使用率超过 80%时会飘红显示，可用磁盘空间不足可能会影响应用的正常运行甚至宕机，需要尽快进行磁盘清理或扩容</p> <p>展示应用各个节点：</p> <p>已使用：工程 temp 目录已使用磁盘空间</p> <p>磁盘余量：工程 temp 目录剩余可用磁盘空间</p> <p>日均增量：过去 10 天内，工程 temp 目录磁盘日均增长占用空间</p>
logdb 目录磁盘使用情况	<p>展示应用各个节点：</p>

已使用：工程 logdb 日志目录已使用磁盘空间

磁盘余量：工程 logdb 日志目录剩余可用磁盘空间

日均增量：过去 10 天内，工程 logdb 日志目录磁盘日均增长占用空间

仅 FineBI 工程显示此内容，展示应用各个节点：

已使用：FineBI 工程抽取数据存放目录已使用磁盘空间

BI 抽取目录磁盘使用情况

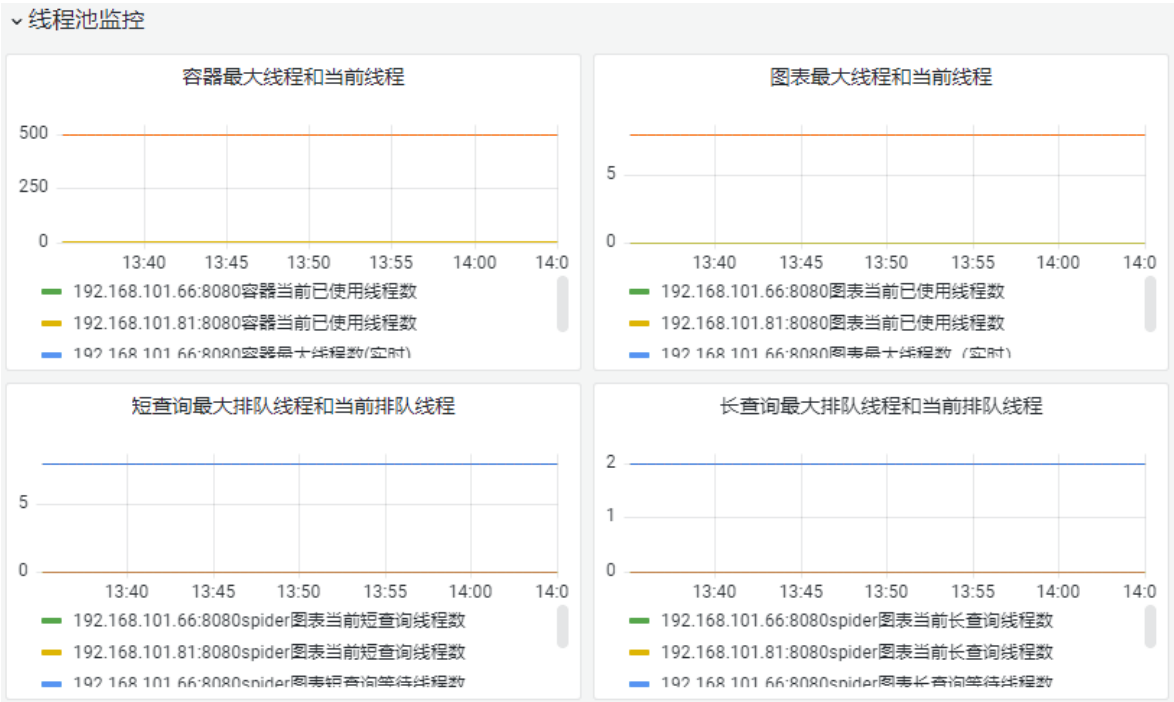
磁盘余量：FineBI 工程抽取数据存放目录剩余可用磁盘空间

日均增量：过去 10 天内，FineBI 工程抽取数据存放目录磁盘日均增长占用空间

5. 线程池监控

1) 仪表盘示例：

与「应用状态总览」联动，展示选中的运维项目的 4 种线程监控。



2) 仪表盘指标说明：

指标	具体说明
	展示应用各个节点：
容器最大线程和当前线程	容器当前已使用线程数
	容器最大线程数(实时)
	展示应用各个节点：
图表最大线程和当前线程	图表当前已使用线程数

图表最大线程数（实时）

展示应用各个节点：

短查询最大排队线程和当前排队线程

spider 图表当前短查询线程数

spider 图表短查询等待线程数

spider 图表短查询最大线程数

展示应用各个节点：

长查询最大排队线程和当前排队线程

spider 图表当前长查询线程数

spider 图表长查询等待线程数

spider 图表长查询最大线程数

## 6.1.2 系统健康观测

### 1. 概述

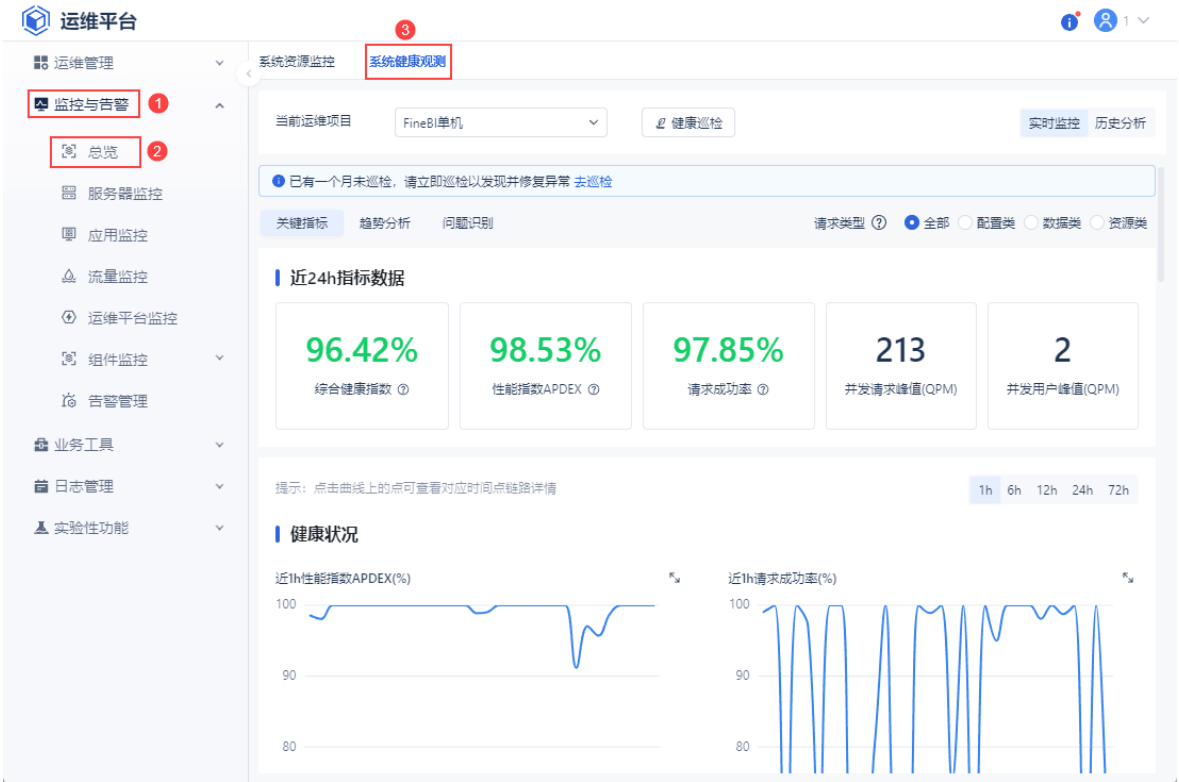
#### 1.1 版本

运维平台版本	功能变更
V1.5.5	-
	系统健康观测界面展示优化
	1) 请求并发量和在线用户数，优化为并发峰值和并发用户峰值
V1.5.6	2) 各指标图支持互相「定位当前时间」
	3) 各指标图支持拖动调整时间区间
	4) 各指标图支持放大查看
V1.5.8	新增「占用高内存的资源」清单
	1) 新增「健康巡检」强提醒按钮
V1.5.9	2) 各指标图和明细表标题联动显示时间点，快速定位时间
	1) 新增模块切换、请求类型筛选等全局控件
V1.10.0	2) 「节点情况明细表」更名为「节点概况表」
V1.12.0	为确保精确定位链路信息，系统健康观测联动跳转链路详情时，部分跳转项从「首屏链路」调整为「调用链路」

#### 1.2 功能简介

运维平台提供系统健康观测仪表盘，对系统的健康状态和运行状态进行智能化监测。

系统健康观测仪表盘，主要站在用户体验的角度，观测系统的稳定情况、性能情况、运维人效等。并通过观测请求，输出问题请求所在的主体（仪表板、模板等）。



## 2. 使用前提

### 2.1 运维项目要求

如需完整使用「系统健康观测」功能：

- 请确保接入的 FineReport 版本为 **11.0.12** 及以上、FineBI 版本为 **6.0.9** 及以上。
- 请确保 FR/BI 项目安装了 **1.4.9** 及以上版本的系统运维插件。

### 2.2 运维平台要求

系统健康观测与链路追踪功能互通，由多个运维组件提供功能：elasticSearch、skywalking\_oap。

如需完整使用「系统健康观测」功能，必须先启动这些组件，否则系统健康观测界面中数据为空。

管理员登录运维平台，点击「运维管理>运维组件」，先启动 elasticSearch 组件，再启动 skywalking\_oap 组件。



2.3 磁盘可用空间

请确保 FineOps 运维平台所在分区，已使用磁盘空间不到 90%，否则系统健康观测界面中数据为空。

```
[root@iZ28x3egfh1Z ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/vda1       79G   43G   33G   58% /
devtmpfs        32G    0    32G    0% /dev
tmpfs           32G    0    32G    0% /dev/shm
tmpfs           32G  2.4M    32G    1% /run
tmpfs           32G    0    32G    0% /sys/fs/cgroup
/dev/vdb1       246G   89G  146G   38% /data_bak
/dev/vdc1       246G  224G   11G   96% /data
tmpfs           6.3G    0    6.3G    0% /run/user/0
```

3. 使用方法

- 1) 管理员登录 FineOps 运维平台，点击「监控与告警>总览>系统健康观测」。
- 2) 管理员可点击切换选择运维项目。
- 3) 管理员可点击跳转查看不同观测内容。

观测内容	说明
关键指标	展示可衡量整个系统健康使用情况的 KPI 指标 即下文第四章内容，包括：指标数据
趋势分析	展示不同时段下各监控观测指标的差异情况，支持联动跳转分析等 即下文第五章内容，包括：健康状况、用户使用统计、节点概况表、性能状况
问题识别	展示被运维平台识别出的项目的问题列表 即下文第六章内容，包括：问题请求列表、占用高内存的资源

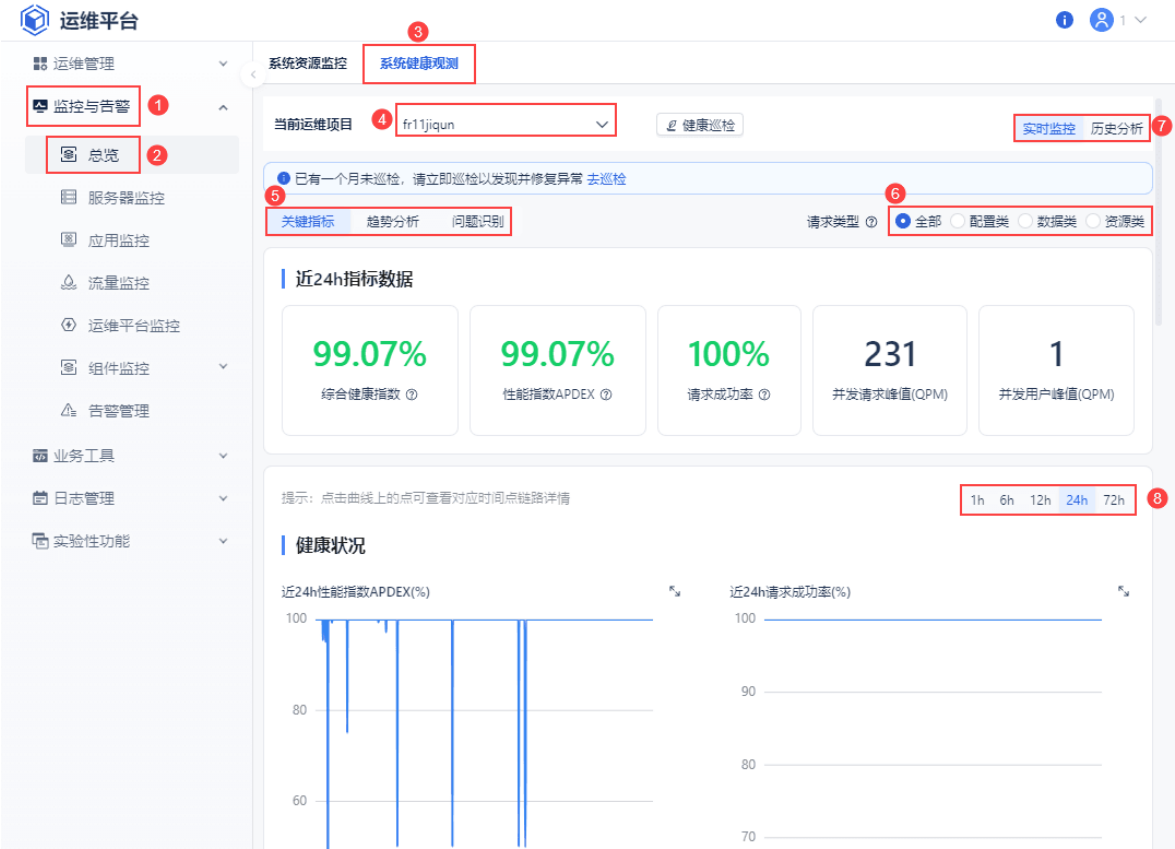
- 4) 用户可点击筛选不同请求类型。

请求类型影响指标数据、健康状况、用户使用统计、节点概况表。

请求类型	说明
全部	包括下文所有请求类型
配置类	打开目录、搜索、计算权限等平台操作行为的请求
数据类	访问报表、数据表，查看数据结果的相关请求
资源类	前端 JS、CSS、字体、图标等静态资源请求

5) 用户可点击切换指定监控时间范围。

时间范围	说明
实时监控	1) 当观测内容为「关键指标」和「占用高内存的资源」时：  时间范围为过去 24 小时内  2) 当观测内容为「趋势分析」和「问题请求列表」时：  时间范围可选过去 1h、6h、12h、24h、72h
历史分析	时间范围可选过去任一月份



4. 关键指标

界面说明：

展示五个重点指标。指标值如有异常，会显示为红色。

指标说明：

指标	具体说明
综合健康指数	1) 指标说明：用于表示系统健康程度的综合指标  2) 计算逻辑：

	<p>综合健康得分=(满意请求数+可容忍请求数/2)/总请求数)*100%</p> <p>满意请求：耗时&lt;3s 且成功的请求</p> <p>可容忍请求：3s&lt;=耗时&lt;12s、且成功的请求</p> <p><b>3) 计算范围：</b></p> <p>实时监控：过去 24 小时内</p> <p>历史分析：所选当月范围内</p> <p><b>4) 建议值：</b>建议保持在 95%以上</p> <p><b>1) 指标说明：</b>用于评估应用性能的行业标准</p> <p><b>2) 计算逻辑：</b></p> <p>APDEX=(满意请求数+可容忍请求数/2)/总请求数 )*100%</p> <p>满意请求：耗时&lt;3s 的请求</p>
性能指数 APDEX	<p>可容忍请求：3s&lt;=耗时&lt;12s 的请求</p> <p><b>3) 计算范围：</b></p> <p>实时监控：过去 24 小时内</p> <p>历史分析：所选当月范围内</p> <p><b>4) 建议值：</b>建议保持在 95%以上</p> <p><b>1) 指标说明：</b>用于评估应用性能的行业标准</p> <p><b>2) 计算逻辑：</b>请求成功率=(成功请求 / 所有请求数) × 100%</p>
请求成功率	<p><b>3) 计算范围：</b></p> <p>实时监控：过去 24 小时内</p> <p>历史分析：所选当月范围内</p> <p><b>4) 建议值：</b>建议保持在 95%以上</p> <p><b>指标说明：</b></p> <p>实时监控：使用请求切片去计算每分钟的并发数，展示过去 24 小时内</p>
并发峰值	<p>并发量最大值</p> <p>历史分析：使用请求切片去计算每分钟的并发数，展示所选月份的并发量最大值</p>



指标说明：

实时监控：使用请求切片去计算每分钟的并发用户数，展示过去 24 小时并发用户峰值 小时内并发用户数最大值

历史分析：使用请求切片去计算每分钟的并发用户数，展示所选月份的并发用户数最大值



5. 趋势分析

5.1 健康状况

界面说明：

展示指定时间范围内的性能指数、请求成功率。

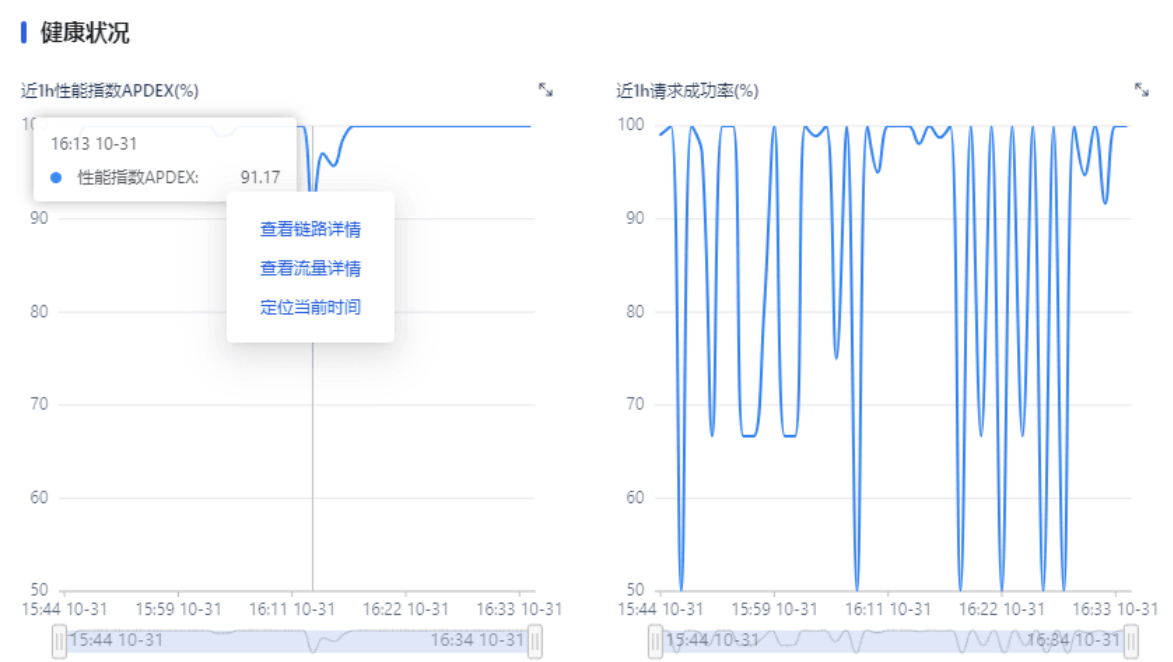
- 实时监控下，点击异常点，可联动跳转至「链路追踪>[调用链路](#)」页面，默认带入查询时间前后一分钟进行查询。
- 实时监控下，点击异常点，可联动跳转至「监控与告警>[流量监控](#)」页面，默认带入查询时间前后一分钟进行查询。
- 实时监控下，点击异常点，可同时定位所选时间点的各项指标值。

指标说明：

指标	具体说明
性能指数 APDEX	1) 指标说明：用于评估应用性能的行业标准
	2) 计算逻辑：
	APDEX=(满意请求数+可容忍请求数/2)/总请求数 )*100%
	满意请求：耗时<3s 的请求
	可容忍请求：3s<=耗时<12s 的请求
请求成功率	3) 建议值：建议保持在 95%以上
	1) 指标说明：用于评估应用性能的行业标准

2) 计算逻辑：请求成功率=(成功请求 / 所有请求数) × 100%

3) 建议值：建议保持在 95%以上



实时监控：使用请求切片去计算每分钟的并发用户数，  
展示过去 24 小时内并发用户数最大值  
历史分析：使用请求切片去计算每分钟的并发用户数，  
展示所选月份的并发用户数最大值



5.3 节点概况表

界面说明：展示指定时间范围内，每个节点的系统健康观测指标。

指标说明：指标计算逻辑可参考上文。

近1h节点概况表

节点	性能指数 APDEX(%)	请求成功率 (%)	服务响应时间 (ms)	数据响应时间 (ms)	并发请求峰值 (QPM)	并发用户峰值 (QPM)
192.168.101.82:8085	99.97	100	115	0	41	1
192.168.5.12:8085	99.74	100	266	0	28	1

< 1 >

5.4 性能状况

界面说明：

展示指定时间范围内的看板情况。

- p90 曲线表示有 90%的请求耗时都小于该曲线对应值。p95、p99 曲线以此类推。

- 实时监控下， 点击服务响应时间、数据响应时间异常点， 可联动跳转至「链路追踪>[首屏链路](#)」页面， 默认带入查询时间前后一分钟进行查询。
- 实时监控下， 点击用户白屏时间、用户首屏时间异常点， 可联动跳转至「链路追踪>[调用链路](#)」页面， 默认带入查询时间前后一分钟进行查询。
- 实时监控下， 点击异常点， 可同时定位所选时间点的各项指标值。

指标说明：

指标	具体说明
服务响应时间	有效请求的平均耗时。服务器端完成请求响应的时长， 用来衡量帆软整体服务的影响情况
数据响应时间	有效请求的平均耗时。数据引擎/数据库完成请求响应的时长， 用来衡量数据计算的性能情况
用户白屏时间	用户打开一张模板， 模板界面完全白屏的时间
用户首屏时间	用户打开一张模板， 模板完全加载完成的时间



6. 问题识别

6.1 问题请求列表

界面说明：

- 展示指定时间范围内所有报错请求和耗时超过 10S 的慢请求。
- 根据模板资源，统计每个资源的问题类型、查询次数、影响用户数
- 实时监控下，支持「查看链路详情」操作，可联动跳转至「链路追踪>[调用链路](#)」页面，默认带入 sessionID 查询，查询开始时间控件带入 3 天范围。

近72h问题请求列表 报错请求及耗时超过10s的慢请求

资源类型	资源名称	问题类型	查询次数	影响用户数	操作
entry	管理与驾驶舱/操作型驾驶舱/实时服务数据监控	TIMEOUT	1	1	<a href="#">查看链路详情</a>
entry	公共数据目录_内置	TIMEOUT	1	1	<a href="#">查看链路详情</a>
entry	参数查询/条件查询与图表联动	TIMEOUT	1	1	<a href="#">查看链路详情</a>
entry	参数查询/自定义时间段查询	TIMEOUT	1	1	<a href="#">查看链路详情</a>

< 1 >

6.2 占用高内存的资源

界面说明：

- 展示过去 24 小时/所选月份内识别到的占用高内存的资源。
- 根据资源类型，统计每个资源的类型、资源名称、资源创建者、占用内存、访问用户、识别时间。

8月占用高内存的资源

资源类型	资源名称	资源创建者	占用内存（GB）	访问用户	识别时间
widget	Ivy.Weng的分析/honeypot_近6月明细/组件		4.55	Ivy.Weng	2023-08-30 15:21
widget	Ivy.Weng的分析/honeypot_近6月明细/组件		8.21	Ivy.Weng	2023-08-29 11:06
widget	Ivy.Weng的分析/honeypot_近6月明细/组件		3.87	Ivy.Weng	2023-08-29 11:05
widget	Ivy.Weng的分析/honeypot_近6月明细/组件		5.41	Ivy.Weng	2023-08-29 11:05
dashboard	Ryan.Song的分析/京津区域/分析主题/仪表盘1		4.66	Ryan.Song	2023-08-28 19:57

< 1 2 3 4 5 >

7. 下一步操作

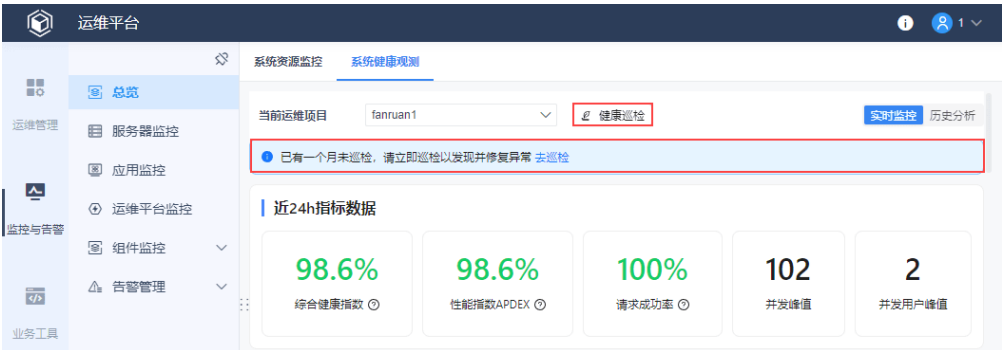
下一步操作

说明

管理员应当对应用进行定期健康巡检，确保应用所在的环境及应用内配置合理，以保证应用的正常运行。

当出现「已有一个月未巡检，请立即巡检以发现并修复异常」提示时，建议即刻进行巡检操作，并配置定期自动巡检。

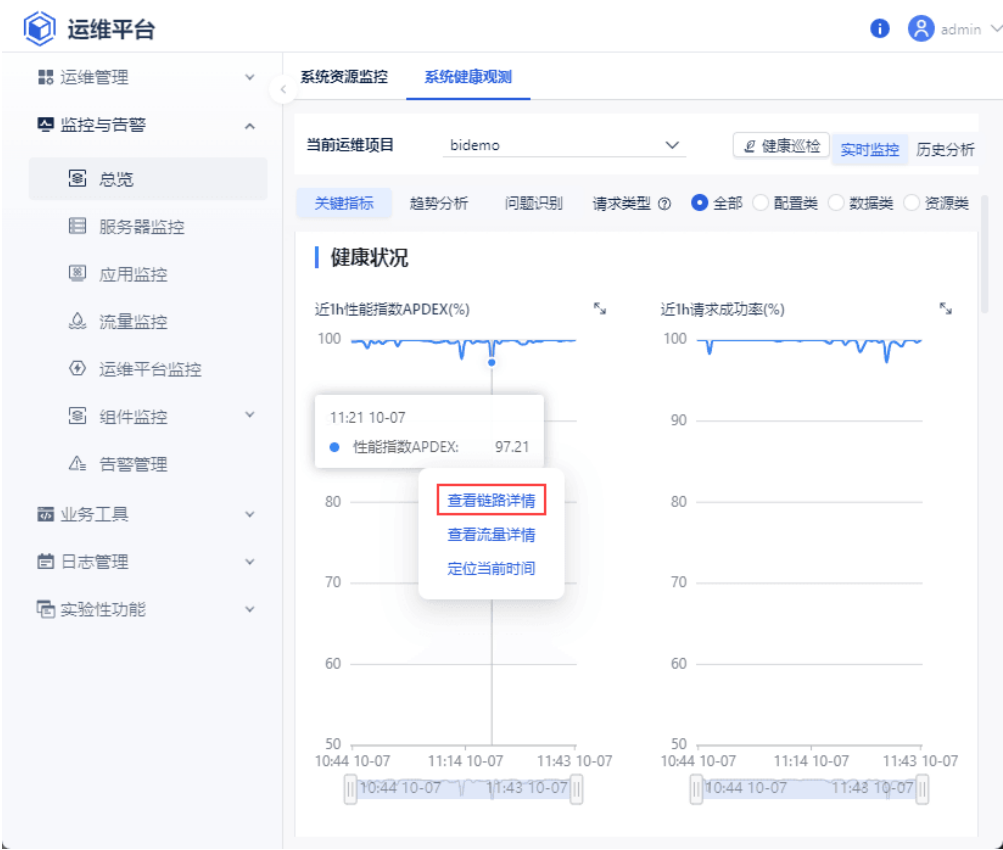
健康巡检



链路追踪

运维平台提供「链路追踪」功能，帮助用户对查询请求进行采集和分析。

对于系统健康观测各个图表异常处，可点击图表联动跳转查看具体链路，定位系统性能异常点。



对于实时监控下的问题请求列表，可点击跳转查看具体链路，定位系统性能异常点。

The screenshot shows the 'System Health Monitoring' interface with the 'Problem Requests' tab selected. The table below lists three entries, each with a 'View Link Details' link highlighted by a red box.

Component	Description	Status	Count 1	Count 2	Action
widget	2336290的分析/2000装机地图/组件	TIMEOUT	1	1	<a href="#">查看链路详情</a>
dashboard	715610的分析/第一个数据分析案例/第一个数据分析案例	TIMEOUT	1	1	<a href="#">查看链路详情</a>
dashboard	2092253的分析/分析完成/周销售+活动分析+大促同比/活动分析38-99	TIMEOUT	1	1	<a href="#">查看链路详情</a>

## 6.1.3 服务器监控



## 6.1.4 应用监控

## 6.1.5 流量监控

### 1. 概述

#### 1.1 版本

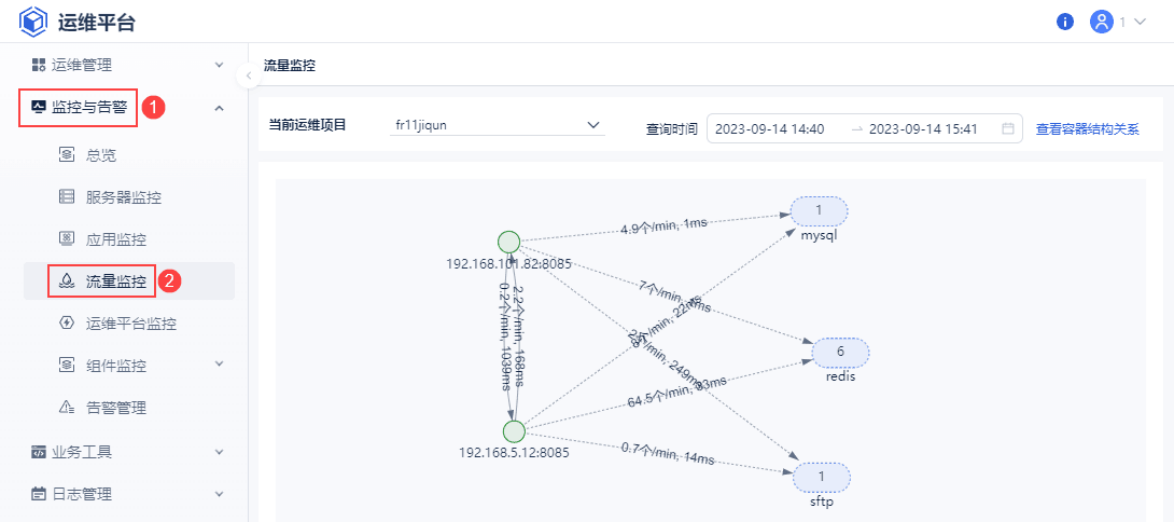
运维平台版本	功能变更
V1.6.0	-

#### 1.2 应用场景

对于集群工程，管理员希望了解各个工程节点和组件之间的请求流量情况，以便排查性能问题、联通性问题、节点平衡等。

#### 1.3 功能简介

FineOps 运维平台提供「流量监控」功能，帮助管理员查看不同时间段内各个节点间的请求流量情况。



### 2. 使用前提

流量监控功能与链路追踪功能互通，由多个运维组件提供功能：elasticSearch、skywalking\_oap。

如需完整使用「流量监控」功能，必须先启动这些组件，否则提示「链路追踪组件未启用，请在运维组件目录中启动 elasticsearch 以及 skywalking\_oap」。

管理员登录运维平台，点击「运维管理>运维组件」，先启动 elasticSearch 组件，再启动 skywalking\_oap 组件。

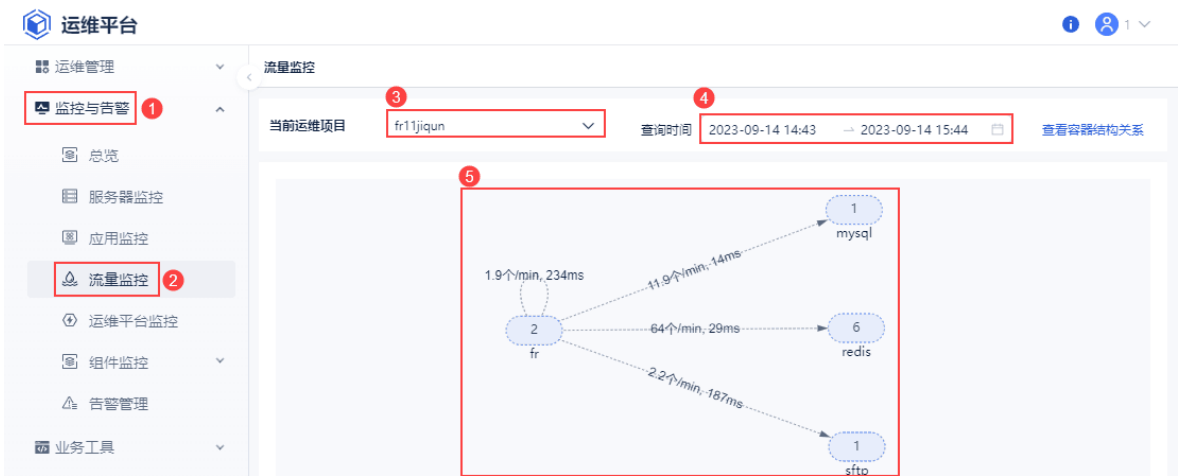


### 3. 功能入口

有三种方式进入流量监控界面。

#### 3.1 直接进入流量监控界面

- 1) 管理员登录运维平台，点击「监控与告警>流量监控」，选择需要进行查询的运维项目。
- 2) 选择查询时间：默认为当前时间前一分钟
- 3) 展示查询结果。如下图所示：



#### 3.2 系统健康观测异常点跳转

应用场景：

根据健康异常点，直接定位指定时间，快速查看流量情况。

操作步骤：

- 1) 在「监控与告警>总览>系统健康观测」界面，选择需要进行查询的运维项目。
- 2) 在实时监控下，点击各个折线图上的异常点，出现「查看流量详情」按钮。
- 3) 点击该按钮，即可自动跳转至「流量监控」界面，并带入运维项目和所选时间，展示流量情况。



### 3.3 项目总览界面跳转

#### 应用场景：

在监控集群项目组件是否联通时，可跳转查看各组件间的流量关系，辅助判断集群是否负载均衡。

#### 操作步骤：

- 1) 在「运维管理>项目总览」界面，选择需要进行查询的运维项目。
- 2) 点击「查看流量关系」按钮，即可自动跳转至「流量监控」界面。
- 3) 自动带入运维项目，查询时间为当前时间前一分钟，展示流量情况。



## 4. 展示信息

### 4.1 切换视图

展示各类组件之间的结构拓扑关系。

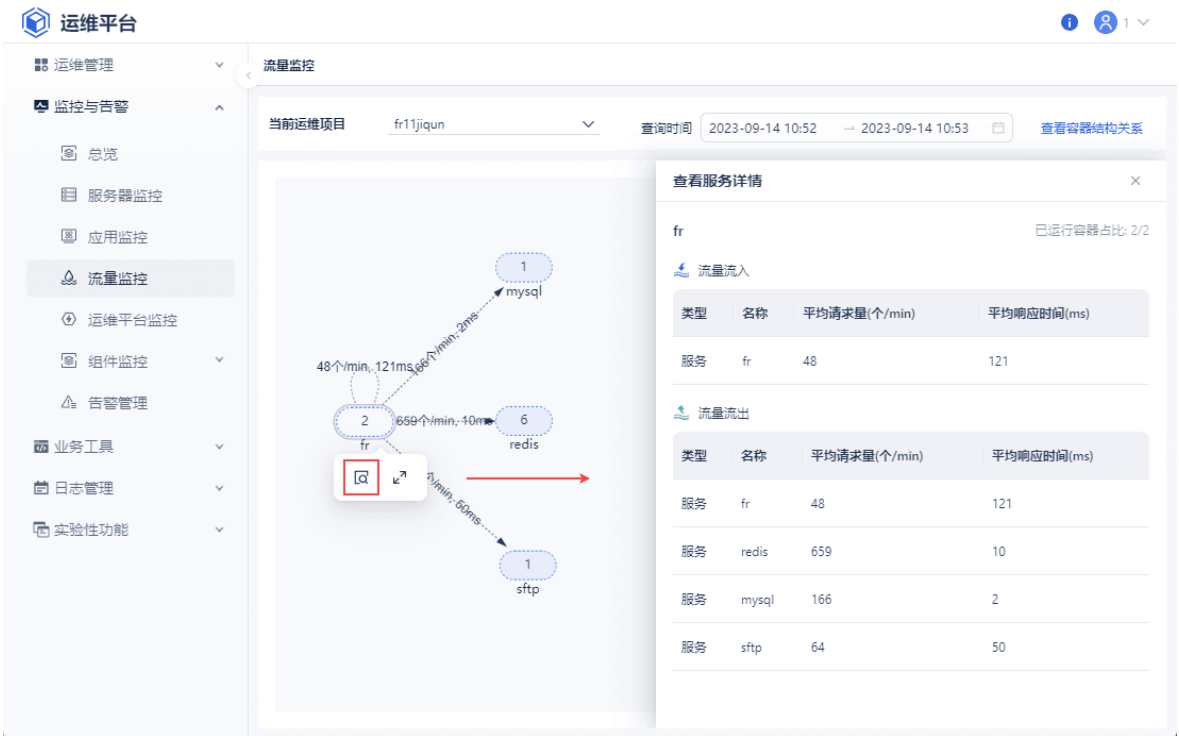
- 椭圆：代表单类组件，展示组件名称。点击可展开为此类组件下的多个容器。
- 圆圈：代表单个容器，展示容器的 IP 和端口。
- 连线：连线代表组件间存在流量流入/流出关系，箭头代表流向关系，连线上提示平均请求量和响应时间。



4.2 查看服务详情

选中拓扑图中的组件，点击「查看详情」，右侧可查看该组件流量流入流出的详情。

如下图所示：



信息

说明

组件名称

展示组件名称

已运行容器占比

此类组件下已运行容器数量/总容器数量

类型

流量类型

流量流入

名称

例如此处名称为 X

	代表 X 组件向拓扑图中所选组件发起了请求
	求
平均请求量(个/min)	代表所选时间内，每分钟流入到组件的请求数量
平均响应时间(ms)	代表所选时间内，所有请求的平均响应时间
类型	流量类型
	例如此处名称为 X
流量流出 名称	代表拓扑图中所选组件向 X 组件发起了请求
	求
平均请求量(个/min)	代表所选时间内，组件每分钟流出的请求数量
平均响应时间(ms)	代表所选时间内，所有请求的平均响应时间

## 6.1.6 运维平台监控

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.5.5	-

#### 1.2 功能简介

应用场景：

运维平台监控仪表板用于监控运维平台中各个运维组件的运行情况。

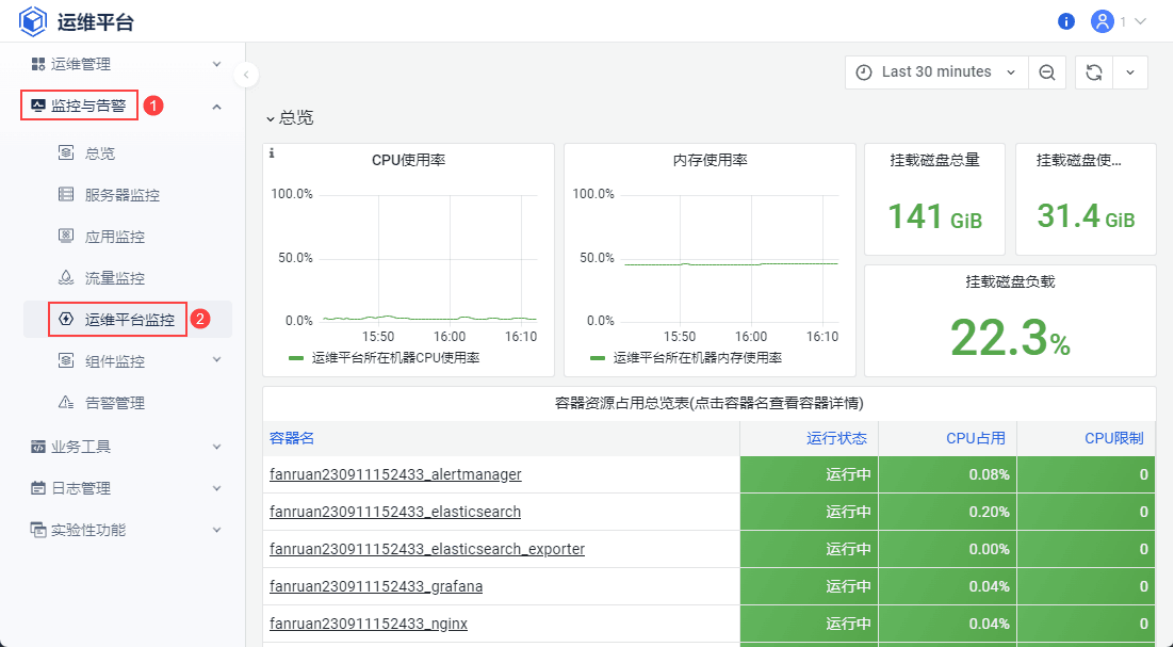
功能入口：

管理员登录 FineOps 运维平台，点击「监控与告警>运维平台监控」，即可查看。

仪表盘分为多个部分：总览、容器资源占用总览表、容器明细。

注意事项：

右上角支持切换查看不同时间段的监控数据，默认时间段为前 30 分钟，支持查看时间范围为前 15 天。



### 2. 使用前提

仪表盘功能，由这些运维组件提供：grafana、prometheus

因此如需正常查看相关仪表盘，需要确保这些组件正常运行。

管理员登录运维平台，点击「运维管理>运维组件」，可启动相关组件。





3. 总览

1) 仪表盘示例：

展示运维平台的整体监控看板，如果出现异常指标将标红/标黄显示。



2) 仪表板指标说明：

指标	具体说明
CPU 使用率	运维平台所在机器 CPU 使用率
内存使用率	运维平台所在机器内存使用率
挂载磁盘总量	运维平台挂载磁盘的空间总量
挂载磁盘使用量	运维平台挂载磁盘已使用空间占比
挂载磁盘负载	运维平台挂载磁盘的负载

4. 容器资源占用总览表

1) 仪表盘示例：

展示运维平台所管理组件的指标看板，如果出现异常指标将标红/标黄显示，通过点击容器名称可以联动下方明细指标。

容器资源占用总览表(点击容器名查看容器详情)															
容器名	运行状态	CPU占用	CPU限制	CPU超限周期	CPU超限时间	内存占用	内存使用率	内存限制	内存超限次数	网络接收	网络传输	磁盘写	磁盘读	磁盘读次数	磁盘写次数
fanruan230614103102_alertmanager	运行中	0.00%	0	0	0	12.65 ...	0.08%	15.67 GiB	0	0 B/s	0 B/s	0 B/s	0 B/s	0	0
fanruan230614103102_grafana	运行中	0.00%	0	0	0	64.40 ...	0.40%	15.67 GiB	0	7.39 B/s	9.12 B/s	0 B/s	0 B/s	0	0
fanruan230614103102_prometheus	运行中	0.00%	0	0	0	185.02 ...	1.15%	15.67 GiB	0	20.3 Ki...	928 B/s	0 B/s	0 B/s	0	0
fanruan230614103102_skywalking_o...	运行中	4.40%	0	0	0	1.20 GiB	7.65%	15.67 GiB	0	1.10 Ki...	1.14 KiB...	0 B/s	0 B/s	0	0
fanruan230614103102_elasticsearch	运行中	0.60%	0	0	0	2.80 GiB	17.90%	15.67 GiB	0	1017 B/s	1.23 KiB...	12.6 Ki...	0 B/s	0	2.50
fanruan230614103102_elasticsearch...	运行中	0.00%	0	0	0	10.94 ...	0.07%	15.67 GiB	0	636 B/s	654 B/s	0 B/s	0 B/s	0	0
fanruan230614103102_nginx	运行中	0.00%	0	0	0	81.73 ...	0.51%	15.67 GiB	0	416 B/s	422 B/s	0 B/s	0 B/s	0	0
fanruan230614103102_nginx_exporter	运行中	0.00%	0	0	0	11.94 ...	0.07%	15.67 GiB	0	51.3 B/s	60.7 B/s	0 B/s	0 B/s	0	0
fanruan230614103102_node_exporter	运行中	0.00%	0	0	0	17.55 ...	0.11%	15.67 GiB	0			0 B/s	0 B/s	0	0
fanruan230614103102_ops_Sbodaef...	未运行		0	0	0				0						
fanruan230614103102_ops	运行中	5.00%	0	0	0	1.73 GiB	11.07%	15.67 GiB	0	416 B/s	205 B/s	556 B/s	0 B/s	0	0.152
fanruan230614103102_ops_agent	运行中	0.20%	0	0	0	233.17 ...	1.45%	15.67 GiB	0	72.4 B/s	2.17 KiB...	0 B/s	0 B/s	0	0

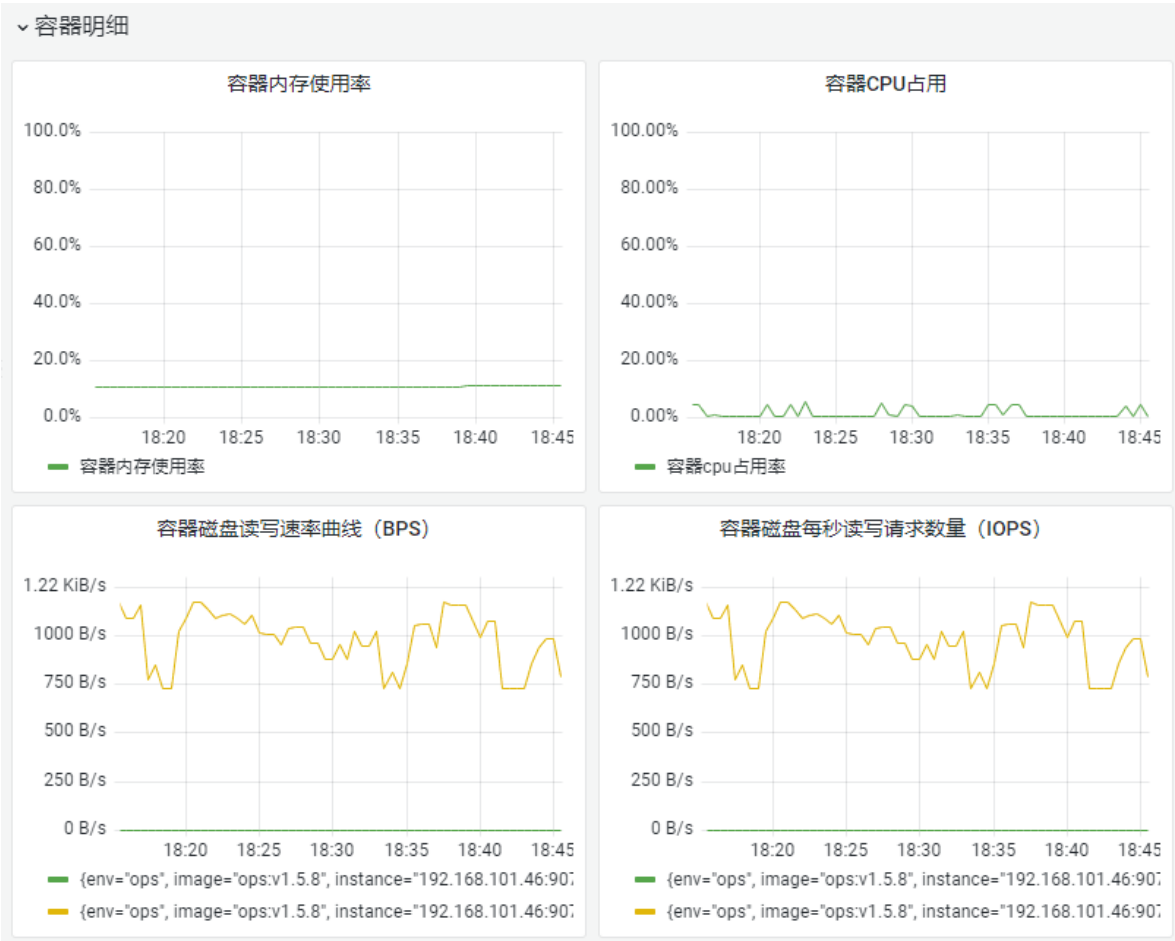
2) 仪表板指标说明：

指标	具体说明
容器名	运维组件名称
运行状态	运维组件是否运行
CPU 占用	运维组件占用 CPU
CPU 限制	运维组件 CPU 限制
CPU 超限周期	CPU 超过前面限制周期
CPU 超限时间	CPU 超过签名限制时间大小
内存占用	运维组件占用内存
内存使用率	运维组件内存使用率
内存限制	运维组件内存限制
内存超限次数	内存超过前面限制次数
网络接收	运维组件网络接收速率
网络传输	运维组件网络传输速率
磁盘写	磁盘的读速
磁盘读	磁盘的写速
磁盘读次数	磁盘读频率，固定时间内的次数
磁盘写次数	磁盘写频率，固定时间内的次数

5. 容器明细

1) 仪表盘示例：

展示运维组件容器的运行状况，跟随上方容器名点击联动。



2) 仪表板指标说明：

指标	具体说明
容器内存使用率	内存利用率：每个容器的内存限制、已用内存；内存饱和度：内存故障计数
容器 CPU 使用率	容器 cpu 的占用情况
容器磁盘读写速率曲线	网络上下行速率
容器磁盘每秒读写请求数量	网络读写请求数量

## 6.1.7 组件监控

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.5.5	-
V1.11.0	新增「加速引擎监控」，帮助管理员监控管理 BI 大数据引擎组件

#### 1.2 功能简介

##### 应用场景：

组件监控，主要用来监控运维项目中，各类组件（外接数据库、Nginx、Elasticsearch 等）的运行状况，使用情况等。

##### 功能入口：

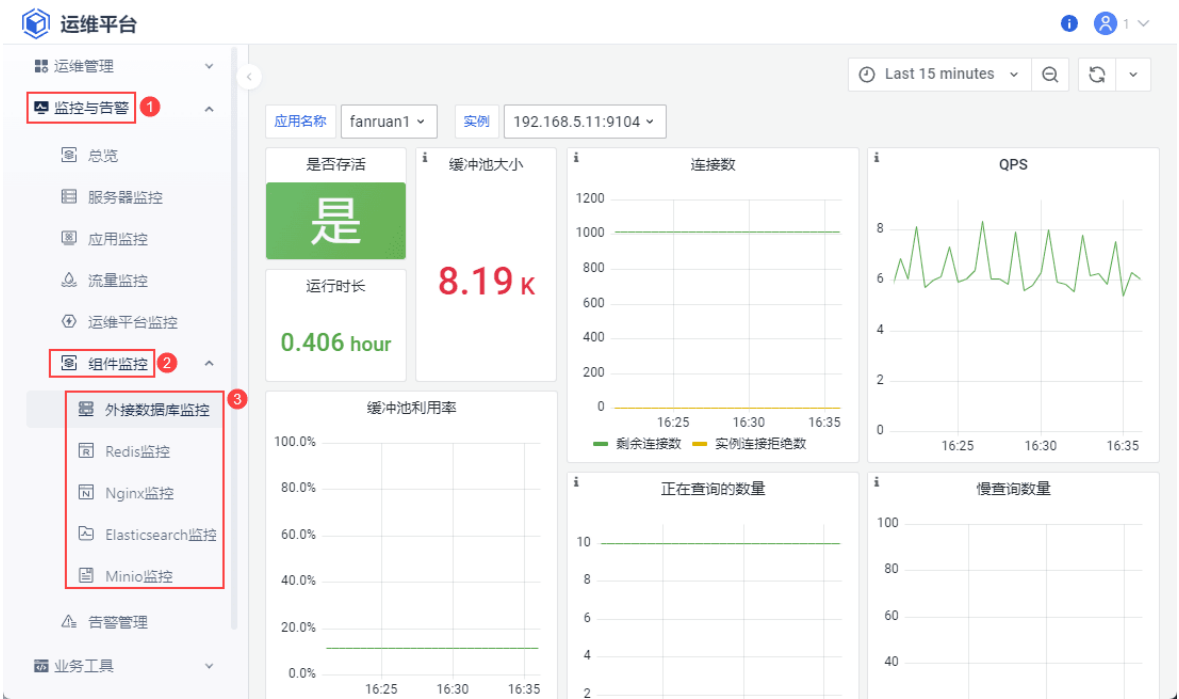
管理员登录 FineOps 运维平台，点击「监控与告警>组件监控」，即可查看各类组件监控仪表盘。

仪表盘分为多个部分：外接数据库监控、Redis 监控、Nginx 监控、Elasticsearch 监控、Minio 监控。

##### 注意事项：

右上角支持切换查看不同时间段的监控数据，默认时间段为前 30 分钟，支持查看时间范围为前 15 天。

左上角支持切换查看不同运维项目的监控数据。



## 2. 使用前提

### 2.1 运维项目要求

- **组件是否存在**

根据运维项目中的组件构成，会导致下文提及的部分仪表盘无数据。

例如若未安装 Elasticsearch 组件，则 Elasticsearch 监控仪表盘无数据。

例如若未安装 MinIO 组件，则 MinIO 监控仪表盘无数据。

- **非容器化项目安装 exporter 组件**

仪表盘中的数据，由相关 exporter 指标收集组件提供。容器化部署的帆软项目，默认安装了相关组件。

非容器化部署的帆软项目，在接入运维平台时，需要安装相关组件，否则无法正常使用本文功能。

请确保参考「[接入已有非容器化项目](#)」文档，安装了相关日志信息采集和服务器信息采集组件。

如未在接入时安装，可在「项目管理」中再次「编辑」该项目，进行安装设置，如下图所示：



2.2 运维平台要求

仪表盘功能， 由这些运维组件提供： grafana、prometheus

因此如需正常查看相关仪表盘， 需要确保这些组件正常运行。

管理员登录运维平台， 点击「运维管理>运维组件」， 可启动相关组件。



3. 外接数据库监控

外接数据库监控主要用来监控外接数据库的压力情况，避免因为外接数据库的不稳定而影响帆软系统的正常使用。

注：仅支持监控 MySQL5.6 以上版本外接数据库。

1) 仪表板示例：



2) 仪表板指标说明：

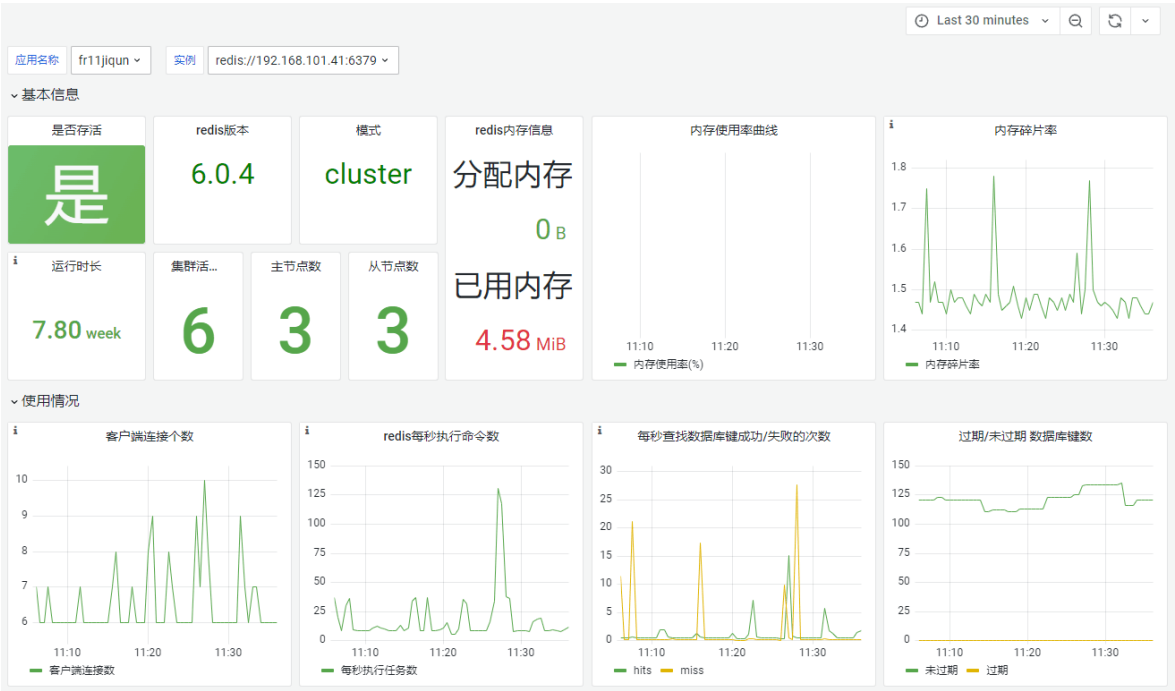
指标	具体说明
是否存活	外接数据库是否存活
运行时长	外接数据库举例上次启动的时长
缓冲池大小	缓冲池大小，内存允许的情况下缓冲池越大性能越好
缓冲池利用率	展示一段时间内缓冲池利用率的变化趋势
磁盘读取请求次数	磁盘读取请求次数，磁盘读取请求次数持续增长可能出现外接数据库性能差，磁盘 io 压力大等问题
连接数	展示外接数据库的剩余连接数和实例拒绝连接数，如果剩余连接数不足可能会出现应用访问异常
QPS	展示外接数据库每秒处理的请求数量，用来判断外接数据库的压力情况
正在查询的数量	展示外接数据库正在查询的语句数量，持续增长可能说明外接数据库超负荷
慢查询数量	展示外接数据库的慢查询数量，如果慢查询数量持续增长可能影响外接数据库的性能及可用性，需要关注具体慢查询的情况
死锁信息	展示数据库的死锁信息：  ts:检测到死锁的时间戳  thread:产生死锁的线程 id，这个 id 和 show processlist 里面的线程 id 是一致的  txn_id:innodb 的事务 ID  txd_time:死锁检查到前，事务执行时间

- user:执行 transction 的用户名
- db:发生死锁的 DB 名
- tbl:死锁发生的表名
- idx:产生死锁的索引名(在上面这个 demo 里面，我们直接走的主键，加的记录锁)
- lock\_type:锁的类型(记录锁，gap 锁，next-key 锁)
- lock\_mode:锁模式(S,X)
- wait\_hold:是否等着锁释放，一般死锁都是两个 wait
- victim:该会话是否做了牺牲，终止了执行
- query:造成死锁的 SQL 语句

4. Redis 监控

Redis 监控仪表板用于监控帆软系统使用的 Redis 状态服务器的运行情况。

1) 仪表板示例：



2) 仪表板指标说明：

指标	具体说明
是否存活	redis 的存活状态
redis 版本	redis 的版本
模式	单机还是集群
运行时长	redis 本次启动的运行时长
集群活跃节点数	集群活跃的节点数
主节点数	集群主节点数量
从节点数	集群从节点数量



redis 内存信息	redis 分配的内存和使用情况，通常来说 redis 内存占用比较小，如果出现内存使用率很高，那么可能存在异常
内存使用率曲线	一段时间内已用内存存在分配内存中的占比
内存碎片率	<p>内存碎片率表示「Redis 向操作系统中申请的内存」与「分配器分配的内存总量」的比值</p> <p>指数&gt;1 表明有内存碎片，越大表明越多，&lt;1 表明正在使用虚拟内存，虚拟内存其实就是硬盘，性能比内存低得多，这是应该增强机器的内存以提高性能。</p> <p>一般来说，内存碎片率数值在 1 ~ 1.5 之间是比较健康的。</p> <p>大于 1.5 表示，系统分配的内存大于 Redis 实际使用的内存，Redis 没有把这部分内存返还给系统，产生了很多内存碎片。在 Redis 4.0 版以前，只能通过安全重启解决这个问题。Redis 4.0 及以上版本可以支持内存自动清理。</p> <p>小于 1 表示，系统分配的内存小于 Redis 实际使用的内存，而 Redis 很有可能在使用 Swap 了！使用 swap 是相当影响性能的。</p>
客户端连接个数	用于查看当前 redis 的客户端连接个数，默认的客户端连接数最大限制为 10000，如果连接数过高，会影响 redis 吞吐量。>5000 时通常就需要进行告警
redis 每秒执行的命令数	展示一段时间内 redis 每秒执行命令数的曲线
每秒查找数据库键成功/失败的次数	展示每秒查找数据库的 hit/miss 的次数，用于参考根据业务需求调优 Redis 配置
过期/未过期数据库键数	展示 redis 中过期/未过期数据库键数，用于参考根据业务需求调优 Redis 配置

5. Nginx 监控

Nginx 监控仪表板用于监控帆软系统使用的 Nginx 服务器的运行情况，重点关注的是请求情况。

1) 仪表板示例：



2) 仪表板指标说明：

指标	具体说明
是否存活	Nginx 的存活状态
读取客户端的连接数	展示当前读取中的客户端连接数
回写客户端的连接数	展示当前回写中的客户端连接数
等待中的客户端的链接数	展示当前等待中的客户端连接数
用户请求总数	展示一段时间内的用户请求总数变化情况
活跃用户连接数	展示一段时间内活跃的用户连接数变化的情况
接收到的用户连接数	展示一段时间内接收到的用户连接数变化情况
处理的连接数	展示一段时间内处理的连接数变化情况

6. Elasticsearch 监控

注：若未安装 Elasticsearch 组件，则 Elasticsearch 监控仪表盘无数据。

1) 仪表板示例：

es / ElasticSearch仪表盘

Last 30 minutes

应用名称: fanruan1Interval: autoCluster: fanruan221110160628\_esNode name: All实例: 192.168.5.105:9114OSMySQLMongoDBApp

KPI

Cluster healthN/A

Trippe...0

CPU usage Avg.2%

JVM memory used Avg.65.9%

Nodes1

Data n...1

Pendin...0

Open file descriptors pe...1.62K

Shards

Active primary shards218

Active shards218

Initializing shards0

Relocating shards0

Delayed shards0

Unassigned shards157

JVM Garbage Collection

GC count

	min	max	avg	current
es-node0 - young	0	0.133	0.0258	0.133
es-node0 - old	0	0	0	0

GC time

	min	max	avg	current
es-node0 - young	0 s	1.27 ms	211 μs	1.27 ms
es-node0 - old	0 s	0 s	0 s	0 s

Translog

Total translog operations

	min	max	avg	total
es-node0	5	1.92 K	575	17.8 K

Total translog size in bytes

	min	max	avg	current
es-node0	29.7 KiB	2.11 MiB	727 KiB	2.06 MiB

Breakers

Tripped for breakers

	min	max	avg
es-node0: eqL_sequence	0	0	0
es-node0: fielddata	0	0	0
es-node0: inflight_requests	0	0	0
es-node0: model_inference	0	0	0
es-node0: parent	0	0	0
es-node0: request	0	0	0

Estimated size in bytes of breaker

	min	max	avg
es-node0: eqL_sequence	0 B	0 B	0 B
es-node0: fielddata	2.74 KiB	2.74 KiB	2.74 KiB
es-node0: inflight_requests	0 B	5.73 MiB	189 KiB
es-node0: model_inference	0 B	0 B	0 B
es-node0: parent	480 MiB	1.56 GiB	1.04 GiB
es-node0: request	0 B	0 B	0 B
es-node0: limit for eqL_sequence	1 GiB	1 GiB	1 GiB

CPU and Memory

Load average

	min	max	avg	current
load5: es-node0	1.51	2.46	2.07	2.07
load1: es-node0	0.880	3.25	2.04	1.96
load15: es-node0	1.83	2.18	2.02	2.02

CPU usage

	min	max	avg	current
es-node0	0%	2%	0.581%	2%

JVM memory usage

	min	max	avg	current
es-node0 max: heap	2 GiB	2 GiB	2 GiB	2 GiB
es-node0 peak used pool: young	1.19 GiB	1.19 GiB	1.19 GiB	1.19 GiB
es-node0 used: heap	480 MiB	1.56 GiB	1.04 GiB	1.11 GiB
es-node0 peak used pool: old	610 MiB	610 MiB	610 MiB	610 MiB

JVM memory committed

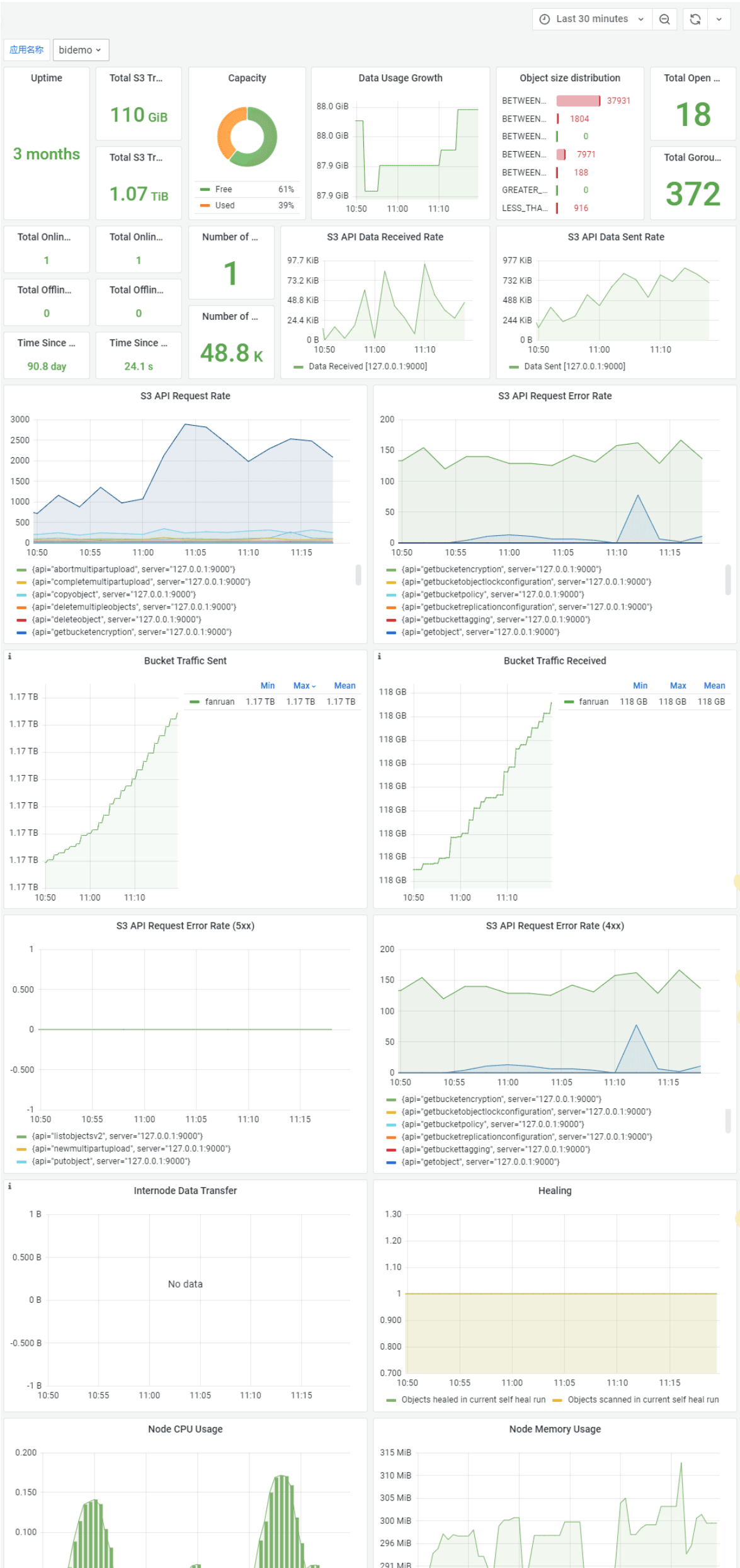
	min	max	avg	current
es-node0 max: heap	2 GiB	2 GiB	2 GiB	2 GiB
es-node0 committed: heap	2 GiB	2 GiB	2 GiB	2 GiB
es-node0 committed: non-heap	228 MiB	228 MiB	228 MiB	228 MiB

Disk and Network

## 7. Minio 监控

注：若未安装 Minio 组件，则 Minio 监控仪表盘无数据。

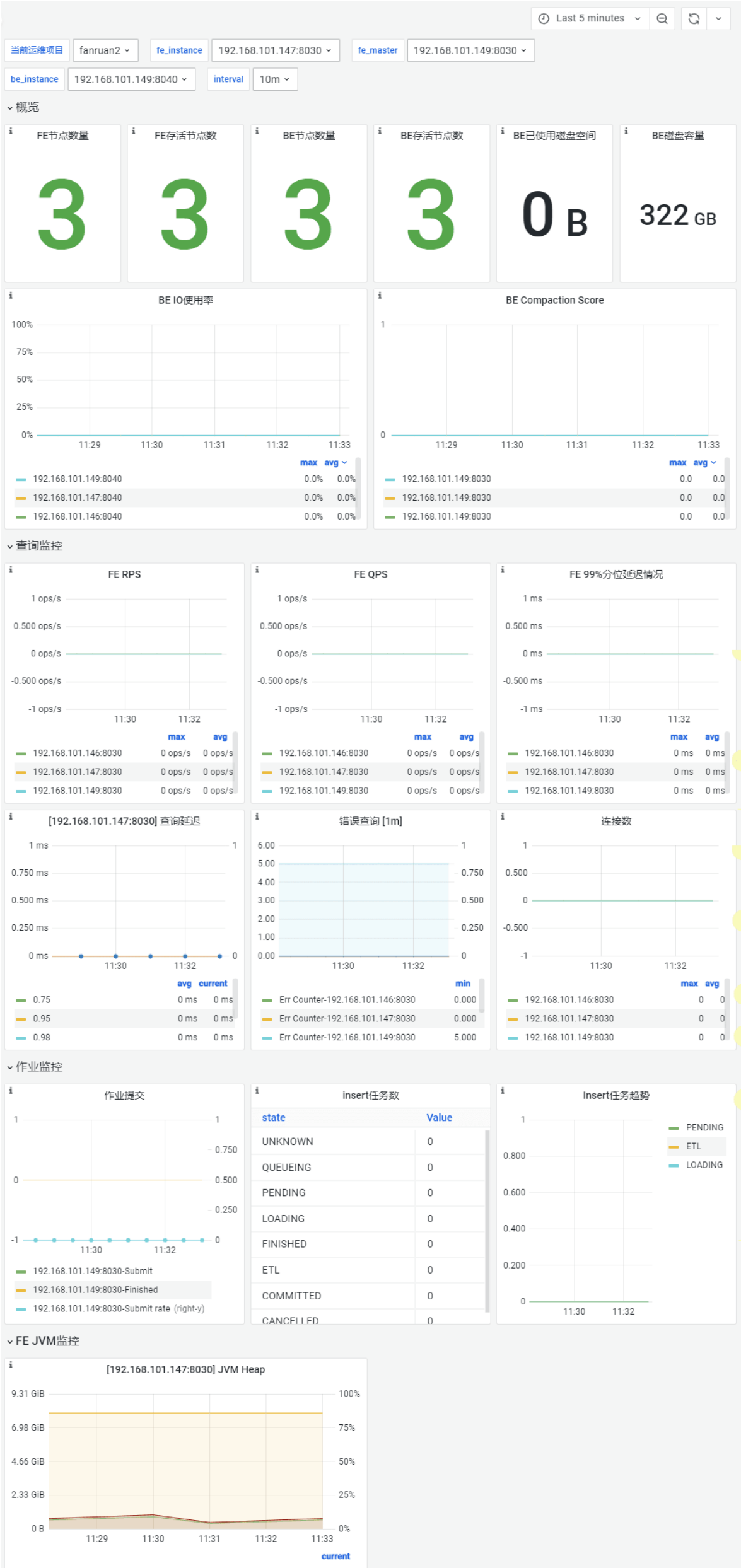
### 1) 仪表板示例：



## 8. 加速引擎监控

注：若未安装 StarRocks 组件，则加速引擎监控仪表盘无数据。

### 1) 仪表板示例：







## 6.2 警告

## 6.2.1 告警任务

### 1. 概述

#### 1.1 版本

运维平台 版本	功能变更
V1.5.5	-
V1.7.0	支持自定义告警
V1.9.0	「负载过高告警——节点无响应」更名为「应用卡顿告警——节点无响应」，降低用户理解成本
V1.11.0	新增默认告警「运维平台挂载磁盘预警」，覆盖通用运维场景
V1.14.0	「应用卡顿告警——节点无响应」触发逻辑优化：异常状态持续触发时长从 30s 优化为 60s

#### 1.2 应用场景

当运维项目遇到高负载、宕机风险、磁盘空间不足等问题时，需要及时通知管理员。FineOps 提供告警管理功能，提供内置的默认告警任务。用户也可自定义异常触发条件和达成告警的持续时间。



### 2. 默认告警

管理员登录运维平台，点击「监控与告警>告警管理>告警任务>默认告警」，可查询

内置的告警规则。如下图所示：



2.1 任务一览

运维平台内置的规则如下表所示，不支持更改。

分类	告警名称	判断逻辑
应用告警	应用打开文件数不足	异常项：此应用节点打开文件数已超过 85%，存在导致应用节点故障的风险 触发时长：异常状态持续超过 2min 处理建议：请及时调整对应应用的最大文件打开数 nofile ，修改方法参考 <a href="#">Linux 最大打开文件数</a>
	应用卡顿	异常项：应用节点响应异常
	告警——节点无响应	触发时长：异常状态持续超过 60s 处理建议：请访问应用节点检查其是否存在异常或及时重启异常节点，避免影响系统使用
	系统宕机告警——无响应	异常项：应用节点发生宕机 触发时长：异常状态持续超过 2min 处理建议：请尽快重启并排查宕机问题，避免影响系统使用
	http 线程数阻塞	异常项：http 线程数已经使用超过 85%，可能导致应用请求阻塞 触发时长：异常状态持续超过 30s 处理建议：请排查是否由于系统使用并发量增加导致，如因并发过大，请

	尽快提升 http 线程最大值，如非并发增大可能出现了业务异常，建议使用堆栈分析定位
	异常项：工程磁盘剩余空间不足 15%
工程磁盘空间预警	<p>触发时长：异常状态持续超过 5min</p> <p>处理建议：当前服务器工程磁盘剩余空间不足 15%，请及时清理磁盘或增加磁盘空间</p> <p>异常项：应用节点 FullGC 发生异常</p> <p>触发时长：5min 内，特定标签 target_belong_app 和 instance 下的 JVM 全量垃圾回收，次数总和大于 3 次且时间总和超过 50s</p> <p>处理建议：当前应用节点发生连续 FullGC，且 FullGC 持续时间过长，可能导致访问出现明显卡顿并有宕机风险，请及时排查系统使用问题并判断是否需要重启恢复使用</p> <p>异常项：项目当前性能指数过低，可能出现访问异常（性能指数 APDEX 小于 90%且 CPM 每分钟事务数大于 100)</p>
性能告应用访问异常	<p>触发时长：异常状态持续超过 3min</p> <p>处理建议：请尽快关注系统健康观测页面中的性能指数趋势，如持续降低需确认应用是否出现访问异常，并尽快输出堆栈、dump 等日志排查故障或重启恢复使用</p> <p>异常项：服务器 cpu 使用率持续过高（超过 90%）</p>
服务器 cpu 预警	<p>触发时长：异常状态持续超过 6min</p> <p>处理建议：请检查是否存在异常进程并修复、减少服务器内的业务用量或尽快扩容服务器，避免影响系统使用</p> <p>异常项：服务器内存占用持续过高（超过 90%）</p>
服务器内存预警	<p>触发时长：异常状态持续超过 6min</p> <p>处理建议：请检查是否存在异常进程并修复、减少服务器内的业务用量或尽快扩容服务器，避免影响系统使用</p> <p>异常项：服务器打开文件数已超过 85%，存在导致系统故障的风险</p>
硬件告预警	<p>触发时长：异常状态持续超过 2min</p> <p>处理建议：请及时调整对应节点的最大文件打开数 file_max，修改方法参考 <a href="#">Linux 最大打开文件数</a></p> <p>异常项：应用节点和外置数据库连通异常</p>
组件告外置库连通性告警	<p>触发时长：异常状态持续超过 30s</p> <p>处理建议：请检查外置库状态是否异常，如无异常请排查异常节点与外置</p>

	数据库的网络问题
	异常项：外置库状态异常
外置库状态异常告警	<div>触发时长：异常状态持续超过 30s</div> <div>处理建议：请尽快检查外置库状态并恢复外置库访问，避免影响帆软应用提供服务</div>
	异常项：应用节点和状态服务器 Redis 连通异常
状态服务器连通性告警	<div>触发时长：异常状态持续超过 30s</div> <div>处理建议：请检查状态服务器 Redis 是否异常，如无异常请排查异常节点与状态服务器的网络问题</div>
	异常项：状态服务器 Redis 内存使用率超过 85%
状态服务器内存占用过高	<div>触发时长：异常状态持续超过 5min</div> <div>处理建议：请排查状态服务器 Redis 使用是否正常，如使用正常建议后续增加 Redis 内存配置避免 Redis 压力过大影响帆软应用提供服务</div>
状态服务器状态异常告警	<div>异常项：状态服务器 Redis 异常</div> <div>触发时长：异常状态持续超过 30s</div> <div>处理建议：请尽快检查状态服务器并恢复状态服务器使用，避免影响帆软应用提供服务</div>
	异常项：应用节点和文件服务器连通异常
文件服务器连通性告警	<div>触发时长：异常状态持续超过 30s</div> <div>处理建议：请检查文件服务器是否异常，如无异常请排查异常节点与文件服务器的网络问题</div>
	异常项：Nginx 状态异常
Nginx 状态异常告警	<div>触发时长：异常状态持续超过 30s</div> <div>处理建议：请尽快检查 Nginx 并恢复 Nginx 使用，避免影响帆软应用提供服务</div>
	异常项：运维平台挂载磁盘剩余空间不足 15%
	触发时长：异常状态持续超过 5min
运维平台挂载磁盘预警	处理建议：运维平台挂载磁盘剩余空间不足 15%，磁盘空间不足可能导致运维平台出现故障，请及时清理磁盘或增加磁盘空间

2.2 管理任务

1) 任务状态

告警任务有四种状态：

- 正常：任务已启用，适用项目均未触发限制项。
- 异常：任务已启用，存在适用项目触发了限制项，但未达到告警通知的持续时长。

- 告警：任务已启用，存在适用项目触发了限制项，且已达到告警通知的持续时长。
- 禁用：任务未启用

2) 任务启停

- 对于默认告警，支持启动、禁用操作。
- 不支持删除、修改默认告警任务，不支持修改任务分组。



3) 查看告警任务详情

- 按照分组存放并展示每个任务的简要信息：告警名称、当前状态、适用项目、标签、触发条件、是否启用
- 可点击「查看告警详情」按钮，查看本条告警任务的详细信息和告警通知模板



4) 查看告警记录

- 可点击「查看告警记录」按钮，查看曾触发本条告警任务的明细信息，了解触发项目、触发时间等。



### 3. 自定义告警

#### 3.1 添加任务

##### 1) 创建告警

管理员登录 FineOps 运维平台, 点击「监控与告警>告警管理>告警任务>自定义告警」, 点击「创建告警」按钮, 进入配置界面。



##### 2) 告警任务设置

在「告警任务设置」页面, 支持设置任务的基础信息和告警通知的文本。



点击「保存」，完成任务设置。

设置项	说明
告警名称	该告警任务的名称 不支持重名
通用项目	可下拉选择该告警任务生效的项目，支持多选 为空代表不限制，全选项目
分组	可下拉选择该告警任务所属的分组 支持添加分组，分组名称不支持重复，不支持多选 如添加了分组，但不存在任何一个告警任务使用到该分组，保存任务时该分组不会保存
标签	可下拉选择该告警任务的标签 支持添加标签，标签名称不支持重复，不支持多选 如添加了标签，但不存在任何一个告警任务使用到该标签，保存任务时该标签不会保存
支持对以下指标设置告警判断条件和触发时长	

触发条件	• 服务器内存使用率
	• 服务器 CPU 使用率
	• 服务器剩余文件打开数
	• 近 5 分钟 FullGC 总耗时
	• 近 5 分钟 FullGC 次数
	• 应用内存使用率
	• 应用 CPU 使用率
	• 应用剩余文件打开数
	• 应用工程目录磁盘使用率
	• 应用 http 线程池使用率

设置告警通知中的文本内容

支持输入文本和参数，支持换行，支持的参数包括：

通知文本	<ul style="list-style-type: none"><li>项目名称：{{ \$labels.target_belong_app }}</li><li>节点名称：{{ \$labels.ClusterNodeName }}</li><li>节点地址：{{ \$labels.instance }}</li></ul>
------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------

运维平台

告警任务设置

取消

保存

基础信息

告警名称

应用内存使用率过高

适用项目

bi X

分组

应用节点告警

标签

正式

触发条件

应用内存使用率

大于

0.8

且持续时间>

5

分钟

告警通知

通知文本

亲爱的管理员  
当前应用节点内存使用率超过80%，请及时关注。  
项目名称: {{ \$labels.target\_belong\_app }}  
节点地址: {{ \$labels.instance }}  
节点名: {{ \$labels.ClusterNodeName }}

通知模板

```
{{ define "fine_notification.html" }}  
[[ range .Alerts ]]  
<pre>  
[[ .Labels.target_belong_app ]]项目出现告警 //告警项目信息  
  
告警项: [[ .Labels.alertname ]] //告警名称  
  
[[ .Annotations.description ]] //告警详情
```

3) 效果预览

- 保存的告警任务，展示在「自定义告警」的指定分组中。
- 展示该告警任务的适用项目、标签、触发条件、任务状态等信息，任务默认启用。

运维平台

告警任务

告警记录

告警通知设置

默认告警

自定义告警

创建告警

应用节点告警(1)

正常

应用内存使用率过高

适用项目: bi

标签: 正式

触发条件: 应用内存使用率 > 0.8, 持续时间 > 5min

🔍

🔗

🗑️

🔴

🟢

3.2 管理任务

1) 任务状态

告警任务有四种状态：

- 正常：任务已启用，适用项目均未触发限制项。

566

- 异常：任务已启用，存在适用项目触发了限制项，但未达到告警通知的持续时长。
- 告警：任务已启用，存在适用项目触发了限制项，且已达到告警通知的持续时长。
- 禁用：任务未启用

2) 任务启停删除

- 对于自定义告警，支持启动、禁用和删除操作。

3) 修改分组名称

- 对于自定义告警分组，支持修改分组名称。



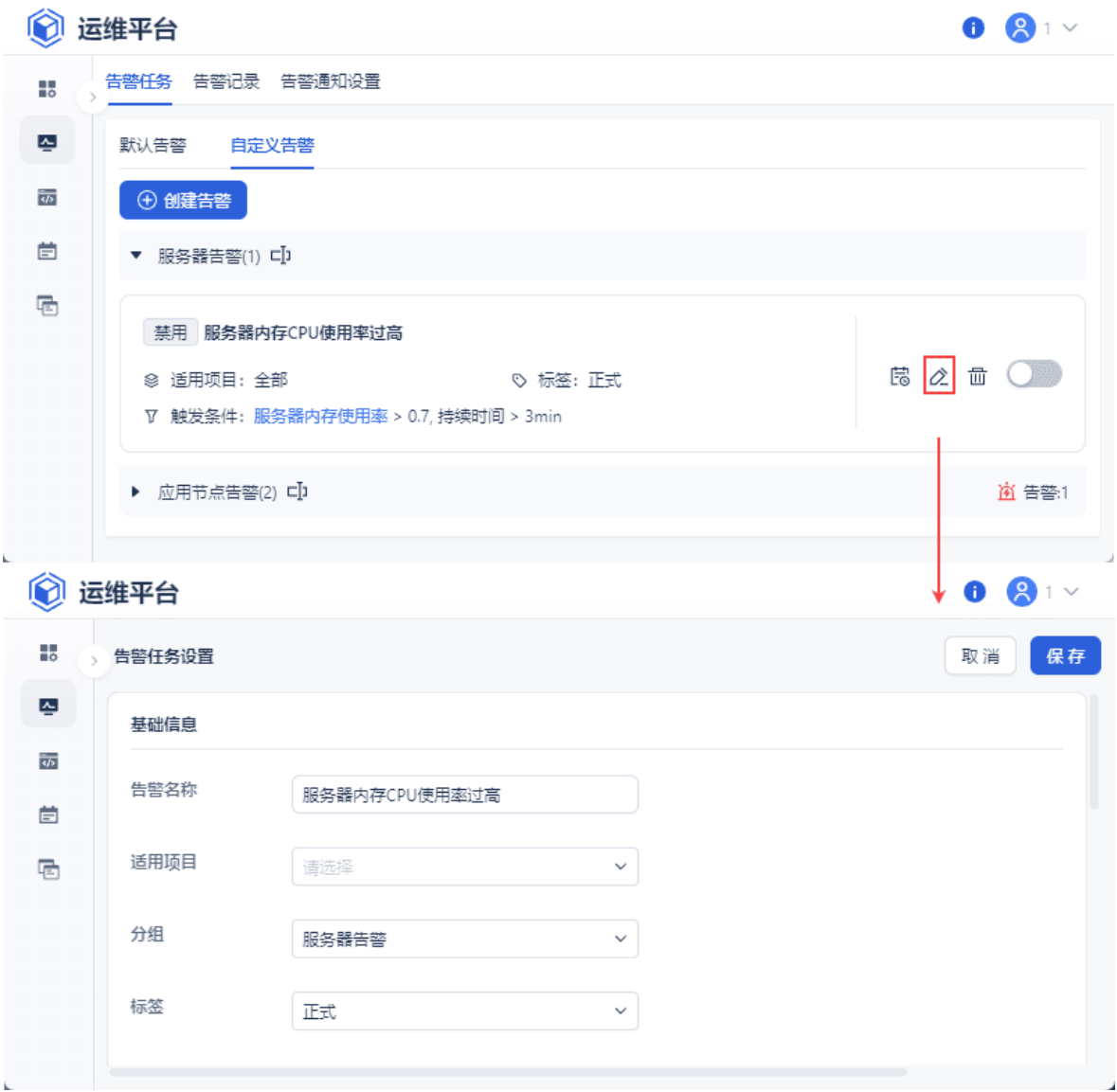
4) 查看告警记录

- 可点击「查看告警记录」按钮，查看曾触发本条告警任务的明细信息，了解触发项目、触发时间等。



### 5) 编辑告警

- 可点击「编辑告警」按钮，编辑本条告警任务的基础信息和告警通知文本。
- 编辑告警，不改变告警任务原本的启停状态。



## 6.2.2 告警记录

## 6.2.3 告警方式

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.5.5	-
V1.7.0	告警方式页面视觉优化，功能无变更
V1.10.0	新增支持告警类型：企业微信机器人、钉钉机器人、飞书机器人

#### 1.2 应用场景

在应用程序出现高负载、宕机风险、磁盘空间不足等重要事件时，需要确保管理员及时收到关键的告警信息，以便能够快速响应和处理问题，从而最大限度地减少应用程序的中断时间和减轻潜在的损失。

支持通过多种方式通知管理员：邮箱、webhook、企业微信、企业微信机器人、钉钉机器人、飞书机器人。

### 2. 邮箱告警

如需使用邮箱告警，需配置发件人和收件人。

#### 2.1 配置发件人

发邮件，需要有一个发件人，否则发送动作无法完成。

- 1) 管理员登录「运维平台」，点击「监控与预警>告警配置>告警通知设置」。
- 2) 点击「添加发件人」，配置发件人账户，点击「确定」。

注 1：发件人邮箱类型不同，配置步骤可能不完全相同，具体请参见：[邮箱](#)。  
注 2：配置完成后，请点击「发送测试邮件」，确保邮箱可使用。



3) 设置「系统默认发件人」。



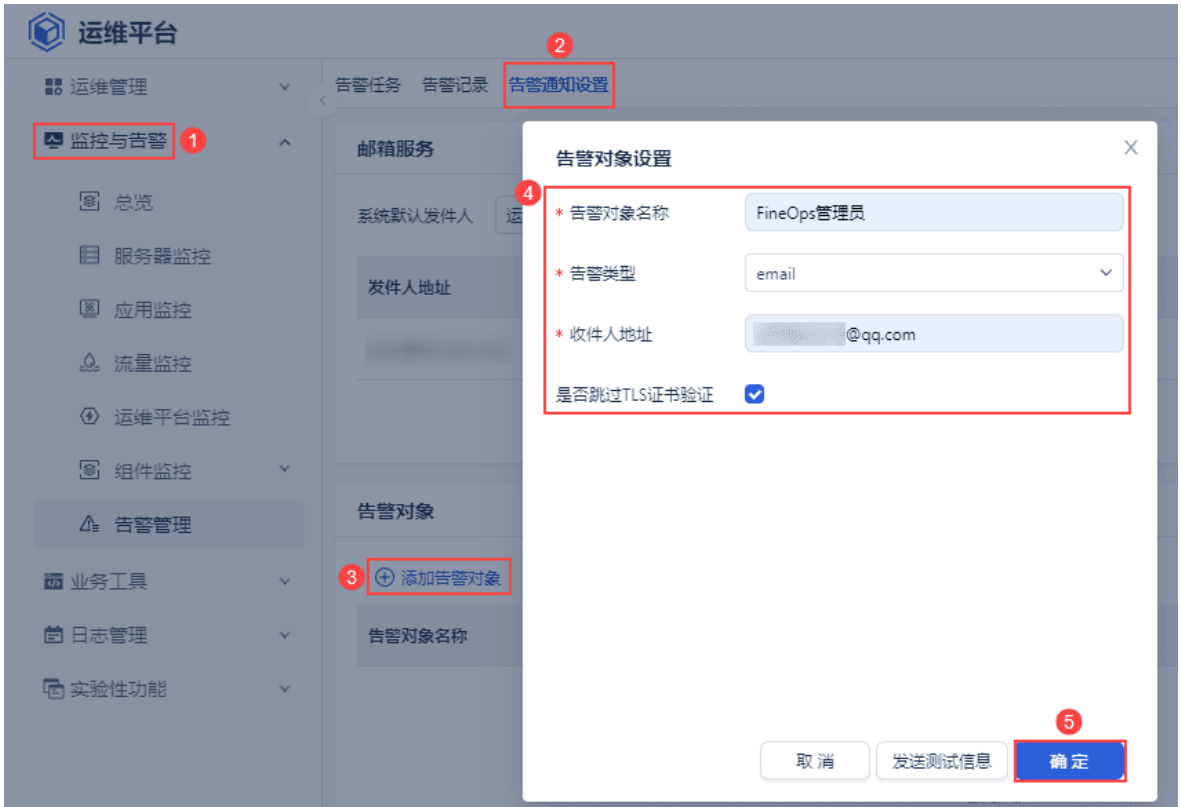
## 2.2 配置告警收件人

发邮件，需要配置收件人，接收告警信息。

- 1) 管理员登录「运维平台」，点击「监控与预警>告警配置>告警规则」，点击「添加告警对象」。
- 2) 设置告警类型为「email」，设置告警对象名称和收件人地址，点击「确定」，点击「保存」。如下图所示：

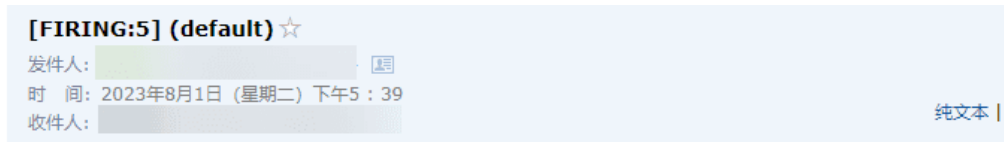
注：配置完成后，请点击「发送测试邮件」，确保邮箱可使用。





## 2.3 效果预览

当 FineOps 监测到项目触发告警任务后，发送邮件给收件人，如下图所示：



bi6jiqun项目出现告警 //告警项目信息

告警项：系统宕机告警——无响应 //告警名称

项目：bi6jiqun

应用节点：[REDACTED]

异常项：应用节点发生宕机

处理建议：请尽快重启并排查宕机问题，避免影响系统使用 //告警详情

告警触发时间：

2023-08-01T11:25:12 UTC+8

bi6jiqun项目出现告警 //告警项目信息

告警项: 负载过高告警——节点无响应 //告警名称

项目: bi6jiqun

应用节点:

异常项: 应用节点响应异常

处理建议: 请访问应用节点检查其是否存在异常或及时重启异常节点, 避免影响系统使用 //告警详情

告警触发时间:

2023-08-01T11:23:42 UTC+8

### 3. webhook 告警

### 3.1 配置告警对象

#### 1) 准备 Webhook 的 URL

管理员需要先准备好一个 Webhook 的 url，当有新数据的告警消息时，运维平台会往这个 url 发数据，提醒用户告警。

由于各人诉求不同，帆软不提供 Webhook 的 url 搭建获取步骤，请自行准备。

#### 2) 添加告警对象

管理员登录「运维平台」，点击「监控与预警>告警管理>告警通知设置」，点击「添加告警对象」。

设置告警类型为「webhook」，设置告警对象名称和 Webhook 的相关信息。

注：若 Webhook 的接口存在验证，需要配置用户名及密码。

#### 3) 发送测试信息

点击「发送测试邮件」，访问 Webhook 的 url，确保收到测试通知，确保 Webhook 可使用。

#### 4) 保存

点击「确定」，即可完成 webhook 告警对象的配置。



### 3.2 效果预览

当 FineOps 监测到项目触发告警任务后，发送消息到 Webhook 的 url，如下图所示：

REQUESTS (2/500)  
Newest First  
Search Query

POST #b537b  
221.228.203.2  
2023/08/01 18:31:08

POST #d1483  
221.228.203.2  
2023/08/01 18:30:51

First

Prev

Next

Raw Content

Format JSONWord-WrapCopy

```
{
  "receiver": "fine_channel",
  "status": "firing",
  "alerts": [
    {
      "status": "firing",
      "labels": {
        "alertname": "系统宕机告警——无响应",
        "instance": "192.168.101.82:8085",
        "job": "app_schedule__fb2ef781-9133-464c-bc65-90ad005929e6",
        "source": "default",
        "target_belong_app": "fr11"
      },
      "annotations": {
        "description": "项目：bi6jqun \n应用节点：192.168.101.82:8085 \n异常项：应用节点发生宕机 \n处理建议：请尽快重启并排查宕机问题，避免影响系统使用"
      },
      "startsAt": "2023-08-24T06:38:57.192Z",
      "endsAt": "0001-01-01T00:00:00Z",
      "generatorURL": "http://42956542b88e:9090/graph?g0.expr=up%7Bjob%3D~%22.%2Aapp_schedule.%2A%22%7D+%3D%3D+0&g0.tab=1",
      "fingerprint": "63131302d98ec860"
    },
    {
      "status": "firing",
      "labels": {
        "alertname": "负载过高告警——节点无响应",
        "instance": "192.168.101.82:8085",
        "job": "app_schedule__fb2ef781-9133-464c-bc65-90ad005929e6",
        "source": "default",
        "target_belong_app": "bi6jqun"
      },
      "annotations": {
        "description": "项目：bi6jqun \n应用节点：192.168.101.82:8085 \n异常项：应用节点响应异常 \n处理建议：请访问应用节点检查其是否存在异常或及时重启异常节点，避免影响系统使用"
      }
    }
  ]
}
```

注：此处附上一个告警内容示例。仅供参考：

点击展开更多

```
{
  "receiver": "fine_channel",
  "status": "firing",
  "alerts": [
    {
      "status": "firing",
      "labels": {
        "alertname": "系统宕机告警——无响应",
        "instance": "192.168.101.82:8085",
        "job": "app_schedule__fb2ef781-9133-464c-bc65-90ad005929e6",
        "source": "default",
        "target_belong_app": "fr11"
      },
      "annotations": {
        "description": "项目：fr11 \n 应用节点：192.168.101.82:8085 \n 异常项：应用节点发生宕机 \n 处理建议：请尽快重启并排查宕机问题，避免影响系统使用"
      },
      "startsAt": "2023-08-24T06:38:57.192Z",
      "endsAt": "0001-01-01T00:00:00Z",
      "generatorURL": "http://42956542b88e:9090/graph?g0.expr=up%7Bjob%3D~%22.%2Aapp_schedule.%2A%22%7D+%3D%3D+0&g0.tab=1",
      "fingerprint": "63131302d98ec860"
    },
    {
      "status": "firing",
      "labels": {
        "alertname": "负载过高告警——节点无响应",
        "instance": "192.168.101.82:8085",
        "job": "app_schedule__fb2ef781-9133-464c-bc65-90ad005929e6",
```

```
"source": "default",
"target_belong_app": "fr11"
},
"annotations": {
"description": "项目：fr11 \n 应用节点：192.168.101.82:8085 \n 异常项：应用节点响
应异常 \n 处理建议：请访问应用节点检查其是否存在异常或及时重启异常节点，避
免影响系统使用"
},
"startsAt": "2023-08-24T06:37:27.192Z",
"endsAt": "0001-01-01T00:00:00Z",
"generatorURL": "http://42956542b88e:9090/graph?g0.expr=up%7Bjob%3D~%22.%2A
app_schedule.%2A%22%7D+%3D%3D+0&g0.tab=1",
"fingerprint": "8b12f8d538c68f97"
},
{
"status": "firing",
"labels": {
"ClusterID": "2076660091",
"ClusterNodeID": "5e971cea-8525-73a6-65cf-9efd4c0cb1c5",
"ClusterNodeName": "12",
"IP": "192.168.5.12",
"Port": "8085",
"ServletURL": "/webroot/tomcat",
"alertname": "工程磁盘空间预警",
"instance": "192.168.5.12:8085",
"job": "app_schedule__fb2ef781-9133-464c-bc65-90ad005929e6",
"source": "default",
"target_belong_app": "fr11"
},
"annotations": {
"description": "项目：fr11 \n 应用节点：192.168.5.12:8085 \n 异常项：工程磁盘剩余
空间不足 15% \n 处理建议：当前服务器工程磁盘剩余空间不足 15%，请及时清理磁
盘或增加磁盘空间"
},
"startsAt": "2023-08-24T07:19:57.192Z",
"endsAt": "0001-01-01T00:00:00Z",
"generatorURL": "http://42956542b88e:9090/graph?g0.expr=disk_use_project+%2F+dis
k_space_project+%3E+0.85&g0.tab=1",
"fingerprint": "e5b6ca5c09f2dbcb"
}
],
"groupLabels": {},
"commonLabels": {
"job": "app_schedule__fb2ef781-9133-464c-bc65-90ad005929e6",
"source": "default",
"target_belong_app": "fr11"
},
"commonAnnotations": {},
```

```
"externalURL": "http://3ef45ec83ecc:9093",
"version": "4",
"groupKey": "{}:{}",
"truncatedAlerts": 0
}
```

### 4. 企业微信告警

支持通过企业微信发送告警通知。

管理员需要先在企业微信端创建一个用于发送告警的应用，并获得相关应用参数，方可进入运维平台配置告警对象。

#### 4.1 微信后台创建应用

企业微信管理员登录 [企业微信管理后台](#)，在「应用管理」Tab 下点击「创建应用」，如下图所示：



上传应用 logo，填写应用名称，应用介绍选填，设置可见范围，设置完成后，点击「创建应用」，如下图所示：

建议将可见范围设置为部门/标签。若「可见范围」是单独的微信用户，那么在平台「微信管理」的「成员管理」Tab 下会看不到用户信息。



## 4.2 获取微信参数

管理员需要在企业微信后台获取以下参数数据。

参数名	含义
CorpID	企业 ID
应用 AgentID	用于发送告警的应用的 AgentID
应用 Secret	用于发送告警的应用的 Secret
接收对象	用于接收告警的部门或用户 ID，请确保该部门/用户在上文设置的应用可见范围内

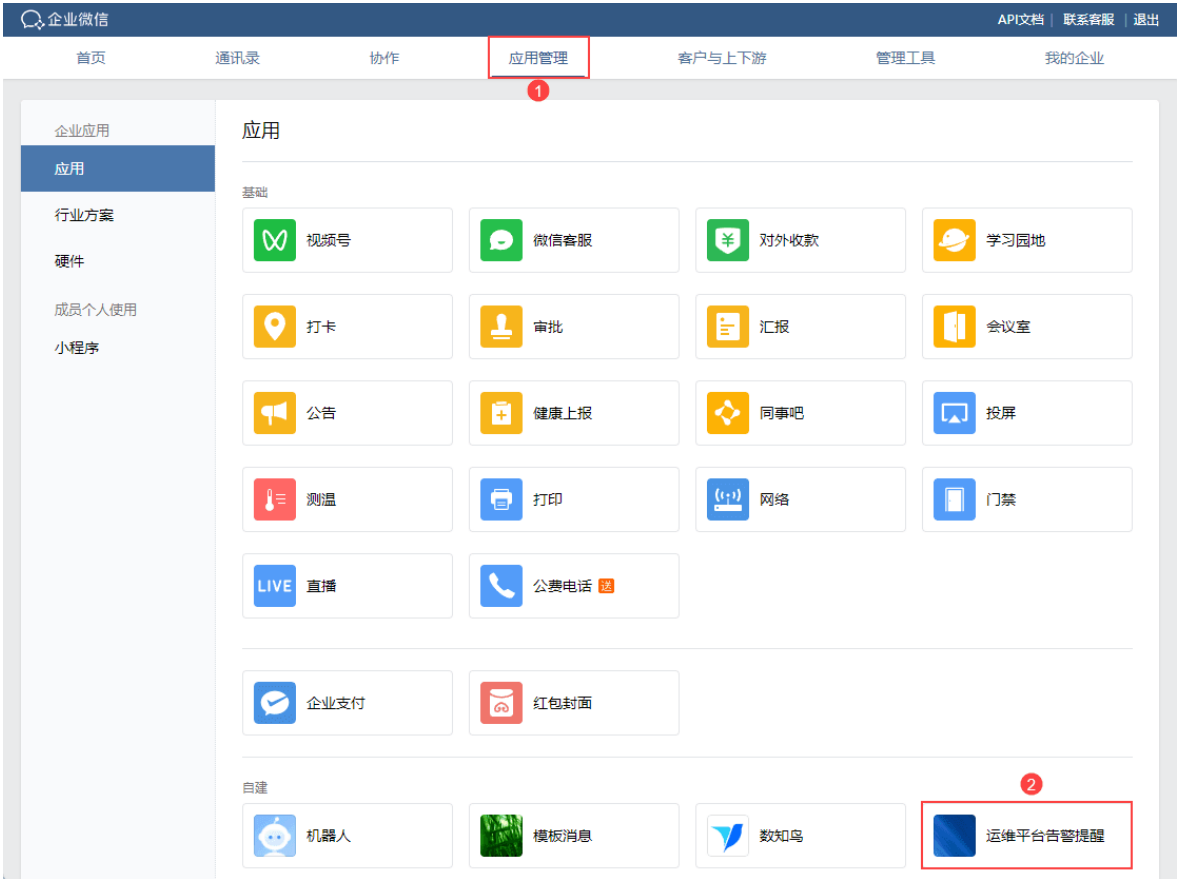
### 1) CorpID

企业微信管理员登录 [企业微信管理后台](#)，在「我的企业」Tab 下，点击「企业信息」，即可获取 CorpID。如下图所示：



## 2) 应用 AgentID 和应用 Secret

企业微信管理员登录 [企业微信管理后台](#)，在「应用管理」Tab 下，点击刚刚新建的用于发送告警信息的应用，即可获取应用 AgentID 和应用 Secret。如下图所示：





### 3) 接收对象的部门或用户 ID

企业微信管理员登录 [企业微信管理后台](#)，在「通讯录」Tab 下。

- 左侧可选择部门，点击后方的按钮，即可查看部门 ID
- 点击单个用户的信息，即可在信息详情页查看用户 ID

请确保该部门/用户在上文设置的应用可见范围内。



### 4.3 配置告警对象

管理员登录「运维平台」，点击「监控与预警>告警配置>告警规则」，点击「添加告警对象」。

设置告警类型为「企业微信」，填写 4.2 节获取的相关微信参数信息，点击「确定」。

如下图所示：

注 1：接收对象仅支持设置单个用户/单个部门。请确保该部门/用户在上文设置的发送告警信息的应用可见范围内。

注 2：若部门 ID/用户 ID 为空，则告警信息将推送给应用可见范围内的全部用户。

注 3：配置完成后，请点击「发送测试邮件」，确保邮箱可使用。





## 5. 企业微信机器人告警

支持通过企业微信机器人发送告警通知到企业微信群。

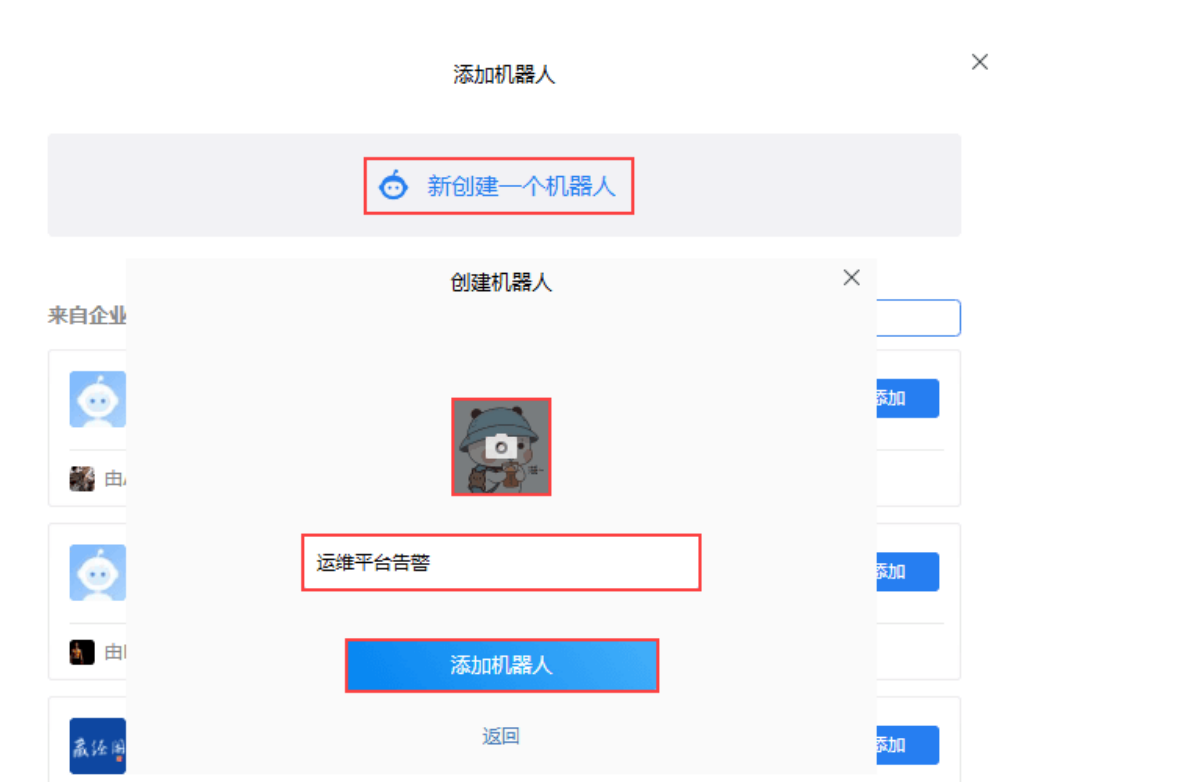
管理员需要先在企业微信群中创建并添加一个用于发送告警的机器人，并获得相关 Webhook 地址，方可进入运维平台配置告警对象。

### 5.1 微信群创建机器人

1) 打开需要推送消息的企业微信群中， 点击「设置>添加群机器人>添加机器人」。

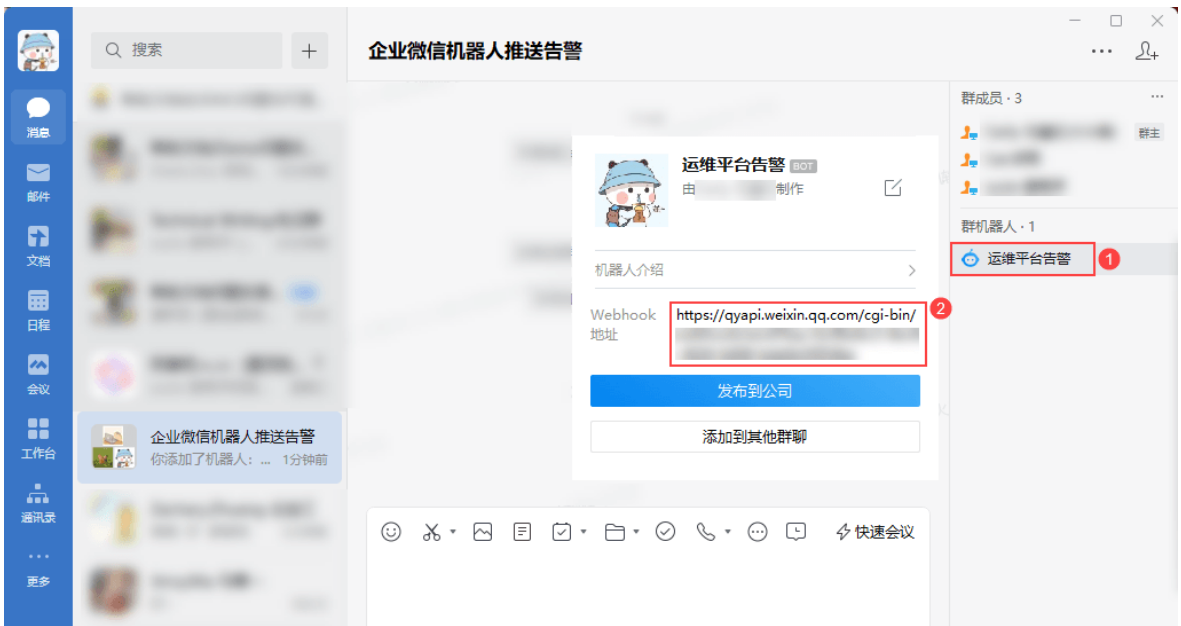


2) 点击「新创建一个机器人」，设置机器人头像和名称，点击「添加机器人」，即可将机器人添加到企业微信群中。



5.2 获取 Webhook 地址

在微信群中，点击查看机器人信息，即可获取 Webhook 地址。如下图所示：



5.3 配置告警对象

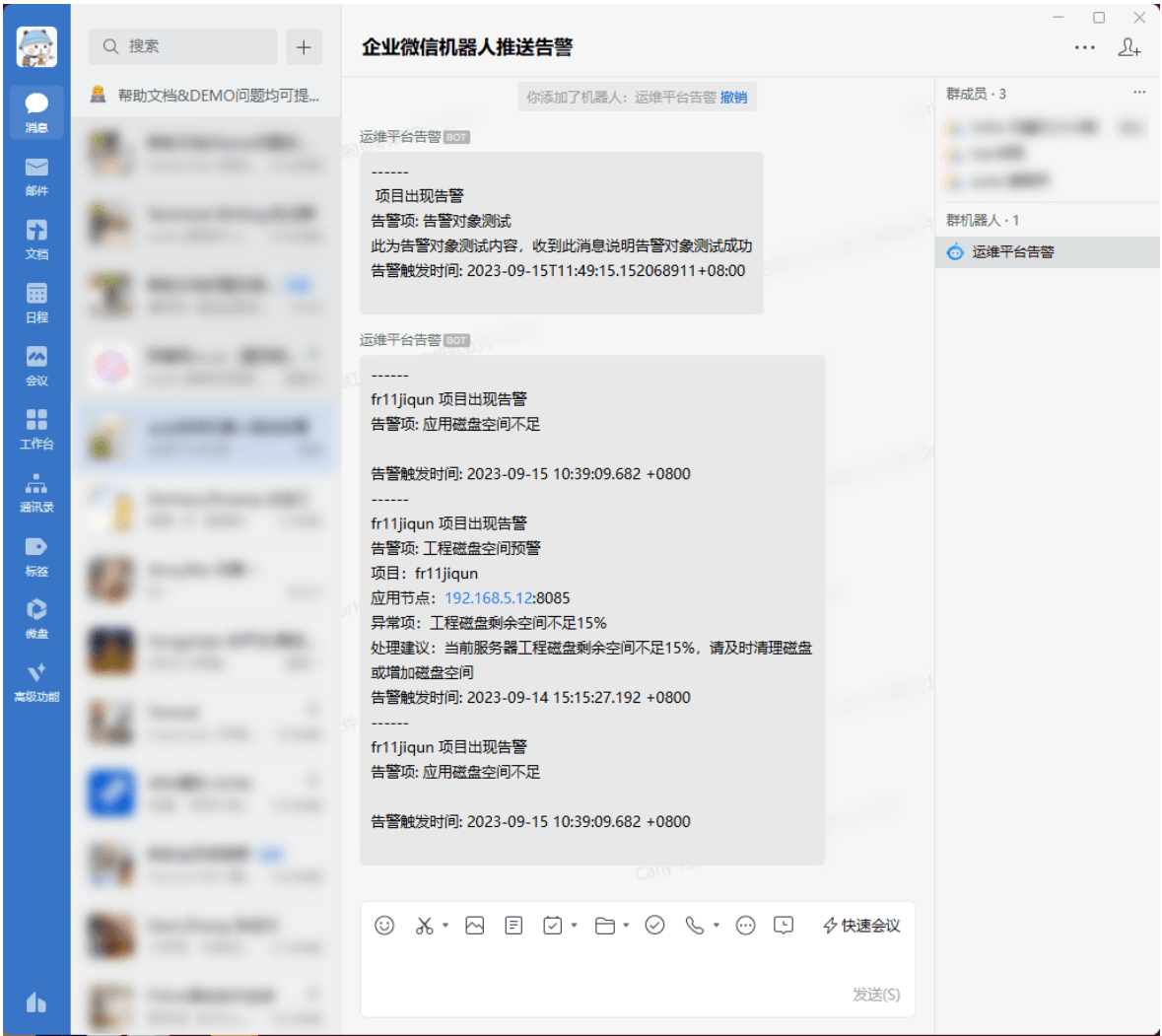
管理员登录「运维平台」，点击「监控与预警>告警管理>告警通知设置」，点击「添加告警对象」。

自定义告警对象名称, 设置告警类型为「企业微信机器人」, 填写 5.2 节获取的 Webhook 地址，点击「确定」。如下图所示：



### 5.4 效果预览

当 FineOps 监测到项目触发告警任务后，机器人会发送消息到所在的企业微信群中，如下图所示：



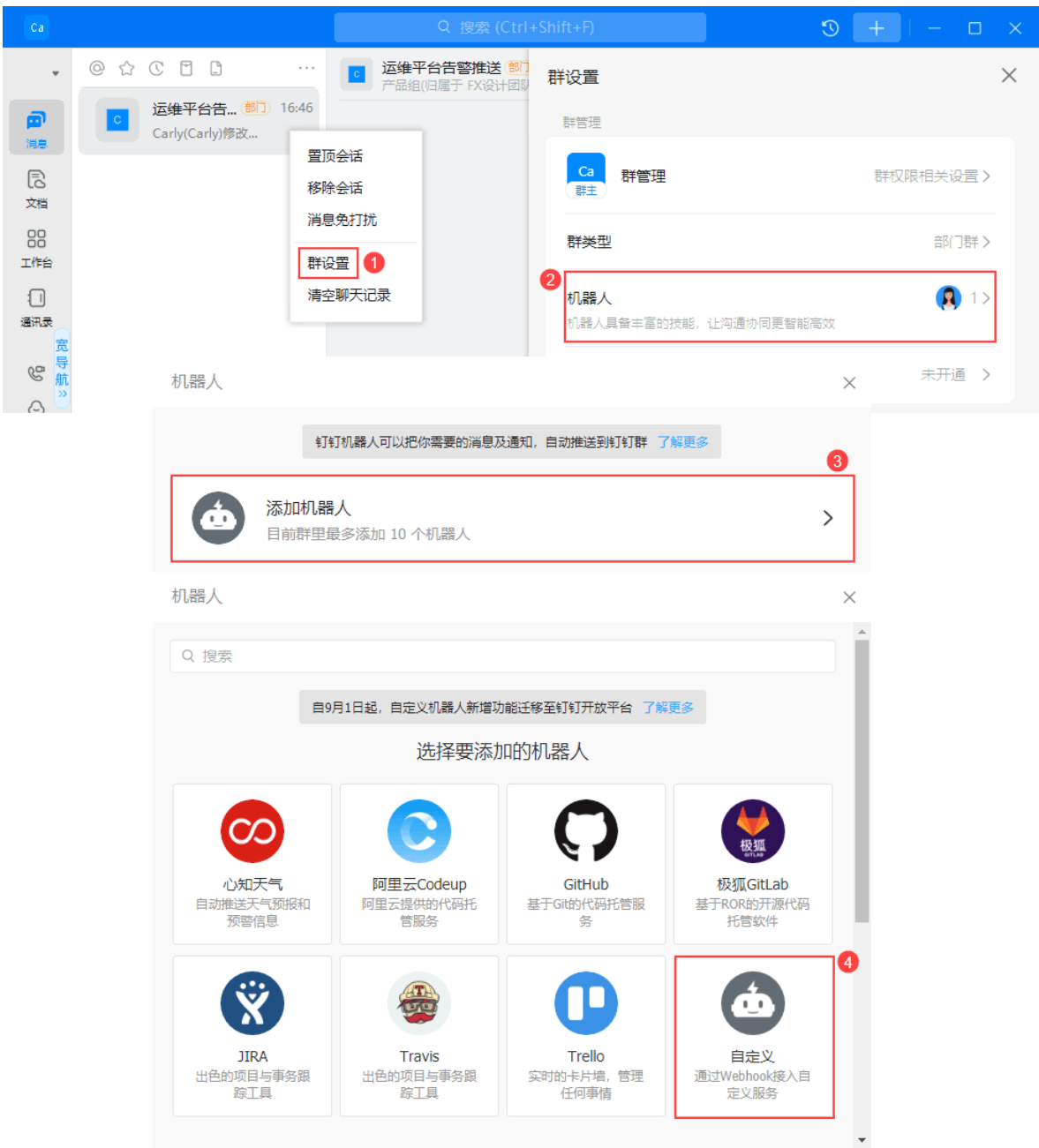
### 6. 钉钉机器人告警

支持通过钉钉机器人发送告警通知到钉钉群。

管理员需要先在钉钉群中创建并添加一个用于发送告警的机器人，并获得相关 Webhook 地址，方可进入运维平台配置告警对象。

### 6.1 钉钉群创建机器人

1) 打开需要推送消息的钉钉群中，点击「群设置>机器人>添加机器人>自定义」。



2) 点击「添加」，设置机器人信息，可开启安全设置中的「加签」，即密钥功能。点击「完成」，即可将机器人添加到钉钉群中。

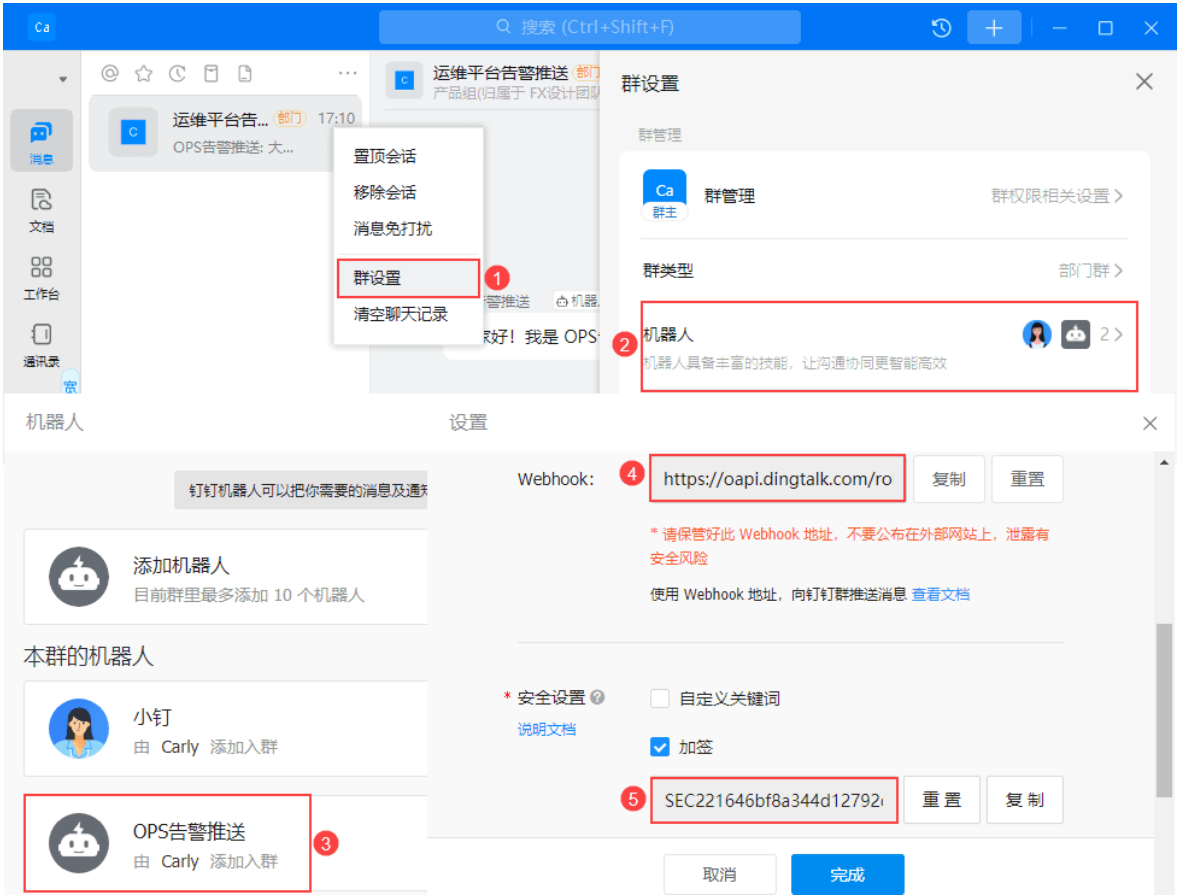
注：安全设置中，可以自行选择是否设置「加签」。

如使用「自定义关键词」，若告警信息中不包含该关键词，会导致消息接收不到。



### 6.2 获取 webhook 地址

在钉钉群中，点击「群设置>机器人」，查看机器人信息，即可获取 Webhook 地址和加签信息。如下图所示：



### 6.3 配置告警对象

管理员登录「运维平台」，点击「监控与预警>告警管理>告警通知设置」，点击「添加告警对象」。

自定义告警对象名称，设置告警类型为「钉钉机器人」，填写 6.2 节获取的 Webhook 地址和加签信息（如未设置加签，可不填写），点击「确定」。如下图所示：



## 7. 飞书机器人

支持通过飞书机器人发送告警通知到飞书群。

管理员需要先飞书群中创建并添加一个用于发送告警的机器人, 并获得相关 Webhook 地址, 方可进入运维平台配置告警对象。

### 7.1 飞书群创建机器人

1) 打开需要推送消息的飞书群中, 点击「设置>群机器人>添加机器人」。



2) 选择「自定义机器人」，设置机器人头像、名称和描述，点击「添加」。

添加机器人

Q 搜索

1自定义机器人  
通过webhook将自定义服务的消息推送至飞书

审批  
简单、高效、开放的审批工具

< 返回

×

2自定义机器人可以通过 webhook 向群聊推送来自外部服务的消息。请填写以下信息完成添加。[查看说明](#)

机器人头像\*

使用默认头像

上传头像

机器人名称\*

运维平台告警

6/32

描述\*

通过webhook将FineOps运维平台的告警消息推送至飞书

31/256

取消

3添加

3) 支持设置「签名校验」等安全设置，点击「完成」，即可将机器人添加到飞书群中。

< 返回

1×

webhook 地址

https://open.feishu.cn/open-apis/bot/v2/hook/

复制

请保管好此 webhook 地址。不要公布在 Github、博客等可公开查阅的网站上。地址泄露后可能被恶意调用发送垃圾信息

安全设置

自定义关键词 ⓘ

IP 白名单 ⓘ

签名校验 ⓘ

kj4wQIooo

重置 复制

密钥如上，签名方法详见 [帮助文档](#)

设置

仅群主和添加者可编辑、移除此机器人

2完成

7.2 获取 Webhook 地址

在飞书群中，点击「设置>群机器人」。查看机器人信息，即可获取 Webhook 地址和签名密钥。如下图所示：

588





7.3 配置告警对象

管理员登录「运维平台」，点击「监控与预警>告警管理>告警通知设置」，点击「添加

告警对象」。

自定义告警对象名称，设置告警类型为「飞书机器人」，填写 7.2 节获取的 Webhook 地址和签名密钥（如未设置签名校验，可不填写），点击「确定」。如下图所示：



# 7. 业务工具

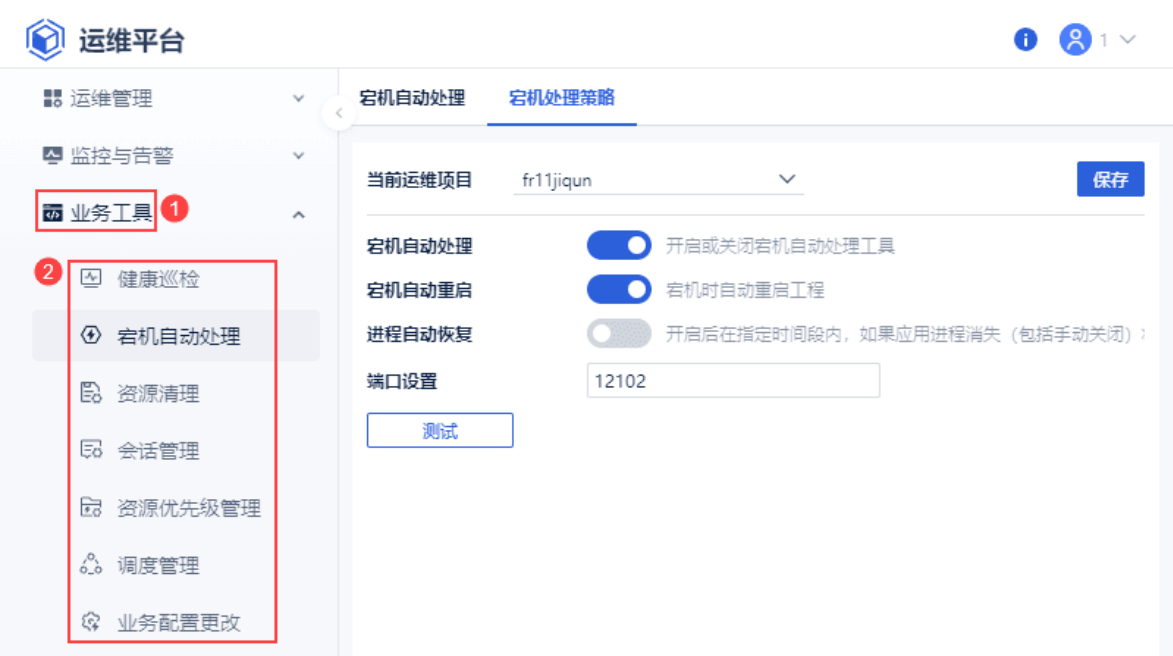
## 1. 概述

### 1.1 版本

运维平台版本	功能变更
V1.5.5	-

### 1.2 功能简介

运维平台提供定期、全面的业务检查功能，帮助用户基于实际业务需求调整工程配置。



## 2. 索引

功能	说明	帮助文档
健康巡检	管理员需要对应用进行定期健康巡检，确保应用所在的环境及应用内配置合理，以保证应用的正常运行。	
	巡检内容包括端口检查、网络服务检查、存储服务检查、环境配置检查以及业务配置检查。	<a href="#">健康巡检</a>
宕机自动处理	运维平台提供「宕机自动处理」功能。发生宕机后，通过分析生成的 DUMP 文件，可以快速定位宕机原因，及时处理宕机问题和做一些有效的预防措施，为用户增加一道保险。	<a href="#">宕机自动处理</a>
宕机自动处理	运维平台提供「宕机处理策略」功能，可为项目绑定的 FineBI/FineRep 等工程自动生成 DUMP 文件、同时可重新启动工程。发生宕机后，通过分析生成的 DUMP 文件，可以快速定位宕机原因，及时处理宕机问题和做一些有效的预防措施，为用户增加一道保险。	<a href="#">宕机处理策略</a>

	1) 提供磁盘清理功能，高效推荐可清理文件。	
资源清理	2) 部分文件可设置自动清理逻辑。	<a href="#">资源清理</a>
	3) 能够解决可高效清理磁盘节省资源和预防磁盘宕机的问题。	
	用户在帆软应用中，打开一张目录中的模板，即为一个会话。	
会话管理	管理员可通过「会话管理」查看应用中会话占用情况，并支持对于异常会话手动进行中止。	<a href="#">会话管理</a>
资源优先级管理	一个系统中存在很多模板，但是业务的优先级有先有后。部分重要业务的资源需要优先保证。	<a href="#">资源优先级管理</a>
调度管理	<p>在 FineReport、FineBI 中，存在各种各样的后台调度任务，例如 BI 表更新、定时调度任务、系统任务等。</p> <p>调度任务堆积在同一时间段执行，系统压力大，甚至导致调度失败。</p> <p>运维平台提供「调度管理」功能，帮助用户了解系统中的调度任务和系统压力，促进任务配置优化。</p>	<a href="#">调度管理</a>
业务配置更改	<p>帆软产品中，有些配置信息记录在 FineDB 的 fine_conf_entity 表中，只能通过修改表字段来进行更改。</p> <p>用户可通过「配置管理」方式安全地修改相关配置。</p>	<a href="#">业务配置更改</a>

# 7.1 健康巡检

## 1. 概述

### 1.1 版本

运维平台版本	功能变更
V1.5.5	-

### 1.2 功能简介

管理员需要对应用进行定期健康巡检，确保应用所在的环境及应用内配置合理，以保证应用的正常运行。

巡检内容包括端口检查、网络服务检查、存储服务检查、环境配置检查以及业务配置检查。

检查项目	原因
系统检查	包括：CPU 主频、核心数、内存大小、已使用内存、空闲内存、交换分区、检测时间、根目录磁盘、根目录磁盘剩余
系统运行状态	包括：工程启动路径、工程下文件夹大小异常情况、xms 配置、插件信息、注册信息、web.xml 检查、容器信息、工程 JAR 包信息、服务器时间
端口检查	帆软应用正常使用过程中，需要服务器开放一些端口，因此需要对这些端口进行检查是否开放，避免出现功能异常
网络检查	检测应用各组件通信是否能够 ping 通，是否存在丢包情况
服务连通性检查	检测服务是否连通
业务配置检查	包括：finedb、日志级别、日志清理、自动备份磁盘占用限制、自动备份保留数量限制、定时调度任务清理配置
环境检查	包括：overcommit_memory、overcommit_ratio、codecache 配置、语言配置-Duser.country、语言配置-Duser.language、zip 相关配置-Dsun.zip.disableMemoryMapping、gblic 内存泄露相关配置
磁盘检查	1) 关注磁盘的性能和使用情况，防止影响业务的正常运行 2) 关注磁盘剩余空间，防止出现磁盘空间不足导致系统运行受影响
虚拟机检查	系统当前内存、headless 模式
非容器化部署检查	包括：进程用户、服务器主要文件夹权限
配置检查	包括：外接数据库编码、Nginx 健康检查、Nginx 参数配置、集群参数配置、更新资源无限制时间段配置、缓存开启情况、SQL 执行时长限制、单模板单元格数限制、导入 excel 单元格数限制、单数据集行数限制、负载预警开启、脏数据检测

## 2. 使用前提

使用「健康巡检」功能前，必须确保需要被诊断的「目标项目」状态可用，否则提示「项目状态异常，请恢复项目后使用或查看其他项目」。

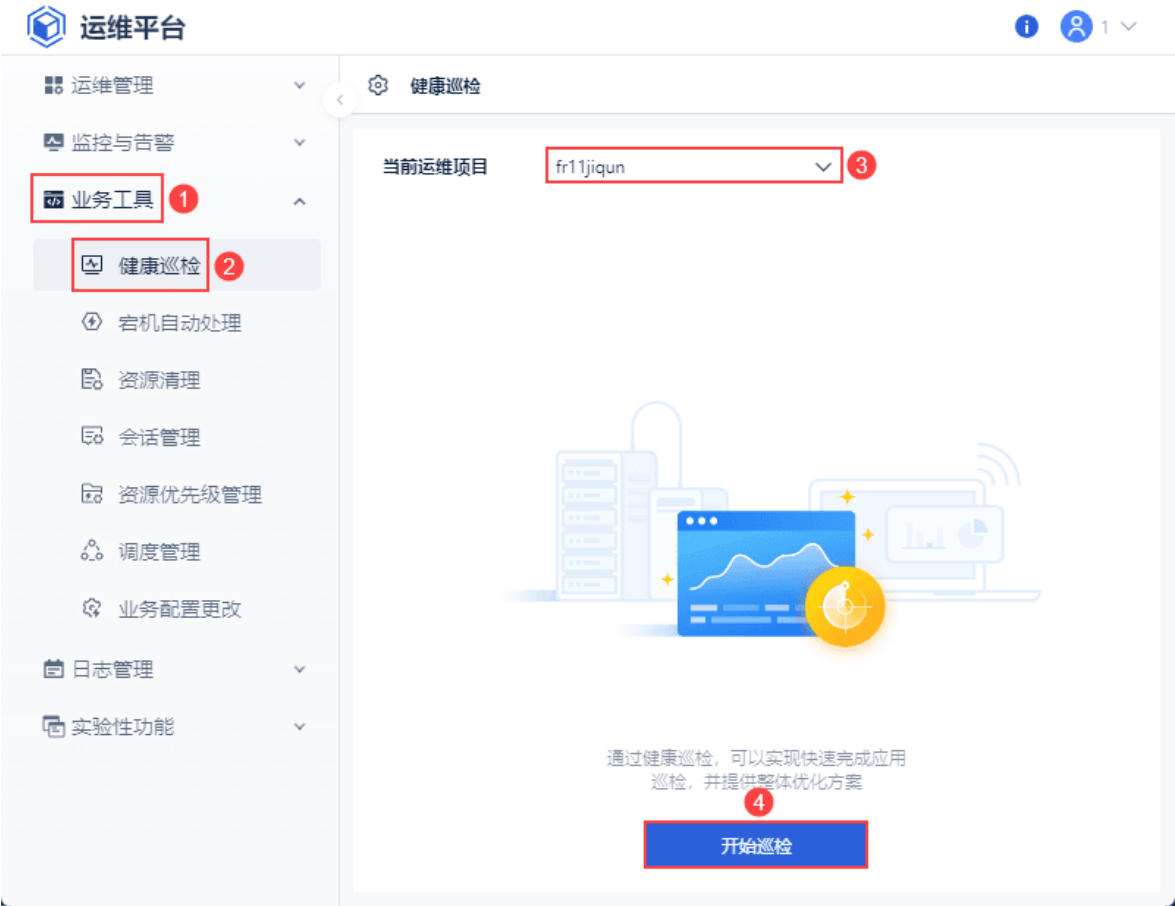


### 3. 诊断异常项

健康巡检支持手动检查和自动定期检查。

#### 3.1 手动检查

管理员登录运维平台，点击「业务工具>健康巡检」，选择需要进行健康诊断的运维应用，点击「开始诊断」按钮，即可开始诊断。如下图所示：



运维平台会对应用进行端口检查、网络服务检查、存储服务检查、环境配置检查以及业务配置检查。如果存在异常将显示在页面上。

进行修复/更改后，管理员可再次点击「重新诊断」按钮，「运维平台」会重新对目标应用进行诊断，输出最新结果。

注：检查项具体内容请参考：[健康巡检内容](#)。



3.2 自动检查

健康诊断支持自动触发检查。

管理员登录运维平台， 点击「业务工具>健康巡检>全局设置」， 支持设置系统自动检查。

默认为每周日上午 11 点进行检测， 用户也可自行配置自动检查的时间。如下图所示：

检测内容和检测结果显示， 与手动检查完全一致。



## 4. 修复异常项

### 4.1 快速修复

部分异常配置支持快速修复。

管理员点击「快速修复」按钮，勾选需要配置的检查项，点击「应用配置」，即可一键配置异常项的值。

配置成功后，跳出提示「应用配置成功，重启服务器后生效」。应用工程重启后，可以发现刚刚的异常项均已成功配置。

注 1：若没有文件读写权限或者无法修改配置文件时，提示「应用配置失败：没有配置文件读写权限」，点击「确定」，则修改失败。

注 2：若没有可快速修复的配置项，则该按钮灰化，不可点击。



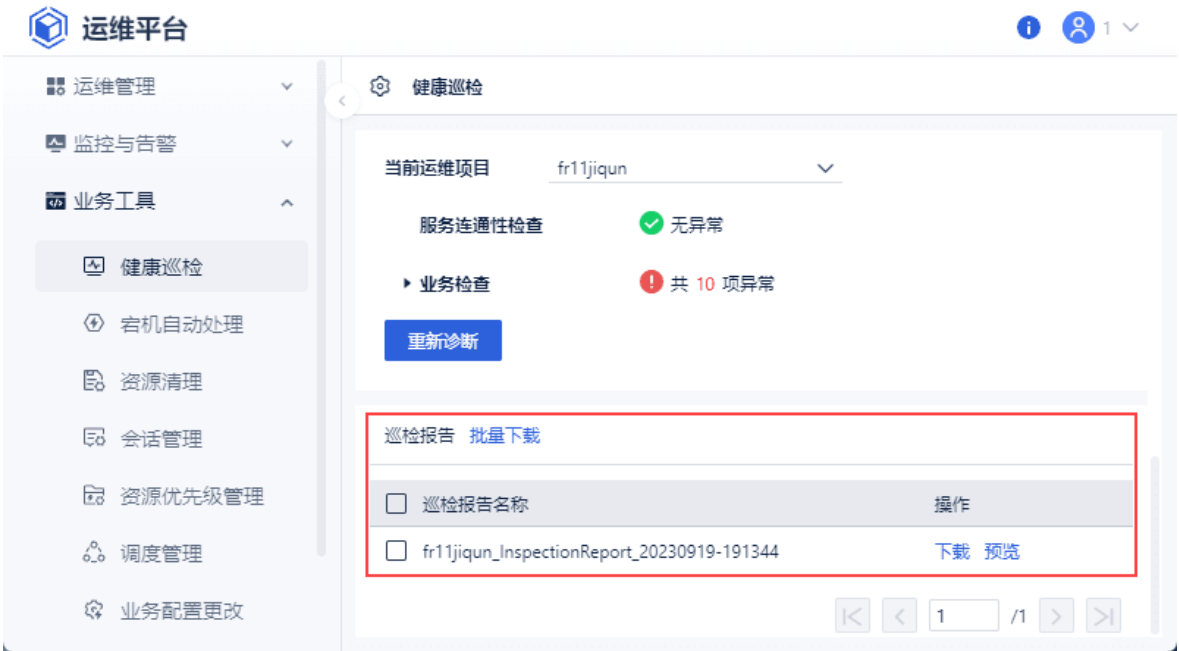


#### 4.2 手动修复

检查项具体设置建议请参考：[健康巡检内容](#)

#### 5. 巡检报告

对于健康诊断的结果，生成巡检报告，支持下载/预览/批量下载。



巡检报告中默认仅展示存在问题的检查项，支持切换查看不同节点的全部检测项。如下图所示：

系统 2023-09-19 19:13:44 健康巡检报告

异常检测项

全部检测项

支持切换检测项

支持切换节点

共发现异常项目 14 项

[如何修复异常?](#)

节点选择: 192.168.101.82

系统检查

共 1 项异常

序号	检查项	状态	当前配置	修改建议
0107	cpu 主频		1.8955Ghz	建议主频大于等于 2.4 gh

系统运行状态

无异常

端口检查

无异常

网络检查

无异常

磁盘检查

共 1 项异常

序号	检查项	状态	当前配置	修改建议
0506	磁盘空间		/opt/Katherine/fr11/webapps/w...	建议磁盘总大小高于100GB

非容器化部署信息

无异常

环境检查

共 1 项异常

序号	检查项	状态	当前配置	修改建议
0702	最大文件打开数		4096	建议open_files参数配置不低于6...

## 7.2 健康巡检内容

### 概述

### 版本

运维平台版本	功能变更
V1.5.5	- <ul style="list-style-type: none"><li>1) 系统检查新增检测项：<ul style="list-style-type: none"><li>0113 CPU 信息</li><li>0114 服务器字体编码</li><li>0115 服务器字体包</li></ul></li><li>2) 业务检查新增检测项：</li></ul>
V1.5.6	<ul style="list-style-type: none"><li>1025 是否开启云端运维</li></ul> <ul style="list-style-type: none"><li>3) 非容器化部署信息新增检测项：<ul style="list-style-type: none"><li>0603 多进程检测</li></ul></li><li>4) Java 虚拟机检查新增检测项<ul style="list-style-type: none"><li>0822 jdk 位置</li></ul></li></ul> <ul style="list-style-type: none"><li>1) 磁盘检查新增检测项：<ul style="list-style-type: none"><li>0509 顺序读取(非缓冲池)</li><li>0510 顺序写入(非缓冲池)</li></ul></li></ul>
V1.5.9	<ul style="list-style-type: none"><li>2) 磁盘检查删除检测项：<ul style="list-style-type: none"><li>0503 随机读</li></ul></li><li>3) 磁盘读写速度若低于 100，则不显示具体数值，仅提示「速率小于 1</li></ul>

00MB/S」

1) 业务检查新增检测项:

V1.8.0

- 1026 密码强度限制检测

1) 环境检查部分检查项支持快速修复

- 0705 gblic 内存泄露相关配置

2) Java 虚拟机检查部分检查项支持快速修复

V1.9.0

- 0807 headless 模式
- 0809 BytecodeRecompilationCutoff 配置
- 0810 PerMethodRecompilationCutoff 配置
- 0813 codecache 配置
- 0816 zip 相关配置-Dsun.zip.disableMemoryMapping

3) Java 虚拟机检查新增检测项:

- 0823 元数据空间配置

系统运行状态新增检测项:

V1.10.0

- 0212 tomcat 后台运行

功能简介

帆软应用对系统有一定的配置要求，如果配置不符合要求可能出现使用异常、宕机等风险，因此需要定期对环境及应用的配置情况进行检测，以确保应用的健康运行。同时管理员需要关注应用的运行状况，及时通过运维手段规避风险或在问题发生后定位并进行排除，实现应用的持续高可用。

系统检查

系统检查的检测项包括:

是否支持				
巡检项	检测项	集群节点是否分别检查	支持快速修复	健康标准及修改建议
编号				
0101	操作系统信息	是	否	仅输出信息
0102	检测时间	是	否	仅输出信息
0103	时区信息	是	否	仅输出信息
0104	glibc/gcc 版本	是	否	仅输出信息
0105	hostname	是	否	仅输出信息
0106	host 文件信息	是	否	仅输出信息
0107	CPU 主频	是	否	检查主频大于等于 2.4 gh 建议主频大于等于 2.4 gh
0108	核心数	是	否	检查核心数大于 4 核 cpu 核心数推荐大于 4
0109	内存大小	是	否	检查内存是否大于等于 15.5GB 建议物理内存大于等于 16G
0110	交换分区	是	否	检查是否关闭 SWAP 配置 输出交换分区大小 建议关闭 SWAP 配置
0111	已使用内存	是	否	仅输出信息
0112	空闲内存	是	否	仅输出信息
0113	CPU 信息	是	否	仅输出信息
0114	服务器字体编码	是	否	仅输出信息
0115	服务器字体包	是	否	仅输出信息

## 0110 关闭 SWAP 分区

### 1) 登录宿主机

使用合适的身份验证方式（例如 SSH、Console 等），登录到运行容器的宿主机。

### 2) 查看当前系统中是否启用了 swap 分区

使用以下命令查看系统中已经启动的 swap 分区信息：

```
sudo swapon -s
```

如果输出结果为空，则表示当前没有启动 swap 分区。

3) 关闭 swap 分区

使用以下命令关闭 swap 分区：

```
sudo swapoff -a
```

该命令将关闭所有已经启动的 swap 分区。

4) 永久禁用 swap 分区

为了避免系统在下一次启动时自动启动 swap 分区，需要在系统配置文件中进行相应的修改。

以 Ubuntu 为例，可以按照以下步骤操作：

- 修改/etc/fstab 文件，找到含有 swap 的行并注释掉或删除该行。
- 执行以下命令更新 grub 引导程序：

```
sudo update-grub
```

5) 验证 swap 分区是否已经关闭

重启系统后，可以再次执行 sudo swapon -s 命令查看系统中是否存在已经启动的 swap 分区。

如果输出结果为空，则表示 swap 分区已经成功关闭。

系统运行状态

系统运行状态的检测项包括：

		集群节点	是否	健康标准
巡检项编号	检测项	是	支持	
		否	快速	
		分别检	修复	
		查		
0201	启动时长	是	否	仅输出信息 包括：本次启动的耗时
0203	工程启动路径	是	否	仅输出信息 包括：目录路径
				注：仅检测本地，不检测文件服务器。
0204	工程下文件夹大小异常情况	是	否	检查工程 WEB-INF 下一层的每个文件夹大小  健康：不存在大于 5G 的文件夹

			不健康：存在大于 5G 的文件夹
			修改建议：
			建议检查确认对应文件夹的内容并进行清理
			提示出超过 5G 大小的文件夹路径
0205	插件信息	否否	仅输出信息 包括：插件名、插件版本、插件是否启用
0206	注册授权信息	否否	仅输出信息 包括：工程是否注册
0207	WEB-INF 下是否有 web.xml	否否	仅输出信息 包括：是/否
0208	容器版本	是否	仅输出信息
0209	工程 jar 包信息	否否	仅输出信息 包括：工程各个模块的版本和 JAR 包日期
0211	临时输出目录	是否	仅输出信息 包括：目录路径、目录剩余可用空间
			对于 Windows 系统、Tomcat 容器下的工程，
			检测是否使用 javaw 后台运行 tomcat
0212	tomcat 后台运行	是	健康：使用 javaw 后台运行 tomcat  不健康：未使用 javaw 后台运行 tomcat  修改建议：建议使用 javaw 后台运行 tomcat

0204 清理文件夹

1) WEB-INF 文件夹存放着什么

- FineReport: [FineReport 安装目录结构](#)
- FineBI: [FineBI 安装目录结构](#)

2) 哪些文件可清理

文件夹/文件	用途	自动清理	手动清理
--------	----	------	------

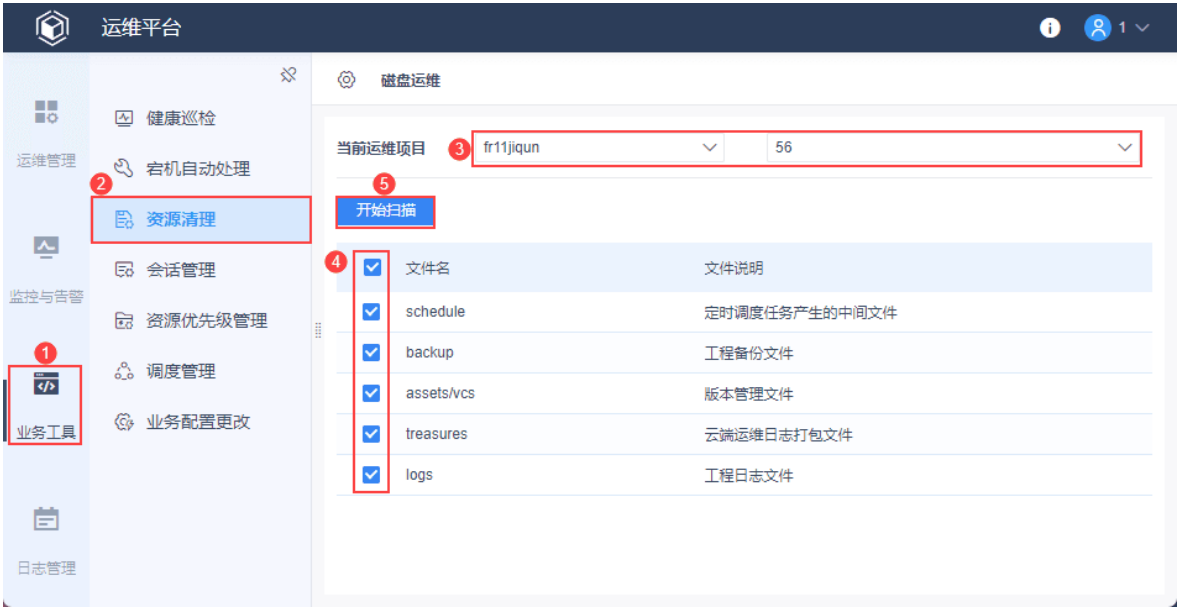
schedule	1) 文件路径： %BI_HOME%\webapps\webroot\WEB-INF\schedule	文件层级直接展示到具
	2) 文件用途： 存储定时调度任务产生的中间文件	体的日期 (日级别), 用户手动选择需要清理的范围即可
	3) 可清理范围： 保留最近一年的文件，其余均可清理	不支持自动清理 推荐清理 12 个月之外的文件
	4) 清理级别： 建议清理	12 个月之内的文件不显示
treasure	1) 文件路径： %BI_HOME%\webapps\webroot\WEB-INF\treasures	文件层级直接展示到具
	2) 文件用途： 云端运维日志打包文件	体的日期 (日级别), 用户手动选择需要清理的范围即可
	3) 清理级别： 建议清理	保留最近 12 个月的文件, 其余文件清理 推荐清理 12 个月之外的文件
		12 个月之内的文件不显示

3) 清理方式

管理员登录运维平台，点击「业务工具>资源清理」。

选择需要扫描的文件类型，点击「开始扫描」，立即开始扫描可以手动清理的冗余文件。





## 0212 tomcat 后台运行

### 原因分析：

Windows 系统中，tomcat 容器下部署的工程，如果不使用后台模式启动，而是使用窗口模式启动，可能会因为误点窗口，导致窗口进入编辑插入状态，阻塞日志输出，从而导致工程卡死。

因此建议 Windows 系统中，tomcat 容器下部署的工程，tomcat 使用后台模式启动。

### 操作步骤：

先修改默认启动模式为 javaw。为了防止重复启动，再添加脚本检测启动命令，按任意键即可结束脚本。

#### 1) 修改 setclasspath.bat

打开%Tomcat\_HOME%\bin\setclasspath.bat，找到并修改 set \_RUNJAVA 的值：

```
set _RUNJAVA="%JRE_HOME%\bin\javaw.exe"
```

#### 2) 修改 catalina.bat

打开%Tomcat\_HOME%\bin\catalina.bat，在最后加上判断提示：

```
set ENV_VAR_VALUE=%_EXECJAVA%
echo %ENV_VAR_VALUE% | findstr /C:"javaw" >nul
if %errorlevel% equ 0 (
    echo tomcat is running in the background
    pause
)
```

### 端口检查

端口检查的检测项包括：

巡检项	检测项	集群节点	是否支	健康标准及修改建议
-----	-----	------	-----	-----------

编号		是	持	
		否	分别检	快速修
		查	复	
0301	websocket 端口	否	否	仅输出信息 包括：端口号  检测集群端口是否畅通  健康：端口间畅通
0302	集群端口	是	否	不健康：端口存在不畅通  修改建议：集群端口连通性状态异常  检测：  1) 宕机自动检测是否开启
0303	宕机检测端口	是	否	2) 宕机端口是否畅通  优化建议：建议开启宕机检测并保证端口可用

0301 websocket 端口

WebSocket 主要用于刷新 token、用户被踢出、平台消息、内存和 CPU 显示、平台日志处当前系统在线人数、数据连接编辑状态的确定。

WebSocket 不开放影响 socket 通信，访问可能异常断开。

用户可根据自己的工程情况选择合适的 WebSocket 端口配置方法，详情请参见下表：

方案	场景	配置方法
容器 Websoc ket 方案	容器 Websocket 方案，无需任何用户操作，系统可自动使用 Web 容器自带的 WebSocket 进行连接，端口复用 http 端口	
	• 只需用户环境满足生效条件，则自动使用该方案	<a href="#">容器 Websocket 方案</a>
	• 若用户环境不满足生效条件，则可考虑自行配置 socket.io 方案	
socket.io 方案	单机环境配置 WebSocket 端口	<a href="#">单机配置 WebSocket 端口</a>

- 集群环境配置 WebSocket 端口

HTTPS 环境配置 Websocket 端口

仅对外开放一个端口
- [集群配置 WebSocket 端口](#)

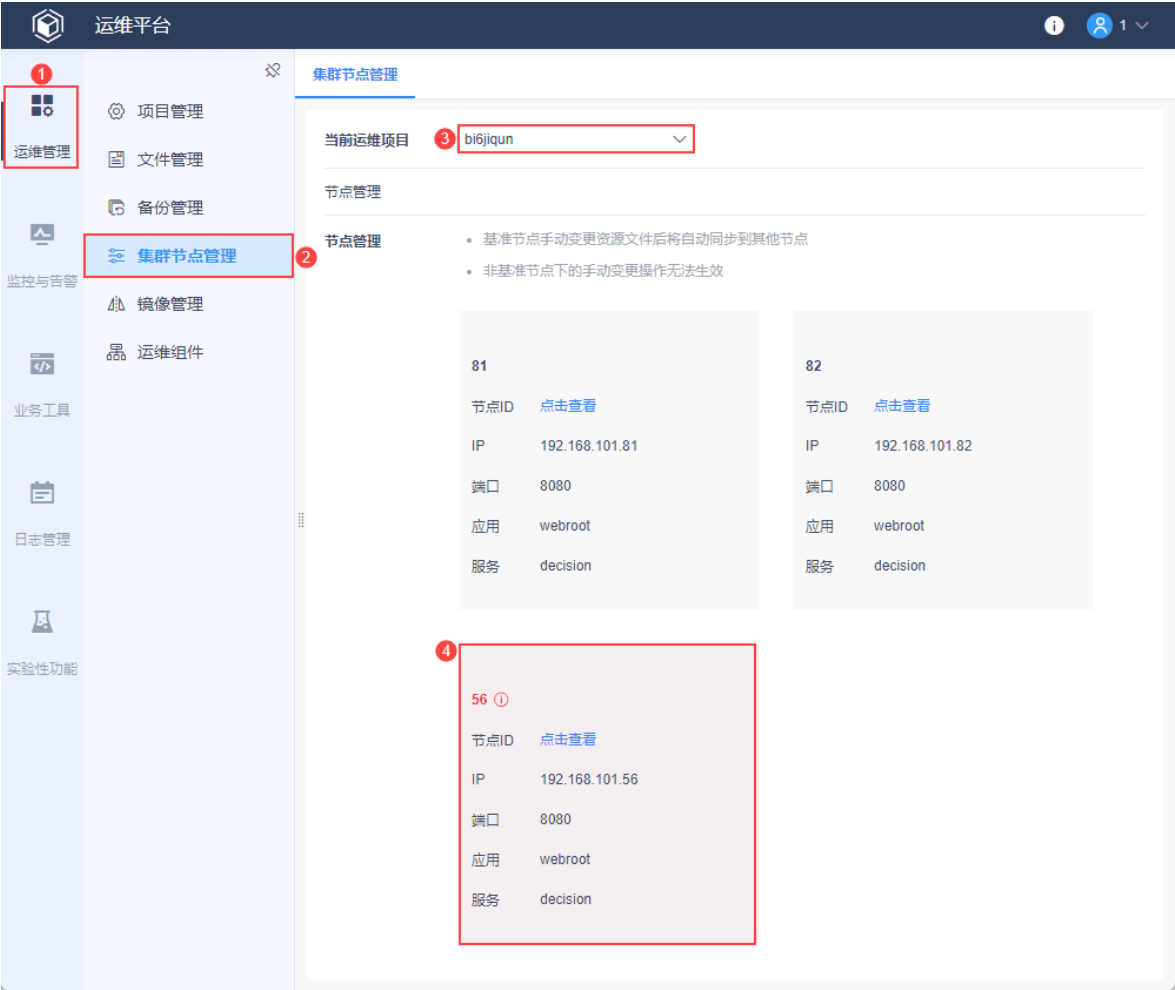
[HTTPS 环境配置 WebSocket](#)

[不额外给 WebSocket 对外开放端口](#)

0302 集群端口

管理员登录运维平台， 点击「运维管理>集群节点管理」， 选择运维项目。

- 可查看集群各个节点的端口， 无法使用的节点飘红。
- 可编辑节点端口， 鼠标悬浮即可出现编辑按钮。



0303 宕机检测端口

管理员登录运维平台， 点击「业务工具>宕机自动处理>宕机处理策略」， 选择运维项目。

- 可查看宕机自动处理工具使用的端口。默认端口为 12100。
- 可修改端口， 点击「测试」， 测试端口可用性， 点击「保存」即可。



网络检查

检查文件服务器、外接数据库、状态服务器和应用之间通信是否正常，是否能够 ping 通，是否存在丢包情况。

磁盘检查

磁盘检查的检测项包括：

健康 巡检 点检 项编 号	集群	是否	健康标准及修改建议
	节点	是否	
	分支	是否	
	检测项	是否	
	分支	是否	
健康 巡检 点检 项编 号	集群	是否	健康标准及修改建议
	节点	是否	
	分支	是否	
	检测项	是否	
	分支	是否	
健康 巡检 点检 项编 号	集群	是否	健康标准及修改建议
	节点	是否	
	分支	是否	
	检测项	是否	
	分支	是否	

仅输出信息

顺序 0501 读取	顺序	是否	包括：顺序读取的速度（若低于 100，则不显示具体数值，仅提示「速率
	是否	是否	小于 100MB/S」)
	注：	注：	windows 系统下若权限不足，请手动使用命令检测「winsat disk -dr
	ive	ive	盘符 -seq -read」
	顺序	是否	仅输出信息
顺序 0502 写	顺序	是否	仅输出信息
	是否	是否	仅输出信息
	注：	注：	windows 系统下若权限不足，请手动使用命令检测「winsat disk -dr
	ive	ive	盘符 -seq -read」
	顺序	是否	仅输出信息

包括：顺序写的速度（若低于 100，则不显示具体数值，仅提示「速率小于 100MB/S」)

注：windows 系统下若权限不足，请手动使用命令检测「winsat disk -drive 盘符 -seq -write」

小文件创建小文件删除

05040505

是是否是否是否

仅输出信息  
包括：每秒创建小文件个数  
仅输出信息  
包括：每秒删除小文件个数

1) FineReport:

健康：磁盘总大小≥100GB

不健康：磁盘总大小 < 100GB

修改建议：建议磁盘总大小高于 100GB

磁盘空间

0506

是是否

2) FineBI:

健康：磁盘总大小≥400GB

不健康：磁盘总大小 < 400GB

修改建议：建议磁盘总大小高于 400GB

已使用磁盘空间

0507

是是否

仅输出信息  
包括：已使用的磁盘空间大小

1) FineReport:

健康：磁盘余量高于 10GB

不健康：磁盘余量不足 10GB

修改建议：磁盘余量不足 10GB，建议清理磁盘大文件避免影响使用

2) FineBI:

健康：磁盘余量高于 50GB

空闲磁盘空间

0508

是是否

不健康：磁盘余量不足 50GB			
修改建议：磁盘余量不足 50GB，建议清理磁盘大文件避免影响使用			
顺序 读取	仅输出信息		
	05 是否	包括：非缓冲池顺序读取的速度（若低于 100，则不显示具体数值，仅提示「速率小于 100MB/S」）	
顺序 写入	仅输出信息		
	05 是否	包括：非缓冲池顺序写入的速度（若低于 100，则不显示具体数值，仅提示「速率小于 100MB/S」）	

非容器化部署信息

非容器化部署的检测项包括：

巡检项编号	检测项	集群节点 是否支持		健康标准
		是	否	
0601	进程用户	否		仅输出信息
		是	否	包括：用户名称
0602	服务器主要文件夹权限	否		仅输出信息
		是	否	包括：缓存文件夹、resources、assets、assist、finedb 的权限
0603	多进程检测	否		注：文件服务器权限不输出。
		是	否	健康：帆软应用所需最大内存+其他进程使用内存 < 总物理内存 不健康：帆软应用所需最大内存+其他进程使用内存 ≥总物理内存

修改建议：建议帆软工程不要与其他应用服务共用服务器，否则会有宕机风险

环境检查

环境检查的检测项包括：

巡检 项 编号	检测项	集群节点	是否支持	健康标准
		是	否	
		否分别检 查	快速修 复	
0701	vm.max_map_count	是	否	健康： vm.max_map_count≥262144
				不健康： vm.max_map_count<262144
				修改建议： 建议配置 vm.max_map_count 参数为 262144
0702	最大文件打开数	是	否	1) FineReport: 健康： open_files≥65536 不健康： open_files<65536 修改建议： 建议 open_files 参数配置不 低于 65536
				2) FineBI: 健康： open_files≥131070 不健康： open_files<131070 修改建议： 建议 open_files 参数配置不 低于 131070
				健康： overcommit_memory = 0 不健康： overcommit_memory≠0 修改建议： 建议 vm.overcommit_memory 参数配置为 0
0704	overcommit_ratio	是	否	健康： overcommit_ratio = 50

				不健康: <code>overcommit_ratio</code> ≠50
				修改建议: 建议 <code>vm.overcommit_ratio</code>
				参数配置为 50
				检测是否配置了 <code>export MALLOC_ARENA_MAX</code>
				<code>ENA_MAX</code> 参数
0705	glibc 内存泄露相关配置	是	是	健康: <code>export MALLOC_ARENA_MAX=1</code>
				不健康: 未配置/配置其他值
				修改建议: 建议配置 <code>export MALLOC_ARENA_MAX=1</code>

0701 `vm.max_map_count`

1) 参数简介:

`vm.max_map_count` 是一个 Linux 系统内核参数，用于控制在一个进程中允许 `mmap` 映射内存区域的最大数量。

如果该参数的值太小，那么在使用一些应用程序时，可能会出现内存映射错误。

2) 建议配置:

建议配置 `vm.max_map_count` 参数为 262144

3) 修改方式:

- 在 `/etc/sysctl.conf` 文件中添加/修改参数值

`vm.max_map_count=262144`

- 运行以下命令使更改生效

`sysctl -p`

0702 最大文件打开数

1) 参数简介:

`open_files` 是指一个进程可以同时打开的文件描述符的最大数量。

文件描述符是 Unix 系统中用于访问文件（包括管道、套接字和其他文件操作）的唯一标识符。



## 2) 建议配置：

建议 FineReport 工程的 open\_files 参数配置不低于 65536

建议 FineBI 工程的 open\_files 参数配置不低于 131070

## 3) 修改方式：

- 修改参数配置

打开 Linux 的软硬件限制文件/etc/security/limits.conf，在文件尾部添加如下代码：

```
* soft nofile 65536
* hard nofile 65536
```

上述命令中，hard 表示系统的硬限制，soft 表示用户的软限制。

- 重启工程并重新登录

配置完成后，终端用户重新登录，并重启 Tomcat 工程，配置方可生效。

可使用 ulimit -n 命令，根据返回值确认是否修改成功。

## 0703 overcommit\_memory

### 1) 参数简介：

vm.overcommit\_memory 是一个 Linux 内核参数，它控制了系统是否允许进程分配超过可用内存的内存量。

当参数值为 0 时，内核将检查是否有足够的可用内存来满足提交的内存需求。如果有足够的空闲内存，则允许进程继续分配内存；否则分配请求将失败并返回错误。

## 2) 建议配置：

建议 vm.overcommit\_memory 参数配置为 0

## 3) 修改方式：

- 在/etc/sysctl.conf 文件中添加/修改参数值

```
vm.overcommit_memory=0
```

- 运行以下命令使更改生效

```
sysctl -p
```

## 0704 overcommit\_ratio

### 1) 参数简介:

vm.overcommit\_ratio 是一个 Linux 内核参数, 它指定了系统 VMware 超大页管理器 (Memory Ballooning)和 Swap 空间使用的比例。

当参数值为 50 时, 内核使用适当的 Swap 两倍大小作为另一个备用空间, 用于模拟物理内存。

### 2) 建议配置:

建议 vm.overcommit\_ratio 参数配置为 50

### 3) 修改方式:

- 在/etc/sysctl.conf 文件中添加/修改参数值

```
vm.overcommit_ratio=50
```

- 运行以下命令使更改生效

```
sysctl -p
```

## 0705 glibc 内存泄露相关配置

### 1) 参数简介:

为了防止 glibc 中的内存泄漏, 可以通过在环境变量中设置 MALLOC\_ARENA\_MAX 来限制 glibc 在分配和释放动态内存时使用的堆空间数量。

### 2) 建议配置:

建议配置 export MALLOC\_ARENA\_MAX=1

### 3) 修改方式:

- 关闭 Tomcat 服务器。
- 进入%Tomcat\_HOME%/bin 目录, 打开 startup.sh 文件。
- 添加/修改行: export MALLOC\_ARENA\_MAX=1
- 保存对 startup.sh 文件的修改并关闭。
- 在终端中运行 startup.sh 文件, 重启 Tomcat 并应用新的环境变量设置。

## Java 虚拟机检查

Java 虚拟机的检测项包括:

巡检  项  编号	检测项	集群节	是否	健康标准
		点是	支持	
		否分别	快速	
		检查	修复	
				健康：
				FR：堆外 NIO 内存为 2GB
				BI：堆外 NIO 内存为 4GB
0801	堆外 NIO 内存	是	是	不健康：未配置该参数，或堆外 NIO 内存配置了但不符合要求  修改建议：建议配置 JVM NIO 占用堆外内存 2GB  健康：堆外 fineIO 读内存为 2GB 或未配置该参数
0802	堆外 fineIO 读内存	是	是	不健康：堆外 fineIO 读内存配置了且不是 2GB  修改建议：建议 JVM fineIO 读内存使用堆外内存设置为 2G  健康：堆外 fineIO 写内存为 1GB 或未配置该参数
0803	堆外 fineIO 写内存	是	是	不健康：堆外 fineIO 写内存配置了且不是 1GB  修改建议：建议 JVM fineIO 写内存使用堆外内存设置为 1G  健康：垃圾回收器类型为 ParallelScavenge
0804	垃圾回收器类型	是	否	不健康：垃圾回收器类型不是 ParallelScavenge  修改建议：建议 JVM 使用垃圾回收器类型为 ParallelScavenge

			健康：不配置 DisableExplicitGC 参数
0805	DisableExplicitGC 参数	是	是 不健康：配置了 DisableExplicitGC 参数 修改建议：建议取消 DisableExplicitGC 参数配置
0806	JDK 版本	是	否 健康：高于帆软内置的 JAVA 版本（一般在 1.8.202 及以上版本） 不健康：低于帆软内置的 JAVA 版本 修改建议：建议使用 1.8.202 及以上版本 JDK
0807	headless 模式	是	是 健康：已配置-Djava.awt.headless=true 不健康：未配置-Djava.awt.headless=true 修改建议：建议配置 -Djava.awt.headless=true 以启用 headless 模式
0808	dump 导出	是	否 健康：已配置 HeapDumpOnOutOfMemoryError 及 HeapDumpPath 不健康：未配置 HeapDumpOnOutOfMemoryError 及 HeapDumpPath 修改建议：建议配置 HeapDumpOnOutOfMemoryError 及 HeapDumpPath 以保证正常导出 dump
0809	BytecodeRecompilationCutoff 配置	是	是 健康：-XX:PerBytecodeRecompilationCutoff=-1 不健康：-XX:PerBytecodeRecompilationCutoff≠-1 修改建议：建议配置-XX:PerBytecodeRecompilationCutoff=-1
0810	PerMethodRecompilationCutoff 配置	是	是 健康：-XX:PerMethodRecompilationCutoff=-1 不健康：-XX:PerMethodRecompilationCutoff≠-1

				<div>修改建议：建议配置-XX:PerMethodRecompilationCutoff=-1</div> <div>注：FineBI 特有检查项</div> <div>健康：非 exe 方式部署</div>
0811 安装包部署	是	否	不健康：exe 方式部署	<div>修改建议：建议使用 web 容器方式部署。</div> <div>详情请参见：<a href="#">Tomcat 服务器部署</a></div> <div>FR：</div> <div>健康：-XX:ReservedCodeCacheSize≥250m</div> <div>不健康：-XX:ReservedCodeCacheSize &lt; 250m</div> <div>修改建议：建议配置-XX:ReservedCodeCacheSize=250m</div>
0813 codecache 配置	是	是	BI：	<div>健康：-XX:ReservedCodeCacheSize≥1024m</div> <div>不健康：-XX:ReservedCodeCacheSize &lt; 1024m</div> <div>修改建议：建议配置-XX:ReservedCodeCacheSize=1024m</div> <div>健康：-Duser.country=CN</div>
0814 语言配置-Duser.country	是	否	不健康：-Duser.country≠CN	<div>修改建议：建议配置-Duser.country=CN</div> <div>健康：-Duser.language=zh</div>
0815 语言配置-Duser.language	是	否	不健康：-Duser.language≠zh	<div>修改建议：建议配置-Duser.language=zh</div> <div>健康：-Dsun.zip.disableMemoryMapping=true</div>
0816 zip 相关配置-Dsun.zip.disableMemoryMapping	是	否	不健康：-Dsun.zip.disableMemoryMapping≠true	<div>修改建议：建议配置-Dsun.zip.disableMemoryMapping=true</div>

				健康： -XX:NewRatio=2
0817	堆内老年代与新生代大小比例	是	是	不健康： -XX:NewRatio≠2 修改建议： 建议堆内老年代与新生代大小比例为 2，即-XX:NewRatio=2
0818	栈空间	是	是	健康： -Xss≤1024 不健康： -Xss > 1024 修改建议： 建议单线程使用栈空间不超过 1024 KB，即 -Xss 的值不超过 1024
0819	debug 模式	是	否	健康： 未配置参数 -agentlib:jdwp 和 -Xrunjdwp 不健康： 配置了参数 -agentlib:jdwp 和 -Xrunjdwp 修改建议： 建议取消 debug 模式，删除两个参数 -agentlib:jdwp 和 -Xrunjdwp
0820	堆内内存	是	是	健康： 根据系统实际情况计算出一个符合健康标准的堆内内存要求范围，当前堆内内存存在计算出的符合要求的堆内内存范围内 不健康： 当前堆内内存不在计算出的符合要求的堆内内存范围内 修改建议： 为保证应用响应性能，建议配置为 XGB FineBI 额外提醒: 如果需要使用更大内存，在保证物理机内存充足的情况下，可以配置为 48G ~ 64G，但可能带来一定的响应性能下降
0821	xms 配置	是	是	健康： Xms=Xmx 不健康： Xms≠Xmx 修改建议： 建议配置 Xms=Xmx
0822	jdk 位置	是	否	仅输出信息

			健康：-XX:MaxMetaspaceSize≥2G
0823 元数据空间配置	是	是	不健康：-XX:MaxMetaspaceSize < 2G，或未配置
			修改建议：建议最大元数据空间设置为 2g

0801 堆外 NIO 内存

1) 建议值

- FR：建议配置 JVM NIO 占用堆外内存 2GB
- BI：建议配置 JVM NIO 占用堆外内存 4GB

2) 修改方式

NIO 由-XX:MaxDirectMemorySize 这个参数控制。请根据自身工程部署情况选择修改方法。本节展示 Tomcat 环境下设置 NIO 内存 的步骤。

- **Windows:** 进入%Tomcat\_HOME%\bin 目录。寻找名为 setenv.bat 的文件。如果文件不存在，可以创建一个新的 setenv.bat 文件。在文件中添加以下行来设置，配置完成后重启工程。

set JAVA\_OPTS=%JAVA\_OPTS% -XX:MaxDirectMemorySize=4g

- **Linux:** 进入%Tomcat\_HOME%\bin 目录。寻找名为 setenv.sh 的文件。如果文件不存在，可以创建一个新的 setenv.sh 文件。在文件中添加以下行来设置，配置完成后重启工程。

JAVA\_OPTS="\$JAVA\_OPTS -XX:MaxDirectMemorySize=4g"

0802 堆外 fineIO 读内存

1) 建议值

建议 JVM fineIO 读内存使用堆外内存设置为 2G。

2) 修改方式

fineIO 读内存由-Dfineio.read\_mem\_limit 这个参数来设置。请根据自身工程部署情况选择修改方法。本节展示 Tomcat 环境下设置 fineIO 读内存 的步骤。

读内存的默认单位为 GB，支持小数，但不支持使用 g、m 等。建议读内存为写内存

的 2-3 倍，缓存大小建议与写内存大小一致。

- **Windows:** 进入 `%Tomcat_HOME%\bin` 目录。寻找名为 `setenv.bat` 的文件。如果文件不存在，可以创建一个新的 `setenv.bat` 文件。在文件中添加以下行来设置，配置完成后重启工程。

```
set JAVA_OPTS=%JAVA_OPTS% -Dfineio.read_mem_limit=2
```

- **Linux:** 进入 `%Tomcat_HOME%\bin` 目录。寻找名为 `setenv.sh` 的文件。如果文件不存在，可以创建一个新的 `setenv.sh` 文件。在文件中添加以下行来设置，配置完成后重启工程。

```
JAVA_OPTS="$JAVA_OPTS -Dfineio.read_mem_limit=2"
```

## 0803 堆外 fineIO 写内存

### 1) 建议值

建议 JVM fineIO 写内存使用堆外内存设置为 1G。

### 2) 修改方式

fineIO 写内存由 `-Dfineio.write_mem_limit` 这个参数来设置。请根据自身工程部署情况选择修改方法。本节展示 Tomcat 环境下设置 fineIO 写内存 的步骤。

写内存的默认单位为 GB，支持小数，但不支持使用 g、m 等。建议读内存为写内存的 2-3 倍，缓存大小建议与写内存大小一致。

- **Windows:** 进入 `%Tomcat_HOME%\bin` 目录。寻找名为 `setenv.bat` 的文件。如果文件不存在，可以创建一个新的 `setenv.bat` 文件。在文件中添加以下行来设置，配置完成后重启工程。

```
set JAVA_OPTS=%JAVA_OPTS% -Dfineio.write_mem_limit=1
```

- **Linux:** 进入 `%Tomcat_HOME%\bin` 目录。寻找名为 `setenv.sh` 的文件。如果文件不存在，可以创建一个新的 `setenv.sh` 文件。在文件中添加以下行来设置，配置完成后重启工程。

```
JAVA_OPTS="$JAVA_OPTS -Dfineio.write_mem_limit=1"
```

## 0804 垃圾回收器类型



## 1) 建议值

建议 JVM 使用垃圾回收器类型为 ParallelScavenge

## 2) 修改方式

本节以工程部署在 Tomcat 上为例，介绍设置垃圾收集器为 Parallel Scavenge（并行收集器）的方法。

- **Windows:** 进入 %Tomcat\_HOME%\bin 目录。寻找名为 setenv.bat 的文件。如果文件不存在，可以创建一个新的 setenv.bat 文件。在文件中添加以下行来设置，配置完成后重启工程。

```
set JAVA_OPTS=%JAVA_OPTS% -XX:+UseParallelGC
```

- **Linux:** 进入 %Tomcat\_HOME%\bin 目录。寻找名为 setenv.sh 的文件。如果文件不存在，可以创建一个新的 setenv.sh 文件。在文件中添加以下行来设置，配置完成后重启工程。

```
JAVA_OPTS="$JAVA_OPTS -XX:+UseParallelGC"
```

## 0805 DisableExplicitGC 参数

### 1) 建议值

建议不对 DisableExplicitGC 参数进行配置，此项配置会导致 System.gc()被禁用，影响系统稳定性。

### 2) 修改方式

-XX:+DisableExplicitGC 参数的作用是禁用 System.gc()。System.gc() 是一种保护机制，例如堆外内存满时清理它的堆内引用对象。用户需要根据实际情况选择是否使用该参数，建议删除该参数。

本节展示 Tomcat 环境下删除 DisableExplicitGC 参数的步骤。

- **Windows:** 进入 %Tomcat\_HOME%\bin 目录。寻找名为 catalina.bat 和 setenv.bat 的文件。

在文件中查找是否存在 XX:+DisableExplicitGC，若存在，删除相关配置，修改完成后重启工程。

- **Linux:** 进入%Tomcat\_HOME%\bin 目录。寻找名为 catalina.sh 和 setenv.sh 的文件。

在文件中查找是否存在 XX:+DisableExplicitGC，若存在，删除相关配置，修改完成后重启工程。

## 0806 JDK 版本

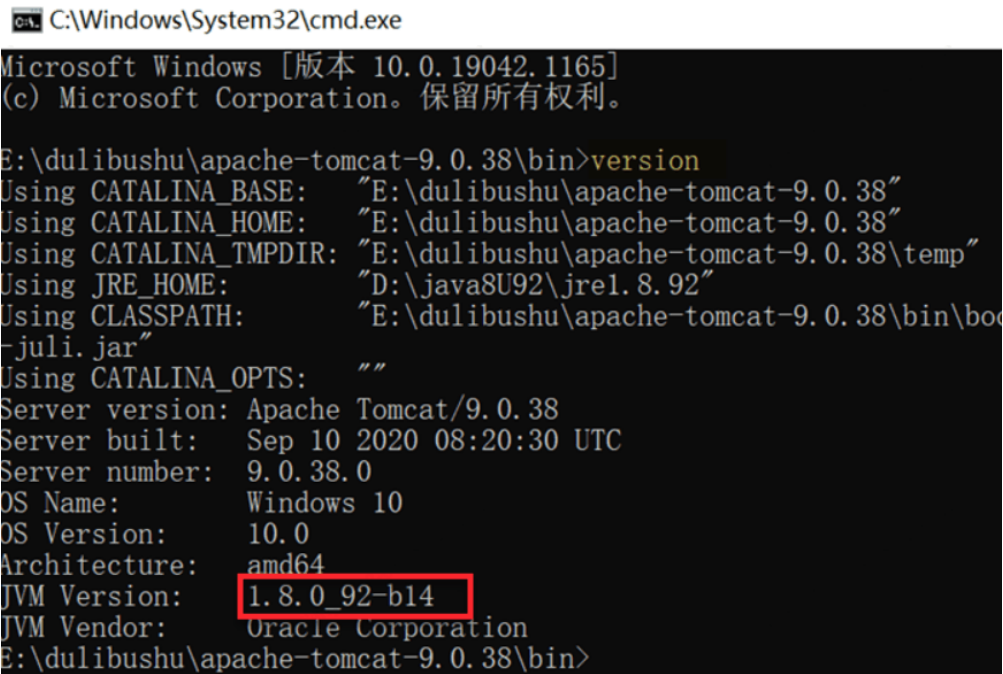
### 1) 建议值

建议使用「JDK8」中 1.8.0\_181 及以上版本

### 2) 检查方式

cmd 命令行进入%Tomcat\_home%\bin 路径下，输入 version，即可查看 JDK 版本。

如下图所示：



```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.19042.1165]
(c) Microsoft Corporation。保留所有权利。

E:\dulibushu\apache-tomcat-9.0.38\bin>version
Using CATALINA_BASE:   "E:\dulibushu\apache-tomcat-9.0.38"
Using CATALINA_HOME:   "E:\dulibushu\apache-tomcat-9.0.38"
Using CATALINA_TMPDIR: "E:\dulibushu\apache-tomcat-9.0.38\temp"
Using JRE_HOME:        "D:\java8U92\jre1.8.92"
Using CLASSPATH:       "E:\dulibushu\apache-tomcat-9.0.38\bin\bootstrap.jar;E:\dulibushu\apache-tomcat-9.0.38\bin\tools.jar"
Using CATALINA_OPTS:   ""
Server version: Apache Tomcat/9.0.38
Server built:   Sep 10 2020 08:20:30 UTC
Server number:  9.0.38.0
OS Name:        Windows 10
OS Version:     10.0
Architecture:   amd64
JVM Version:    1.8.0_92-b14
JVM Vendor:     Oracle Corporation
E:\dulibushu\apache-tomcat-9.0.38\bin>
```

## 0807 headless 模式

### 1) 建议值

建议配置 -Djava.awt.headless=true 以启用 headless 模式

### 2) 修改方式

headless 模式是系统的一种工作模式，如果系统属性 java.awt.headless 被设置 true，那么 headless 工具包就会被使用。

本节展示 Tomcat 环境下启用 headless 模式的步骤。

- **Windows:** 进入 `%Tomcat_HOME%\bin` 目录。寻找名为 `setenv.bat` 的文件。如果文件不存在，可以创建一个新的 `setenv.bat` 文件。在文件中添加以下行来设置，配置完成后重启工程。

```
set JAVA_OPTS=%JAVA_OPTS% -Djava.awt.headless=true
```

- **Linux:** 进入 `%Tomcat_HOME%\bin` 目录。寻找名为 `setenv.sh` 的文件。如果文件不存在，可以创建一个新的 `setenv.sh` 文件。在文件中添加以下行来设置，配置完成后重启工程。

```
JAVA_OPTS="$JAVA_OPTS -Djava.awt.headless=true"
```

## 0808 dump 导出

### 1) 建议值

建议配置 `HeapDumpOnOutOfMemoryError` 及 `HeapDumpPath` 以保证正常导出 dump

1) `-XX:+HeapDumpOnOutOfMemoryError` 参数表示当 JVM 发生 OOM 时，自动生成 DUMP 文件。

2) `-XX:HeapDumpPath=${目录}` 参数表示生成 dump 文件的路径，也可以指定文件名，例如：`-XX:HeapDumpPath=${目录}/java_heapdump.hprof`。

如果不指定文件名，默认为：`java_<pid>_<date>_<time>_heapDump.hprof`。

本节展示 Tomcat 环境下设置 dump 导出的步骤。

- **Windows:** 进入 `%Tomcat_HOME%\bin` 目录。寻找名为 `setenv.bat` 的文件。如果文件不存在，可以创建一个新的 `setenv.bat` 文件。在文件中添加以下行来设置，配置完成后重启工程。

```
set JAVA_OPTS=%JAVA_OPTS% -XX:+HeapDumpOnOutOfMemoryError -
XX:HeapDumpPath=${目录}
```

- **Linux:** 进入 `%Tomcat_HOME%\bin` 目录。寻找名为 `setenv.sh` 的文件。如果文件不存在，可以创建一个新的 `setenv.sh` 文件。在文件中添加以下行来设置，配置完成后重启工程。

```
JAVA_OPTS="$JAVA_OPTS -XX:+HeapDumpOnOutOfMemoryError -
```

XX:HeapDumpPath=\${目录}"

## 0809 BytecodeRecompilationCutoff 配置

## 0810 PerMethodRecompilationCutoff 配置

### 1) 建议值

建议 recompilationCutoff 相关参数配置值为-1

### 2) 修改方式

没有配置 RecompilationCutoff 参数/配置的 RecompilationCutoff 参数太小，可能会导致 spider 引擎数据更新缓慢。

本节展示 Tomcat 环境下设置 recompilationCutoff 参数的步骤。

- **Windows:** 进入%Tomcat\_HOME%\bin 目录。寻找名为 setenv.bat 的文件。如果文件不存在，可以创建一个新的 setenv.bat 文件。在文件中添加以下行来设置，配置完成后重启工程。

```
set JAVA_OPTS=%JAVA_OPTS% -XX:PerMethodRecompilationCutoff=-1 -  
XX:PerBytecodeRecompilationCutoff=-1
```

- **Linux:** 进入%Tomcat\_HOME%\bin 目录。寻找名为 setenv.sh 的文件。如果文件不存在，可以创建一个新的 setenv.sh 文件。在文件中添加以下行来设置，配置完成后重启工程。

```
JAVA_OPTS="$JAVA_OPTS -XX:PerMethodRecompilationCutoff=-1 -  
XX:PerBytecodeRecompilationCutoff=-1"
```

## 0811 安装包部署

注：仅 FineBI 系统检测该项

建议使用 web 容器方式部署。详情请参见：[Tomcat 服务器部署](#)

## 0813 codecache 配置

### 1) 建议值

建议配置-XX:ReservedCodeCacheSize=250m

### 2) 修改方式

ReservedCodeCacheSize 参数是调整内存大小的一个设置参数，值的大小设置直接影响到 Code Cache 的大小。

如果 Code Cache 的空间不足，可能会导致 JVM 无法正常运行，导致性能持续降低。

- **Windows:** 进入 `%Tomcat_HOME%\bin` 目录。寻找名为 `setenv.bat` 的文件。如果文件不存在，可以创建一个新的 `setenv.bat` 文件。在文件中添加以下行来设置，配置完成后重启工程。

```
set JAVA_OPTS=%JAVA_OPTS% -XX:ReservedCodeCacheSize=250m
```

- **Linux:** 进入 `%Tomcat_HOME%\bin` 目录。寻找名为 `setenv.sh` 的文件。如果文件不存在，可以创建一个新的 `setenv.sh` 文件。在文件中添加以下行来设置，配置完成后重启工程。

```
JAVA_OPTS="$JAVA_OPTS -XX:ReservedCodeCacheSize=250m"
```

#### 0814 语言配置-Duser.country

#### 0815 语言配置-Duser.language

##### 1) 建议值

建议配置-Duser.country=CN、-Duser.language=zh

##### 2) 修改方式

为了防止中文出现乱码现象，要配置好 Tomcat 的语言编码。

- **Windows:** 进入 `%Tomcat_HOME%\bin` 目录。寻找名为 `setenv.bat` 的文件。如果文件不存在，可以创建一个新的 `setenv.bat` 文件。在文件中添加以下行来设置，配置完成后重启工程。

```
set JAVA_OPTS=%JAVA_OPTS% -Duser.country=CN -Duser.language=zh
```

- **Linux:** 进入 `%Tomcat_HOME%\bin` 目录。寻找名为 `setenv.sh` 的文件。如果文件不存在，可以创建一个新的 `setenv.sh` 文件。在文件中添加以下行来设置，配置完成后重启工程。

```
JAVA_OPTS="$JAVA_OPTS -Duser.country=CN -Duser.language=zh"
```

#### 0816 zip 相关配置-Dsun.zip.disableMemoryMapping

##### 1) 建议值

建议配置-Dsun.zip.disableMemoryMapping=true

## 2) 修改方式

配置该参数，可预防 jvm crash

- **Windows:** 进入`%Tomcat_HOME%\bin` 目录。寻找名为 `setenv.bat` 的文件。如果文件不存在，可以创建一个新的 `setenv.bat` 文件。在文件中添加以下行来设置，配置完成后重启工程。

```
set JAVA_OPTS=%JAVA_OPTS% -Dsun.zip.disableMemoryMapping=true
```

- **Linux:** 进入`%Tomcat_HOME%\bin` 目录。寻找名为 `setenv.sh` 的文件。如果文件不存在，可以创建一个新的 `setenv.sh` 文件。在文件中添加以下行来设置，配置完成后重启工程。

```
JAVA_OPTS="$JAVA_OPTS -Dsun.zip.disableMemoryMapping=true"
```

## 0817 堆内老年代与新生代大小比例

### 1) 建议值

建议 JVM 老年代与新生代大小比例设置为 2

### 2) 修改方式

堆内老年代与新生代大小比例建议为 2 (-XX:NewRatio=2, JVM 的默认值); NewRatio 大小正常生效需要用户不限制新生代大小。

在设置-Xmx、-Xms 参数的位置，添加 NewRatio 的值即可，本节展示 Tomcat 环境下设置 NewRatio 的步骤。

- **Windows:** 进入`%Tomcat_HOME%\bin` 目录。寻找名为 `setenv.bat` 的文件。如果文件不存在，可以创建一个新的 `setenv.bat` 文件。在文件中添加以下行来设置，配置完成后重启工程。

```
set JAVA_OPTS=%JAVA_OPTS% -Xms512M -Xmx1024M -XX:NewRatio=2
```

- **Linux:** 进入`%Tomcat_HOME%\bin` 目录。寻找名为 `setenv.sh` 的文件。如果文件不存在，可以创建一个新的 `setenv.sh` 文件。在文件中添加以下行来设置，配置完成后重启工程。

```
JAVA_OPTS="$JAVA_OPTS -Xms2048M -Xmx8196M -XX:NewRatio=2"
```

## 0818 栈空间

### 1) 建议值

建议单线程使用栈空间不超过 1024KB

### 2) 修改方式

建议单线程使用栈空间不超过 1024KB，即 -Xss 的值不超过 1024 。

在设置-Xmx、-Xms 参数的位置，添加 -Xss 的值即可，本节展示 Tomcat 环境下设置 -Xss 的步骤。

- **Windows:** 进入 %Tomcat\_HOME%\bin 目录。寻找名为 setenv.bat 的文件。如果文件不存在，可以创建一个新的 setenv.bat 文件。在文件中添加以下行来设置，配置完成后重启工程。

```
set JAVA_OPTS=%JAVA_OPTS% -Xms512M -Xmx1024M -Xss1024K
```

- **Linux:** 进入 %Tomcat\_HOME%\bin 目录。寻找名为 setenv.sh 的文件。如果文件不存在，可以创建一个新的 setenv.sh 文件。在文件中添加以下行来设置，配置完成后重启工程。

```
JAVA_OPTS="$JAVA_OPTS -Xms2048M -Xmx8196M -Xss1024K"
```

## 0819 debug 模式

### 1) 建议值

建议取消 debug 模式

### 2) 修改方式

debug 模式由两个参数控制：-agentlib:jdwp 和 -Xrunjdwp，删除这两个参数即可取消 debug 模式

- **Windows:** 在 %Tomcat%/bin 目录下的 catalina.bat 和 setenv.bat 文件中删除 -agentlib:jdwp 和 -Xrunjdwp 参数，配置完成后重启工程。
- **Linux:** 在 %Tomcat%/bin 目录下的 catalina.sh 和 setenv.sh 文件中删除 -agentlib:jdwp 和 -Xrunjdwp 参数，配置完成后重启工程。

0820 堆内内存

0821 xms 配置

参数介绍	服务器类型	参考文档
-Xmx 参数：最大堆内内存	Tomcat 服务器	<a href="#">Tomcat 服务器内存修改</a>
	WebLogic 服务器	<a href="#">WebLogic 服务器内存修改</a>
-Xms 参数：初始化内存大小	WebSphere 服务器	<a href="#">WebSphere 服务器</a>
	JBoss 服务器	<a href="#">JBoss 服务器</a>
注 1：Xmx/Xms 与数字之间不要有空格	Resin 服务器	<a href="#">Resin 服务器内存修改</a>
注 2：建议配置 Xms=Xmx，以防止内存扩容失败情况		

0823 元数据空间配置

1) 参数简介：

元数据空间（Metaspace）是 Java 虚拟机（JVM）用于存储类元数据的区域。在 JVM 中，元数据包括类的结构信息、方法信息、字段信息等。

元数据空间配置 MaxMetaspaceSize 用于设置元数据空间的最大大小。当元数据空间超过这个限制时，JVM 会发生垃圾回收以释放不再使用的元数据。

2) 建议配置：

建议配置 MaxMetaspaceSize 参数为 2G

3) 修改方式：

本节展示 Tomcat 环境下设置 MaxMetaspaceSize 的步骤。

- **Windows:** 进入%Tomcat\_HOME%\bin 目录。寻找名为 setenv.bat 的文件。如果文件不存在，可以创建一个新的 setenv.bat 文件。在文件中添加以下行来设置，配置完成后重启工程。

set JAVA\_OPTS=%JAVA\_OPTS% -XX:MaxMetaspaceSize=2G

- **Linux:** 进入%Tomcat\_HOME%\bin 目录。寻找名为 setenv.sh 的文件。如果文件不存在，可以创建一个新的 setenv.sh 文件。在文件中添加以下行来设置，配置完成后重启工程。

JAVA\_OPTS="\$JAVA\_OPTS -XX:MaxMetaspaceSize=2G"

服务联通性检查

服务联通性检查的检测项包括：



是否支持				
巡检项	检测项	集群节点是	支持	健康标准
编号		否	分别检查	快速修复
0901	Redis 联通	是	否	检测工程与 Redis 组件之间是否联通
0902	文件服务器联通	是	否	检测工程与文件服务器组件之间是否联通
0903	FineDB 联通	是	否	检测工程与 FineDB 数据库之间是否联通

业务检查

业务检查的检测项包括：

巡检项	检测项	是	否	健康标准
1001	集群节点	否	支持	
	是	支持	快速修复	
	否	分别检查	快速修复	
1001	检测 finedb 数据库的类型	否	否	检测 finedb 数据库的类型
1002	日志级别	否	否	健康：finedb 迁移到了外接数据库
				不健康：finedb 使用内置 HSQL 数据库
				修改建议：建议将 finedb 迁移至更稳定的外接数据库中。
1003	检测「管理系统>智能运维>平台日志>全局设置」中的「系统日志级别」	否	否	健康：日志级别高于 debug 级别
1003	日志清理	否	否	不健康：日志级别为 debug 级别
				修改建议：建议设置日志级别为 info。
				检测「管理系统>智能运维>平台日志>全局设置」中「自动

					清理」周期
					健康：配置了自动清理，自动清理超过三个月的日志
					不健康：未配置自动清理
					修改建议：建议开启平台日志自动清理并配置清理周期小于六个月。
					检测「管理系统>智能运维>备份还原>全局设置」中设置的自动备份可占用磁盘大小
100	自动备份磁盘占用限制	否	否	健康：各模块自动备份可占磁盘不超过 1024M	
4				不健康：各模块自动备份可占磁盘超过 1024M	
				修改建议：建议各模块自动备份可占磁盘不超过 1024M。	
					检测「管理系统>智能运维>备份还原>全局设置」中设置的自动备份保留数量
100	自动备份保留数量限制	否	否	健康：各模块成功/失败自动备份可分别保留不超过 5 份	
5				不健康：各模块成功/失败自动备份可分别保留超过 5 份	
				修改建议：建议自动备份默认保存不超过 5 份。	
					检测定时调度任务>调度对象中，文件处理方式的设置
					健康：仅保留最近一次或任务结束即清理
100	定时调度任务清理配置	否	否	不健康：其他文件处理方式	
6				修改建议：建议定时调度调度对象文件处理设置为「仅保留最近一次」。	
				注：仅 FineBI 系统检测该项	
					检测「管理系统>系统管理>常规>BI 参数」中的「数据访问量」参数
100	内存数据行数限制	否	否	健康：不超过 100w	
7				不健康：超过 100w	
				修改建议：建议数据访问量不超过 100w。	
100	快速分析单元	否	否	注：仅 FineBI 系统检测该项	
8					

			格数限制			检测系统的 spider_fast_compute_limit_unit 参数值
						健康：DistributedOptimizationConfig.spiderConfig.spider_fast_compute_limit_unit 参数值不超过 10000000
						不健康：DistributedOptimizationConfig.spiderConfig.spider_fast_compute_limit_unit 参数值超过 10000000
						注：仅 FineBI 系统检测该项
						检测系统的 spider_fast_compute_limit_memory 参数值
100	快速分析内存大小限制	否	否			健康：DistributedOptimizationConfig.spiderConfig.spider_fast_compute_limit_memory 参数值不超过 500000000
						不健康：DistributedOptimizationConfig.spiderConfig.spider_fast_compute_limit_memory 参数值超过 500000000
						注：仅 FineBI 系统检测该项
						检测系统的 spider_update_fast_compute_limit_cell 参数值
101	快速分析更新单元格数限制	否	否			健康：DistributedOptimizationConfig.spiderConfig.spider_update_fast_compute_limit_cell 参数值不超过 100000000
						不健康：DistributedOptimizationConfig.spiderConfig.spider_update_fast_compute_limit_cell 参数值超过 100000000
						注：仅 FineBI 系统检测该项
						检测系统的 spider_fast_compute_limit_row 参数值
101	快速分析行数限制	否	否			健康：DistributedOptimizationConfig.spiderConfig.spider_fast_compute_limit_row 参数值不超过 10000000
						不健康：DistributedOptimizationConfig.spiderConfig.spider_fast_compute_limit_row 参数值超过 10000000
						注：仅 FineBI 系统检测该项
101	spark 计算结果内存大小限制	否	否			检测系统的 spiderConfig.spark_driver_maxResultSize 参数值
						健康：DistributedOptimizationConfig.spiderConfig.spark_driver_maxResultSize 参数值不超过 10000000

				不健康： DistributedOptimizationConfig.spiderConfig.spark_driver_maxResultSize 参数值超过 10000000
1014	外接库的数据库编码/字符集	否	否	仅输出信息 包括 finedb 外接数据库的数据库编码/字符集 外接数据库配置步骤请参见： <a href="#">配置外接数据库</a>
1015	集群参数配置	否	否	仅输出信息 检测「管理系统>智能运维>集群配置>参数配置」中的内部转发参数 集群参数配置请参见： <a href="#">集群参数配置</a>  注： 仅 FineBI 系统检测该项
1016	更新资源无限制时间段配置	否	否	仅输出信息 检测「管理系统>系统管理>常规>抽取参数」 中的「更新高性能时间区间」 参数 BI 参数配置请参见： <a href="#">BI/Spider 参数</a>
1017	二级缓存开启情况	否	否	仅输出信息 检测系统是否配置二级缓存 二级缓存开启及配置请参见： <a href="#">指定 ehcache 缓存路径</a>
1018	sql 执行时长限制	否	否	检测「管理系统>智能运维>负载管理>模板限制>通用限制」中的 sql 执行时长限制 健康： 开启了 sql 执行时长限制 不健康： 未开启 sql 执行时长限制 修改建议： 建议开启系统管理-智能运维-负载管理-模板限制-通用限制-sql 执行时长限制功能。
1019	单模板单元格数限制	否	否	检测「管理系统>智能运维>负载管理>模板限制>通用限制」中的单模板单元格限制 健康： 开启了单模板单元格限制 不健康： 未开启单模板单元格限制

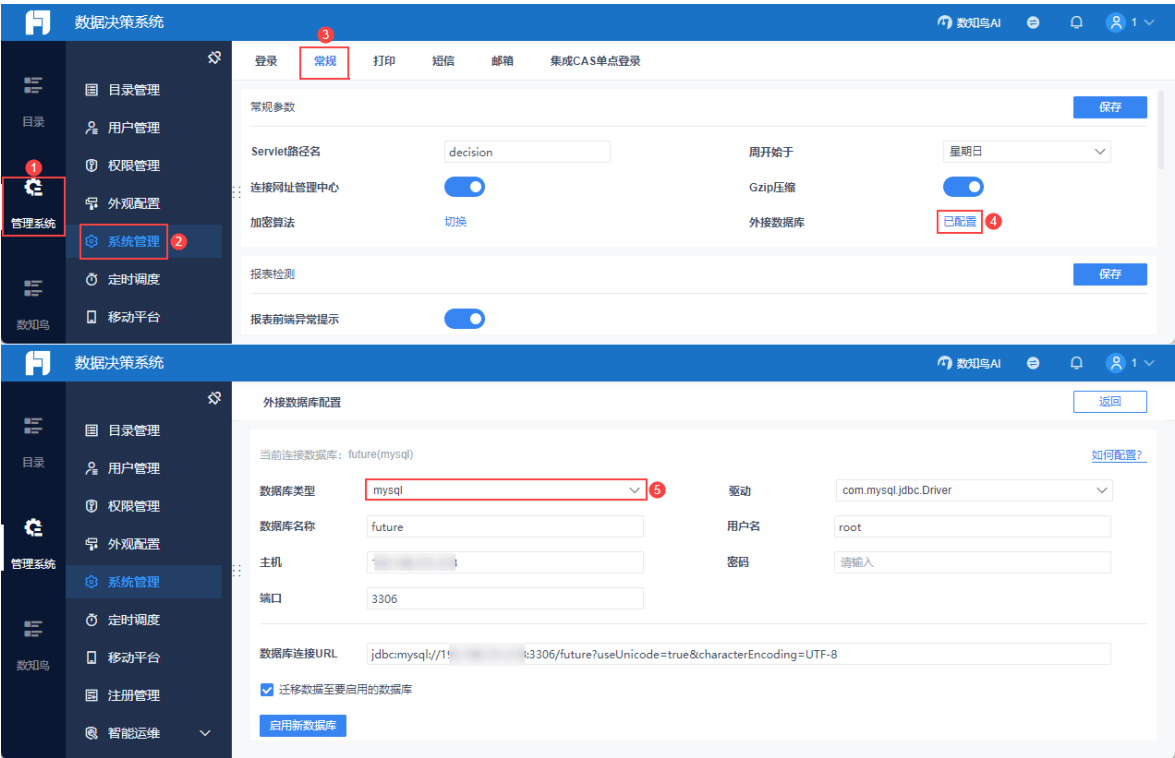
1020	导入 excel 单元格数限制	否	否	修改建议：建议开启系统管理-智能运维-负载管理-模板限制-通用限制-单模板单元格限制功能。
				检测「管理系统>智能运维>负载管理>模板限制>通用限制」中的导入 Excel 单元格限制
				健康：开启了导入 Excel 单元格限制 不健康：未开启导入 Excel 单元格限制
1022	单数据集行数限制	否	否	修改建议：建议开启系统管理-智能运维-负载管理-模板限制-通用限制-导入 excel 单元格限制功能。
				检测「管理系统>智能运维>负载管理>模板限制>通用限制」中的单数据集行数限制
				健康：开启了单数据集行数限制 不健康：未开启单数据集行数限制
1023	负载预警开启	否	否	修改建议：建议开启系统管理-智能运维-负载管理-模板限制-通用限制-单数据集行数限制功能。
				仅输出信息
				检测「管理系统>智能运维>负载管理>负载监控」中是否开启「智能预警」
1024	脏数据检测	否	否	详情请参见： <a href="#">负载管理</a>
				注：仅 FineBI 系统检测该项
				检测是否存在脏数据
1025	是否开启云端运维	否	否	健康：0 条脏数据 不健康：x 条脏数据
				修改方法：有 X 条脏数据，建议在「BI 工具」-「检测」-「脏数据」中进行排查清理
				仅输出信息
1025	是否开启云端运维	否	否	检测系统是否开启云端运维
				云端运维功能请参见： <a href="#">云端运维</a>

检测「管理系统>系统管理>登录」中是否开启「密码强度限制」			
健康： 开启了密码强度限制			
1026	密码强度限制检测	否	否 不健康：未开启密码强度限制
修改建议：未开启密码强度限制，密码强度较低可能容易被攻击，导致安全问题，建议在[管理系统]-[系统管理]-[登录]中打开密码强度限制			

1001 finedb

管理员登录帆软应用，点击「管理系统>系统管理>常规>外接数据库」，将 finedb 迁移至更稳定的外接数据库中。

外接数据库迁移的具体操作步骤请参见：[配置外接数据库](#)



1002 日志级别

管理员登录帆软应用，点击「管理系统>智能运维>平台日志>全局设置」。

在「系统日志级别」中设置日志级别为 info，点击「保存」即可。

日志级别设置请参见：[日志简介](#)



### 1003 日志清理

管理员登录帆软应用，点击「管理系统>智能运维>平台日志>全局设置」。

在「操作日志清理」中开启「自动清理」并配置清理周期小于六个月。

操作日志自动清理设置请参见：[日志简介](#)



### 1004 自动备份磁盘占用限制

管理员登录帆软应用，点击「管理系统>智能运维>备份还原>全局设置」。

在「自动备份」中，建议各模块自动备份可占磁盘不超过 1024M。备份还原设置请参

见：[备份还原](#)



1005 自动备份保留数量限制

管理员登录帆软应用，点击「管理系统>智能运维>备份还原>全局设置」。

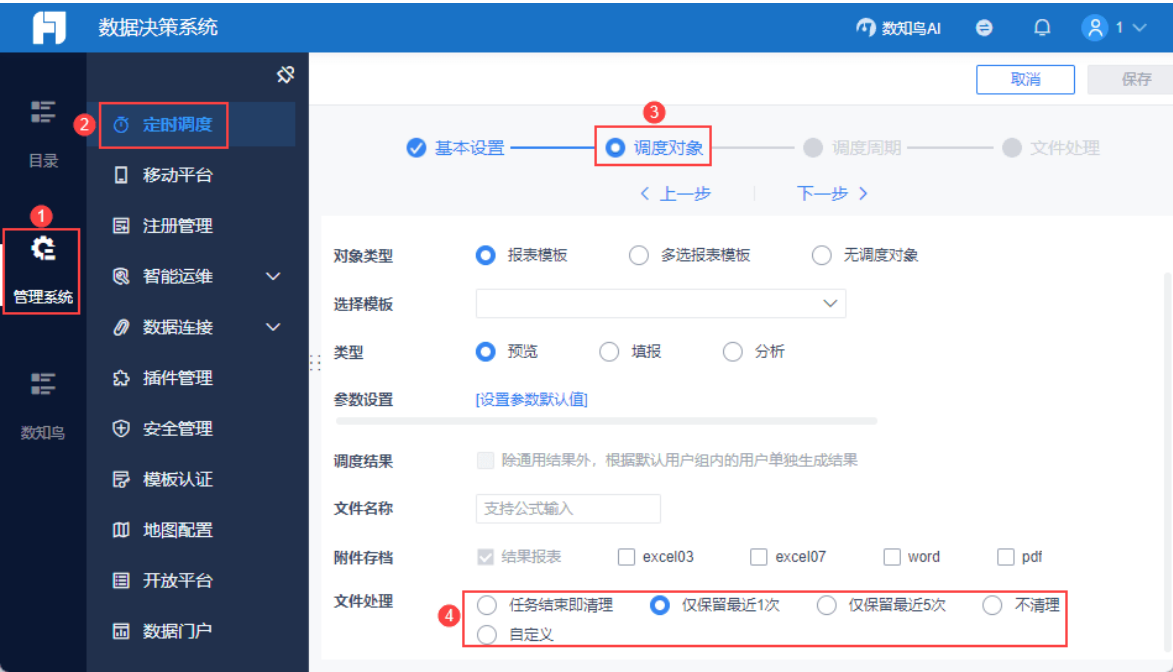
在「自动备份」中，建议自动备份默认保存不超过 5 份。备份还原设置请参见：[备份还原](#)



1006 定时调度任务清理配置

管理员登录帆软应用，在「定时调度」中，对「定时调度任务>调度对象」的文件处理方式设置为「仅保留最近一次」。

定时调度任务设置请参见：[定时调度任务设置](#)



1007 内存数据行数限制

注：仅 FineBI 系统检测该项

管理员登录帆软应用，点击「管理系统>系统管理>常规>BI 参数」。

建议「数据访问量」参数值不超过 100w。BI 参数配置请参见：[BI/Spider 参数](#)

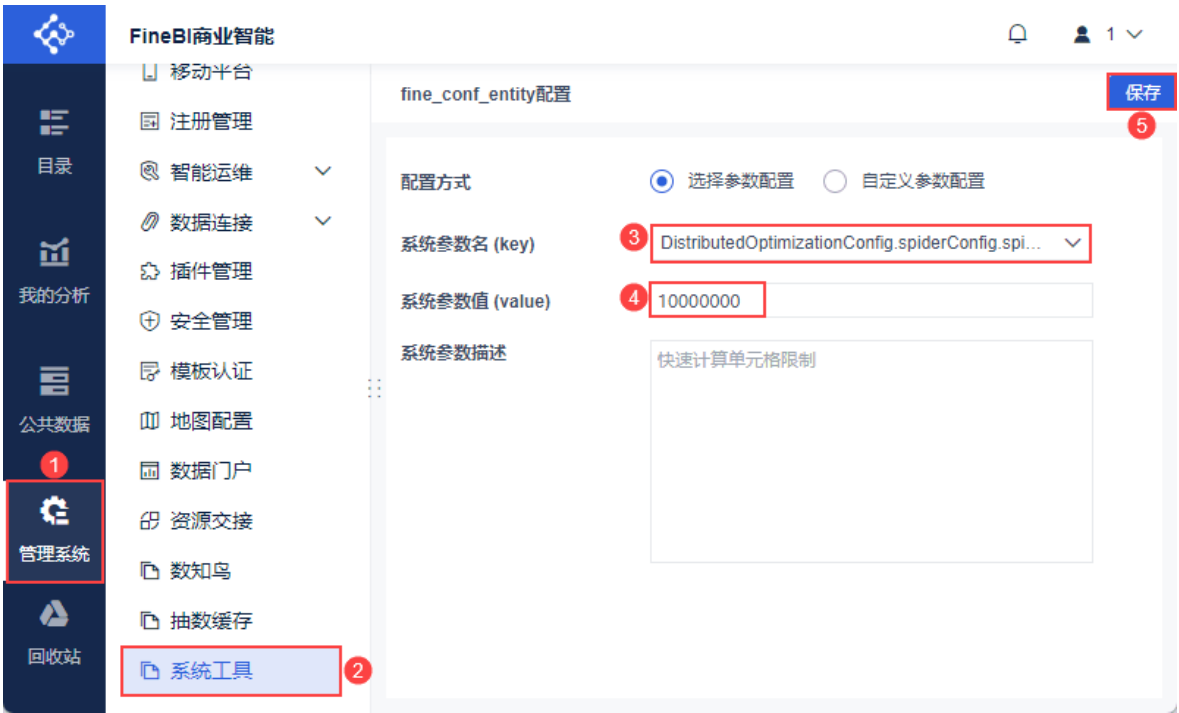




1008 快速分析单元格数限制

注：仅 FineBI 系统检测该项

「快速分析单元格数限制」的配置信息记录在 FineDB 的 fine\_conf\_entity 表中。  
请通过「[fine\\_conf\\_entity 可视化配置插件](#)」，配置 DistributedOptimizationConfig.spiderConfig.spider\_fast\_compute\_limit\_unit 参数值不超过 10000000  
点击「保存」，并重启 FineBI 工程，配置生效。



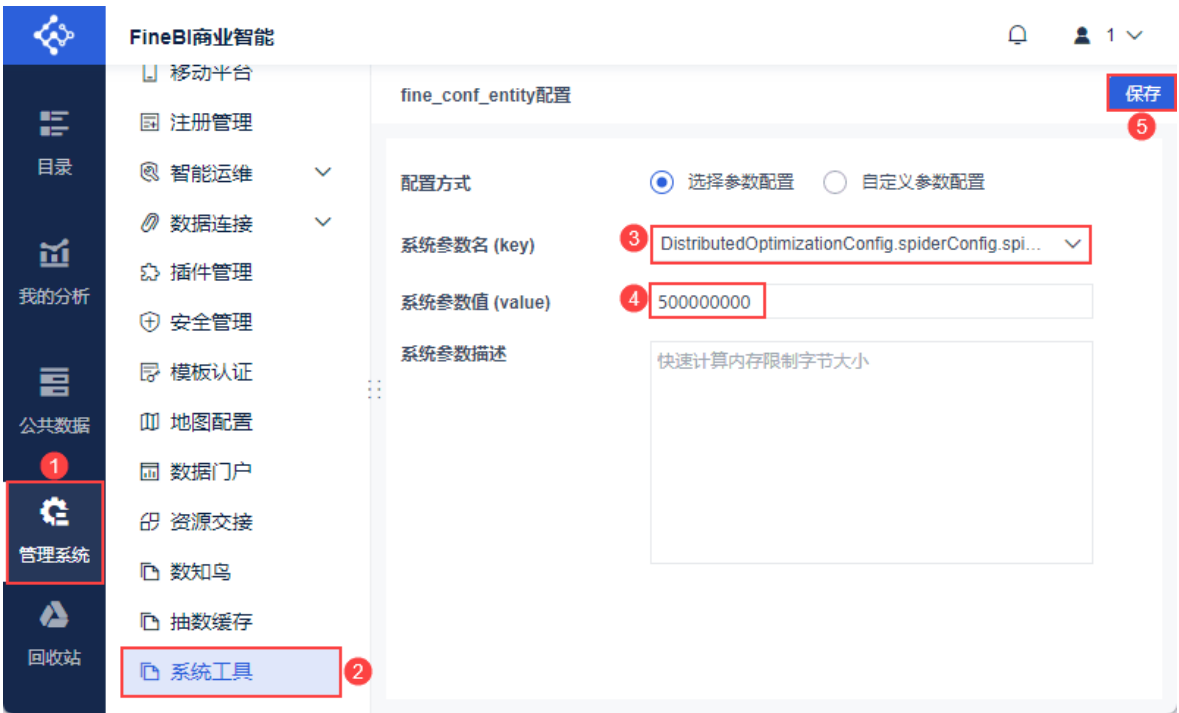
1009 快速分析内存大小限制

注：仅 FineBI 系统检测该项

「快速分析内存大小限制」的配置信息记录在 FineDB 的 fine\_conf\_entity 表中。  
请通过「[fine\\_conf\\_entity 可视化配置插件](#)」，配置 DistributedOptimizationConfig.spiderConfig.spider\_fast\_compute\_limit\_mem

ory 参数值不超过 500000000

点击「保存」，并重启 FineBI 工程，配置生效。



### 1010 快速分析更新单元格数限制

注：仅 FineBI 系统检测该项

「快速分析更新单元格数限制」的配置信息记录在 FineDB 的 fine\_conf\_entity 表中。

请通过「[fine\\_conf\\_entity 可视化配置插件](#)」，配置 DistributedOptimizationConfig.spiderConfig.spider\_update\_fast\_compute\_limit\_cell 参数值不超过 100000000

点击「保存」，并重启 FineBI 工程，配置生效。

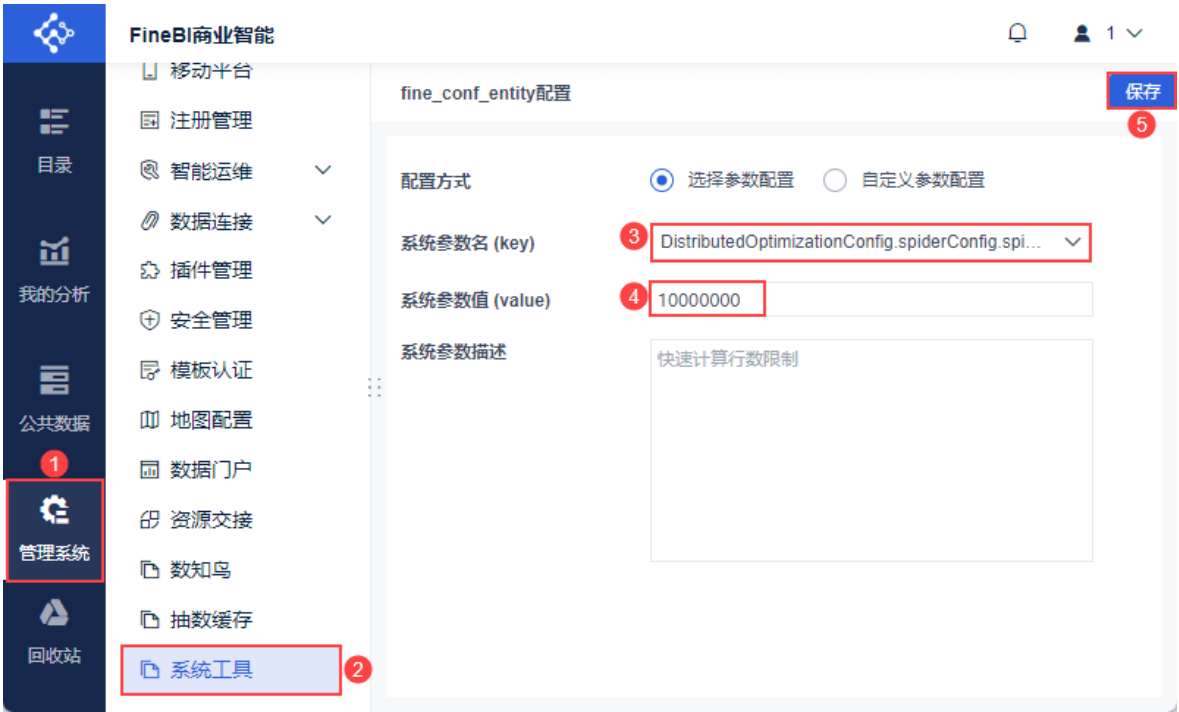


### 1011 快速分析行数限制

注：仅 FineBI 系统检测该项

「快速分析内存大小限制」的配置信息记录在 FineDB 的 fine\_conf\_entity 表中。请通过「[fine\\_conf\\_entity 可视化配置插件](#)」，配置 DistributedOptimizationConfig.spiderConfig.spider\_fast\_compute\_limit\_row 参数值不超过 10000000

点击「保存」，并重启 FineBI 工程，配置生效。



### 1012 spark 计算结果内存大小限制

注：仅 FineBI 系统检测该项

「spark 计算结果内存大小限制」的配置信息记录在 FineDB 的 fine\_conf\_entity 表中。

请通过「[fine\\_conf\\_entity 可视化配置插件](#)」，配置 DistributedOptimizationConfig.spiderConfig.spark\_driver\_maxResultSize 参数值不超过 10000000

点击「保存」，并重启 FineBI 工程，配置生效。



### 1018 sql 执行时长限制

管理员登录帆软应用，点击「管理系统>智能运维>负载管理>模板限制>通用限制」。

建议开启 sql 执行时长限制功能。模板限制请参见：[模板限制](#)



### 1019 单模板单元格数限制

管理员登录帆软应用, 点击「管理系统>智能运维>负载管理>模板限制>通用限制」。

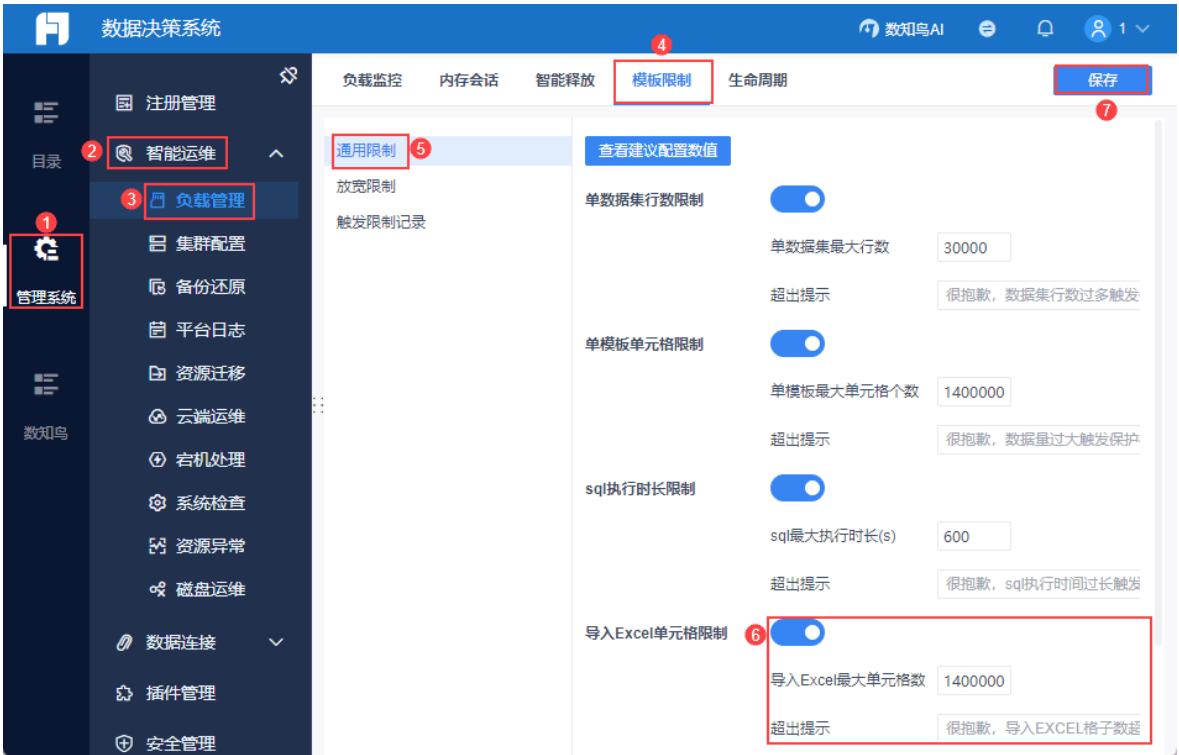
建议开启单模板单元格限制功能。模板限制请参见：[模板限制](#)



### 1020 导入 excel 单元格数限制

管理员登录帆软应用, 点击「管理系统>智能运维>负载管理>模板限制>通用限制」。

建议开启导入 excel 单元格限制功能。模板限制请参见：[模板限制](#)



## 1022 单数据集行数限制

管理员登录帆软应用，点击「管理系统>智能运维>负载管理>模板限制>通用限制」。

建议开启单数据集行数限制功能。模板限制请参见：[模板限制](#)

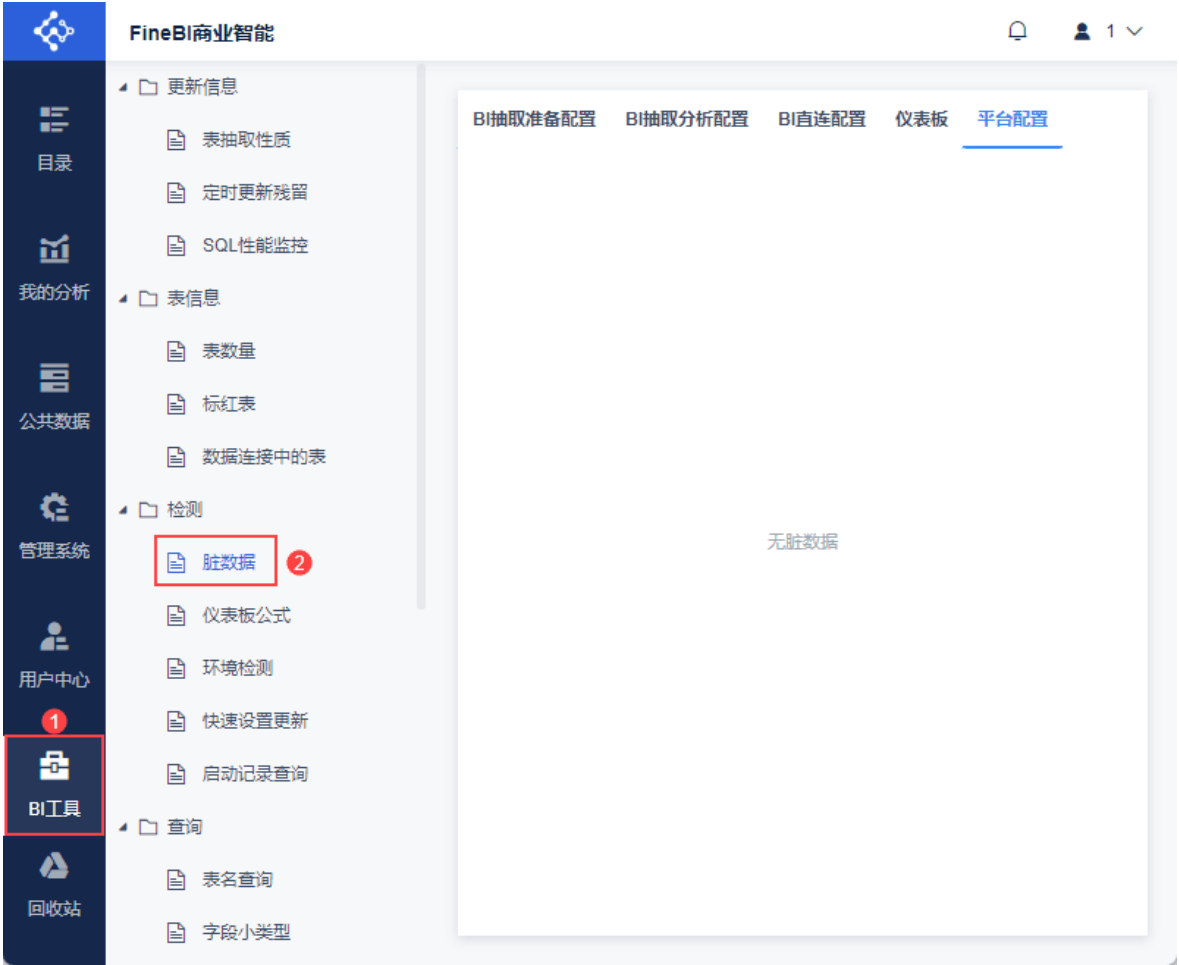


## 1024 脏数据检测

注：仅 FineBI 系统检测该项

管理员登录帆软应用，点击「BI 工具>检测>脏数据」。

检测系统中的脏数据，并进行排查清理。

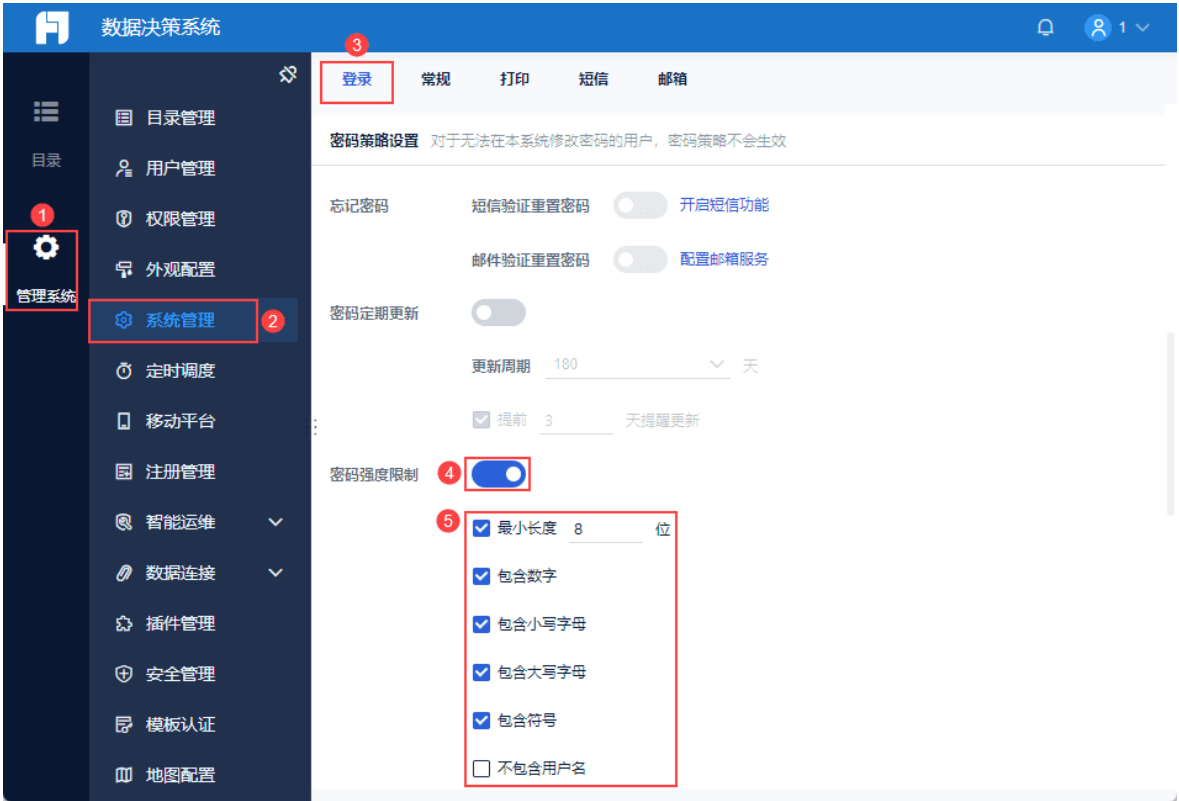


1026 密码强度限制检测

管理员登录帆软应用，点击「管理系统>系统管理>登录」。

在「密码策略设置」中，建议开启「密码强度限制」，并自定义设置密码强度。

详情请参见：[密码策略设置](#)



# 7.3 宕机自动处理

## 1. 概述

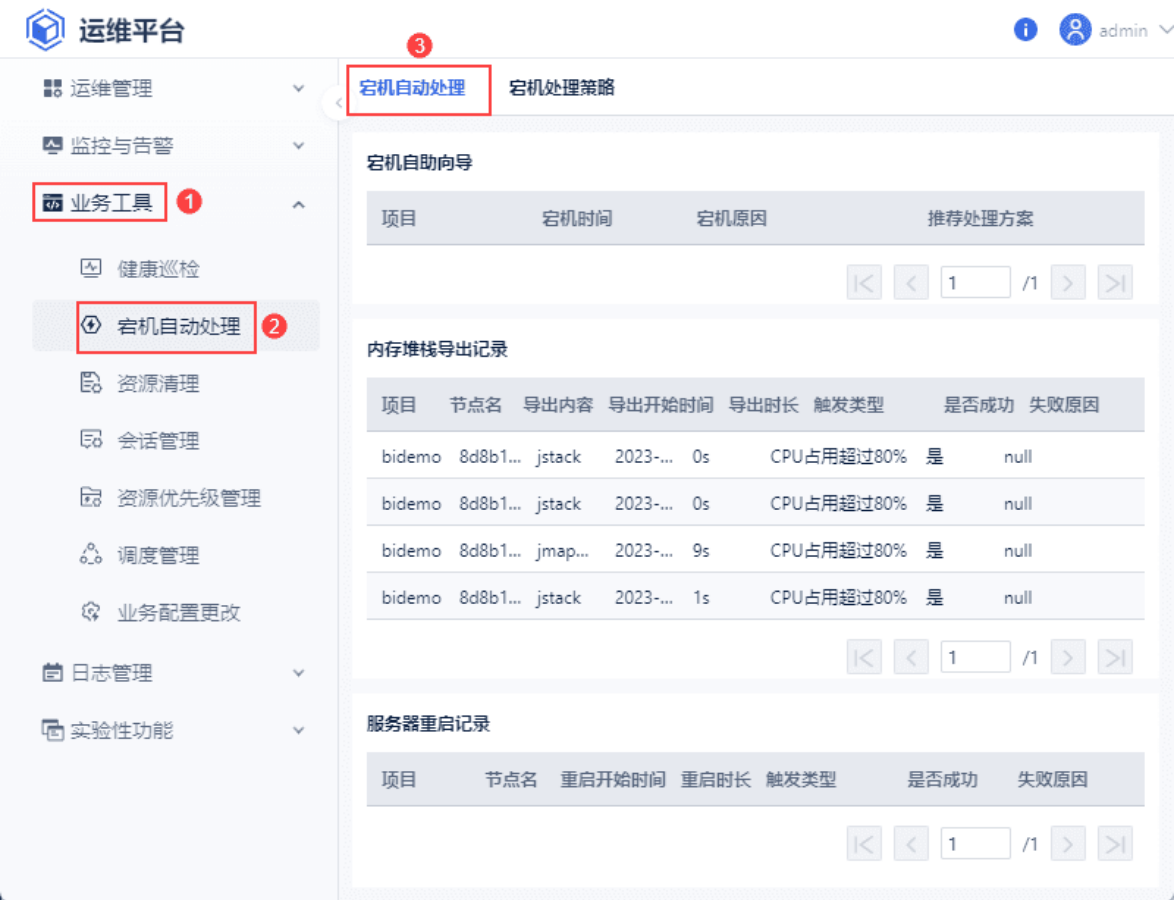
### 1.1 版本

运维平台版本	功能变更
V1.5.5	-
V1.6.0	内存堆栈导出记录，新增「触发类型」，记录自动导出堆栈的触发原因，方便后续分析

### 1.2 应用场景

报表工程部署在服务器上后，会因为各种原因导致宕机。宕机后很难定位到真实原因，从而无法快速解决问题和做一些防止再发生的预防措施，造成的时间和精力损失无法估计，对服务器运维非常不友好。

运维平台提供「宕机自动处理」功能。发生宕机后，通过分析生成的 DUMP 文件，可以快速定位宕机原因，及时处理宕机问题和做一些有效的预防措施，为用户增加一道保险。



## 2. 宕机自助向导

记录工程的宕机时间和宕机原因，并提供对应的推荐处理方案。如下图所示：





常见的宕机原因和推荐处理方案如下表所示：

宕机原因	推荐处理方案
内存溢出，可查看具体引发该问题的模板，包括： 被中止的模板 模板计算占用内存过高 模板计算时间过长 sql 执行时间过长 模板数据集数据行数大于建议限制行数 模板单元格数大于建议限制 单元格数 所用 JDK 版本存在 BUG 系统存在不合理的内存配置 磁盘空间不足 内存映射文件数量设置过低	1) 建议对详情中的模板进行性能排查，同时开启模板限制，配置合理的限制范围以避免模板性能导致宕机。 模板限制请参见： <a href="#">模板限制</a> 2) 如果当前系统堆内内存配置小于推荐配置时，额外建议使用系统检查功能对应用进行诊断，修改堆内内存至推荐值。 系统检查请参见： <a href="#">系统检查</a> 使用系统检查功能对应用进行诊断，并参照推荐值修改内存配置 系统检查请参见： <a href="#">系统检查</a> 请检查磁盘空间并清理不必要文件 磁盘扩容请参见： <a href="#">磁盘扩容</a> 使用系统检查功能对应用进行诊断，并参照推荐值修改内存映射文件数量配置

	系统检查请参见： <a href="#">系统检查</a>
	建议更新升级至最新的小版本
当前所用版本存在宕机 BUG	FR 更新升级请参见： <a href="#">FineReport 版本说明</a>
	BI 更新升级请参见： <a href="#">FineBI 版本说明</a>
未知原因导致应用崩溃	建议上传云端运维日志或联系技术支持进行反馈
	通过 SSH 方式启动应用会导致 SSH 退出时应用也被退出
SSH 退出导致应用退出	建议使用其他命令行远程工具如 secureCRT 或配置其他应用服务器自启动等方式
	例如： <a href="#">Web 应用服务器开机自启动</a>
	建议使用抽数缓存、SQL 优化、数据预处理等方式提升取数性能
数据源取数耗时过长导致线程阻塞	性能优化请参见： <a href="#">性能优化指导手册</a>
	请调高日志输出级别减少日志输出或检查磁盘剩余空间是否不足
日志输出导致线程阻塞	日志级别请参见： <a href="#">日志简介</a>
	如果使用的堆内内存大于等于 64G 时：请将堆内内存调整至小于 64g 的数值
系统内存释放时间过长	如果使用的堆内内存小于 64G 时：建议更换使用更高性能的 CPU

### 3. 内存堆栈导出记录

记录自动生成 DUMP 文件记录。

包括导出的项目、导出内容、导出开始时间、导出时长、触发类型、是否成功、如果失败，失败的原因等。如下图所示：

注：仅读取最近一个月的宕机处理记录。

运维平台

宕机自动处理

宕机处理策略

内存堆栈导出记录

项目	节点名	导出内容	导出开始时间	导出时长	触发类型	是否成功	失败原因
fanruan1	c5e2e1d1...	jmapHisto	2023-10-1...	2s	运维平台手动触发	是	null
fanruan1	ec017397...	jmapHisto	2023-10-1...	3s	运维平台手动触发	是	null
fanruan1	c5e2e1d1...	jstack	2023-10-1...	7s	运维平台手动触发	是	null
fanruan1	ec017397...	jstack	2023-10-1...	7s	运维平台手动触发	是	null
我的fr11	82	jmap	2023-10-1...	60s	连续FullGC后依旧负载较...	是	null
我的fr11	82	jstack	2023-10-1...	17s	连续FullGC后依旧负载较...	是	null
我的fr11	82	jmapHisto	2023-10-1...	43s	连续FullGC后依旧负载较...	是	null
我的fr11	82	jstack	2023-10-1...	17s	连续FullGC后依旧负载较...	是	null
fanruan1	ec017397...	jmap	2023-10-1...	53s	连续FullGC后依旧负载较...	是	null
fanruan1	ec017397...	jstack	2023-10-1...	17s	连续FullGC后依旧负载较...	是	null

4. 服务器重启记录

记录服务器重启的记录。

包括重启的项目、开始时间、时长、是否成功、如果失败，失败的原因等。如下图所示：

注：仅读取最近一个月的宕机处理记录。

运维平台

宕机自动处理

宕机处理策略

服务器重启记录

项目	节点名	重启开始时间	重启时长	触发类型	是否成功	失败原因
我的fr11	82	2023-10-1...	28s	连续FullGC后依旧负载较...	是	null
fanruan1	ec017397...	2023-10-1...	0s	连续FullGC后依旧负载较...	是	null
fanruan1	ec017397...	2023-10-1...	0s	连续FullGC后依旧负载较...	是	null
fanruan1	ec017397...	2023-10-1...	0s	连续FullGC后依旧负载较...	是	null
我的fr11	12	2023-10-1...	30s	连续FullGC后依旧负载较...	是	null
我的fr11	12	2023-10-1...	14s	连续FullGC后依旧负载较...	是	null
我的fr11	12	2023-10-0...	23s	连续FullGC后依旧负载较...	是	null
我的fr11	12	2023-10-0...	24s	连续FullGC后依旧负载较...	是	null

## 7.4 宕机处理策略

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.5.0	-
V1.5.7	「自动导出内存堆栈」功能拆解到「性能堆栈」中，详情请参见： <a href="#">性能堆栈</a>

#### 1.2 应用场景

帆软工程部署在服务器上后，会因为各种原因导致宕机。宕机后若直接重启没有抓取 DUMP 文件，很难定位到真实原因。

从而无法快速解决问题和做一些防止再发生的预防措施，造成的时间和精力损失无法估计，对服务器运维非常不友好。

#### 1.3 功能简介

运维平台提供「宕机处理策略」功能。通过此功能，运维平台可为项目绑定的 FineBI/FineReport 等工程自动生成 DUMP 文件、同时可重新启动工程。

发生宕机后，通过分析生成的 DUMP 文件，可以快速定位宕机原因，及时处理宕机问题和做一些有效的预防措施，为用户增加一道保险。

### 2. 宕机自动处理

管理员登录运维平台，点击「业务工具>宕机自动处理>宕机处理策略」，选择运维项目，即可为该项目配置宕机处理策略。

用户可以在「宕机处理策略」位置做一些配置设置，可设置的项为：宕机自动处理、自动导出内存堆栈、宕机自动重启、进程自动恢复、端口设置。

注 1：若项目不可用，则无法为其配置「宕机处理策略」。

注 2：容器化部署的工程不支持「进程自动恢复」功能。因为容器化部署的工程，可由容器化提供进程自动恢复功能，所以无需运维平台提供该功能。

注 3：容器化部署的工程不支持「端口设置」功能。



## 2.1 宕机自动处理

### 使用说明：

开启该按钮，才支持进行下方功能的配置，否则全部不可用。

此开关默认开启。

### 使用前提：

在工作时段（6:00-23:00），当应用主进程关闭 5 分钟后，宕机自动处理也跟随关闭。

在非工作时段（0:00-6:00, 23:00-24:00），应用主进程关闭，宕机自动处理不跟随关闭。

## 2.2 宕机自动重启

### 使用说明：

开启后，当高负载导致宕机时，会自动重启工程。

### 使用前提：

1) 开启「宕机自动处理」

- 若「宕机自动处理」未开启，此项灰化处理，不能编辑修改。
- 若「宕机自动处理」开启，此项默认关闭。

2) 系统情况

开关开启时，会对当前系统情况做检测。检测是否为 Windows 操作系统，是否为 Windows 服务形式。

- 若系统为 Windows 操作系统且为 Windows 服务形式，弹窗提示：该功能暂不支持当前系统。如下图所示：

点击确定和关闭按钮后弹窗关闭，开关不开启。



- 若系统为 Windows 操作系统且非 Windows 服务形式，弹窗提示：当前系统可能会存在重启失败。如下图所示：

点击确认或关闭按钮后关闭弹窗，开关开启。



2.3 进程自动恢复

注：容器化部署的运维项目，不支持该设置。因为容器化部署的工程，可由容器化提供进程自动恢复功能，所以无需运维平台提供该功能。

使用说明：

开启后，管理员可设置功能生效时间段。在指定时间段内，当应用进程消失时，会自动重启工程。

使用前提：

1) 开启「宕机自动处理」

- 若「宕机自动处理」未开启，此项灰化处理，不能编辑修改。
- 若「宕机自动处理」开启，此项默认关闭。

2) 系统情况

开关开启时，会对当前系统情况做检测。检测是否为 Windows 操作系统，是否为 Windows 服务形式。

- 若系统为 Windows 操作系统且为 Windows 服务形式，弹窗提示：该功能暂不支持当前系统。如下图所示：

点击确定和关闭按钮后弹窗关闭，开关不开启。



- 若系统为 Windows 操作系统且非 Windows 服务形式，弹窗提示：当前系统可能会存在重启失败。如下图所示：

点击确认或关闭按钮后关闭弹窗，开关开启。



## 2.4 端口设置

注：容器化部署的运维项目，不支持该设置。

### 使用说明：

支持用户设置宕机处理工具占用的端口，默认的端口为 12100。

端口号范围要在 1024 到 65535 之间，否则「宕机处理工具」无法启动，「宕机处理」页面无法打开。

### **使用前提：**

- 若「宕机自动处理」未开启，此项灰化处理，不能编辑修改。
- 若「宕机自动处理」开启，默认的端口为 12100。



# 7.5 资源清理

## 1. 概述

### 1.1 版本

运维平台版本	功能变更
V1.5.5	-

### 1.2 应用场景

随着工程运行时间的不断增加，工程占用的内存也不断增大，用户往往焦虑哪些文件可以清理呢？

- 1) 很多处于中间形态的最终没有用处的系统文件，在生成的时候没有考虑到内存等因素，会不断生成而不会被清理掉。
- 2) 随着工程系统中的垃圾文件越来越多，不能及时清理，会影响系统运行状态，甚至导致系统宕机。
- 3) 用户希望工程能不占用太大的空间，节省磁盘资源，但是不知道哪些可以清理哪些不能清理。
- 4) 手动清理日志等文件太耗费时间，用户希望降低运维时间成本，可以设置自动清理的机制。

### 1.3 功能简介

运维平台支持「资源清理」功能。用户可以安全便捷的清理磁盘，不造成业务文件丢失风险，预防磁盘问题带来的宕机问题

- 1) 提供磁盘清理功能，高效推荐可清理文件。
- 2) 部分文件可设置自动清理逻辑。
- 3) 能够解决可高效清理磁盘节省资源和预防磁盘宕机的问题。

## 2. 清理逻辑

磁盘清理功能包括自动清理和手动清理两部分。

- 自动清理无前端设置项，直接在后台进行清理。
- 手动清理可参见第三章。

文件夹/文件	用途	自动清理	手动清理
备份文件	1) 文件路径：  默认路径为%FR_HOME%\webapps\webroot\backup  若自定义备份文件路径，则扫描自定义路径	不支持自动清理	文件层级直接展示到具体的日期（日级别），用户手动选择需要清理的范围即可
	2) 文件用途：  工程备份文件		推荐清理 3 个月之外的文件  3 个月之内的文件不显示
	3) 清理级别：  谨慎清理		
	1) 文件路径：  %FR_HOME%\webapps\webroot\WEB-INF\schedule		
schedule	2) 文件用途：  存储定时调度任务产生的中间文件	不支持自动清理	文件层级直接展示到具体的日期（日级别），用户手动选择需要清理的范围即可
	3) 可清理范围：  保留最近一年的文件，其余均可清理		推荐清理 12 个月之外的文件  12 个月之内的文件不显示
	4) 清理级别：  建议清理		
	1) 文件路径：  %FR_HOME%\webapps\webroot\WEB-INF\assets\vcs	保留最近 30 天的文件，其余文件清理	文件不分日期展示, 30 天前的所有历史版本文件全部合并展示，勾选后将全部清理
/vcs	2) 文件用途：  版本管理文件	理	推荐清理 30 天之外的文件  30 天之内的文件不显示
	3) 可清理范围：		

	保留最近 30 天的文件,	
	其余均可清理	
	4) 清理级别:	
	建议清理	
	1) 文件路径:	
	%FR_HOME%\webapps\webroot\WEB-INF\treasures	文件层级直接展示到具体的日期 (日期级别), 用户手动选择需要清理的范围即可
treasures	2) 文件用途:	保留最近 12 个月的文件, 其余文件清理
云端运维日志打包文件		推荐清理 12 个月之外的文件
	3) 清理级别:	12 个月之内的文件不显示
	建议清理	
	1) 文件路径:	
	%FR_HOME%\logs\fanruan.log	
fanruan.log	2) 文件用途:	保留最近 3 个月的数据, 其余数据清理
帆软工程运行日志		不支持手动清理
	3) 清理级别:	
	谨慎清理	
	1) 文件路径:	
	扫描系统导出的没有及时清理的 dump 文件	文件层级直接展示到具体的日期 (日期级别), 用户手动选择需要清理的范围即可
dump	2) 文件用途:	保留最近 1 周的文件, 其余文件清理
dump 文件		推荐清理 1 周之外的文件
	3) 清理级别:	1 周之内的文件不显示
	建议清理	
	1) 文件路径:	
Catalina.log	%TOMCAT_HOME%\logs\Catalina.log	文件层级直接展示到具体的日期 (日期级别), 用户手动选择需要清理的范围即可

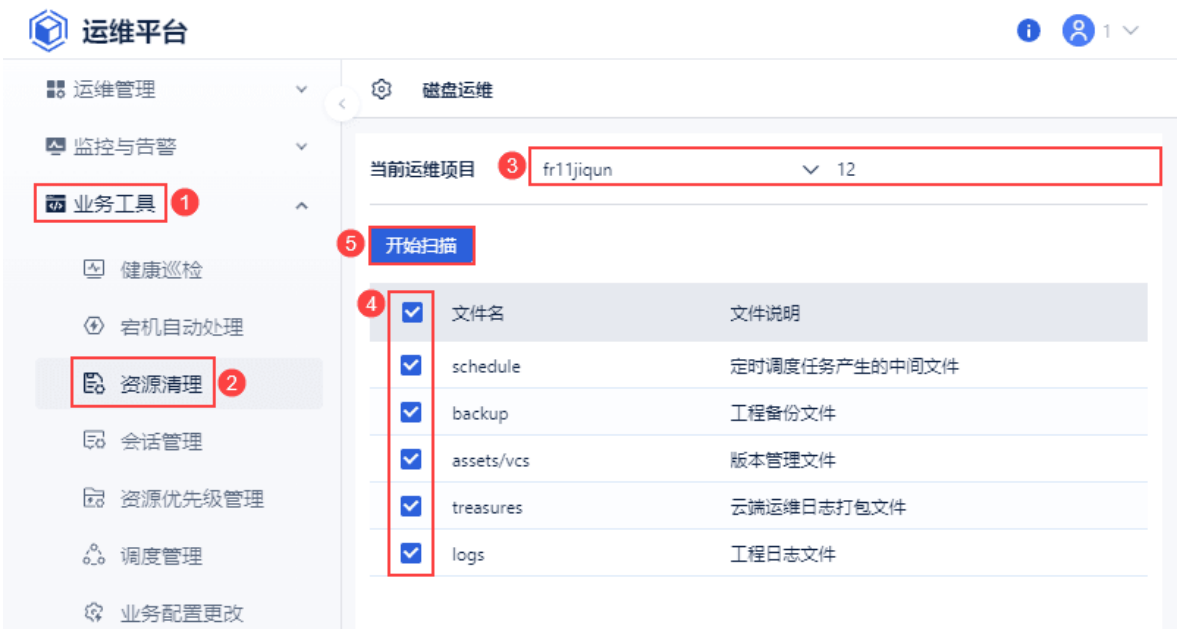
2) 文件用途:		围即可
Tomcat 运行日志		推荐清理 1 周之外的文件
3) 清理级别:		1 周之内的文件不显示
谨慎清理		
4) 注意事项		
若 Tomcat 下该日志未		
分卷, 则不清理		
1) 文件路径:		
%TOMCAT_HOME%\logs\ Catalina.out		
2) 文件用途:		
Tomcat 的标准输出和		文件层级直接展示到具体的日期 (日
标准出错日志		级别), 用户手动选择需要清理的范
Catalin a.out	3) 清理级别:	保留最近 1 周的文 件, 其余文件清理
	谨慎清理	围即可
		推荐清理 1 周之外的文件
	4) 注意事项	1 周之内的文件不显示
若 Tomcat 下该日志未		
分卷, 则不清理		
若 Windows 系统下不		
存在该文件, 则不清理		

3. 手动清理

3.1 扫描磁盘

管理员登录运维平台, 点击「业务工具>资源清理」, 选择需要进行磁盘运维的项目工  
程和节点。

选择需要扫描的文件类型, 点击「开始扫描」, 立即开始扫描可以手动清理的冗余文  
件。如下图所示:



在扫描过程中，若点击「停止扫描」，则结束扫描，返回已扫描出的冗余文件。如下图所示：



扫描完成后：

- 1) 会告知用户可清理的总文件大小。
- 2) 用户可以树层级展开查看可清理的文件。
- 3) 对于单个文件，可查看文件大小。
- 4) 点击「推荐选项」，即可自动勾选所有建议清理的文件。

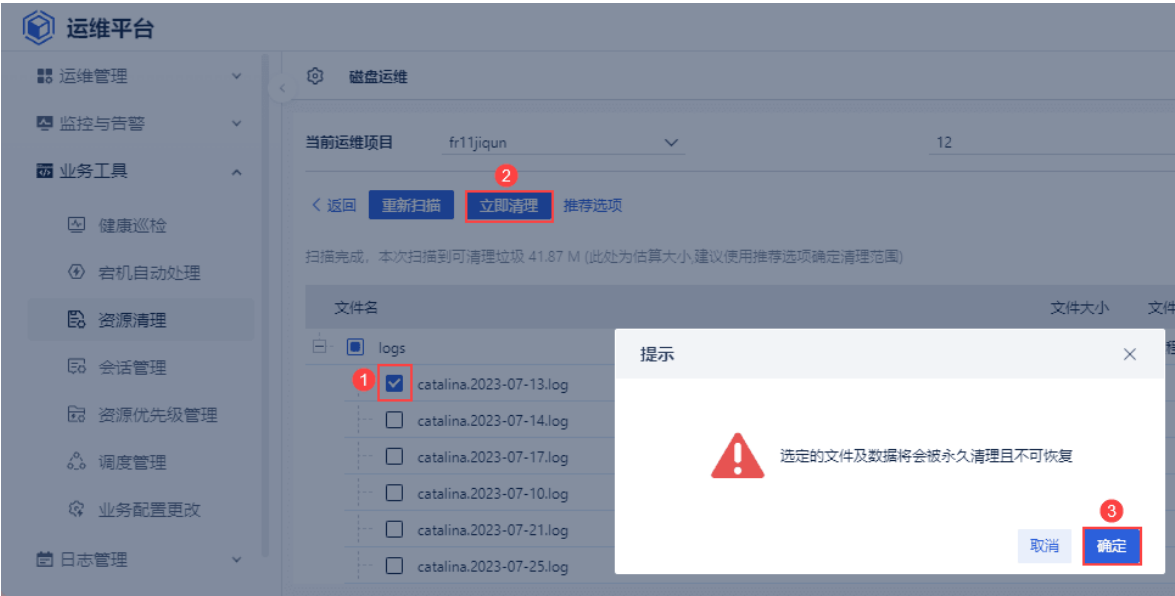
5) 点击「重新扫描」，即可重新开始扫描需要清理的文件。



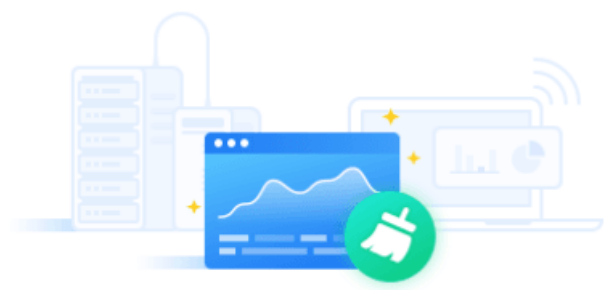
### 3.2 清理文件

在扫描完成的界面，选择需要清理的磁盘文件，点击「立即清理」，跳出弹窗提示「选定的文件及数据将会被永久清理且不可恢复」。

点击「确定」，立刻开始清理需要清理的磁盘文件。



在清理过程中，若点击「停止清理」，则中断清理，已清理的文件无法恢复。如下图所示：



文件正在清理中，请稍后... [停止清理](#)



清理完成后，会告知用户「清理完成，本次清理文件大小为 XXXM」，如下图所示：



清理完成，本次清理文件大小为 207.39 M

[返回](#)

4. 自动清理

管理员登录运维平台， 点击「业务工具>资源清理>全局设置」， 开启「自动清理」。

支持设置自动清理时间间隔， 开启后将按照间隔在晚上 2 点半对支持自动清理的内容进行自动检测及清理。如下图所示：





## 7.6 会话管理

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.5.5	-

#### 1.2 功能简介

用户在帆软应用中，打开一张目录中的模板，即为一个会话。

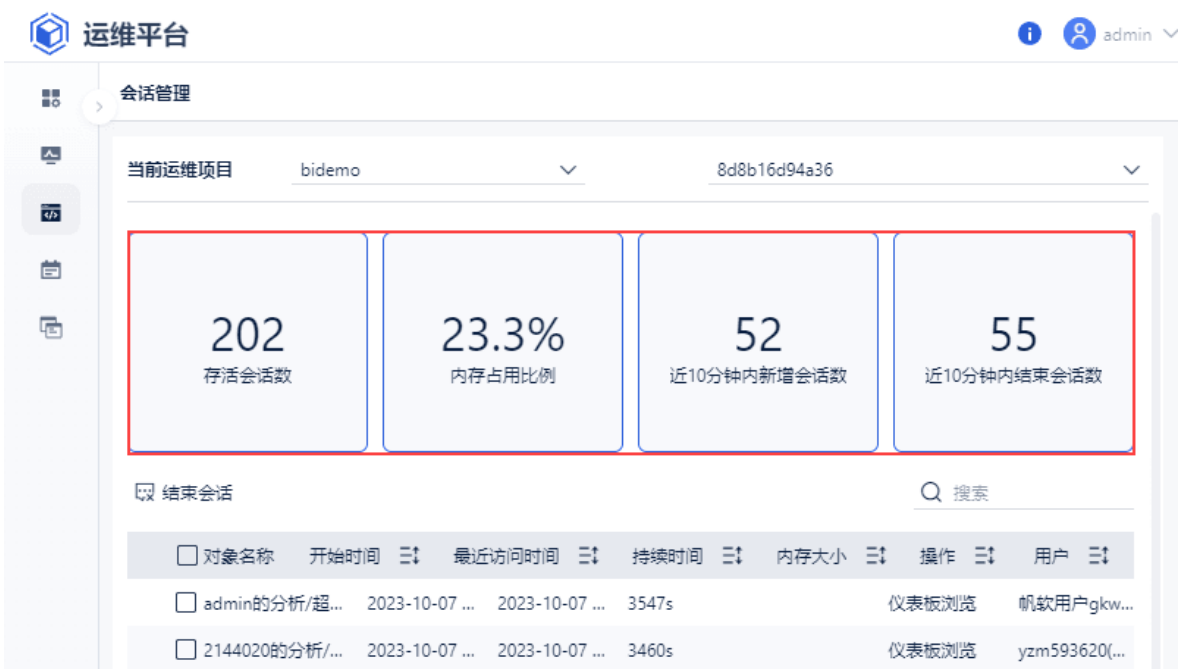
管理员可通过「会话管理」查看应用中会话占用情况，并支持对于异常会话手动进行中止。



### 2. 内存会话

内存会话界面包含存活会话数、内存占用比例、近 10 分钟内新增会话数、近 10 分钟内结束会话数。

- 存活会话数：还存活的会话数量。
- 存活会话内存占比：表示已用内存占比。
- 近 10 分钟内新增会话数：字面意思。
- 近 10 分钟内结束会话数：字面意思。



### 3. 结束会话

支持选中会话，手动强制杀死 Session 。勾选需要结束的会话，点击「结束会话」即可。如下图所示：

- 对象名称：会话对应的模板位置和名称。
- 开始时间：内存会话的开始时间，默认按照「开始时间」升序排列。
- 最近访问时间：用户最新对当前 session 做刷新操作的时间。
- 持续时间：内存会话存活的时长，单位 S。
- 内存大小：该会话占用的内存，单位 kb。
- 操作：用户对该会话进行的操作。
- 用户：发起该会话的用户。



## 7.7 资源优先级管理

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.5.5	-

#### 1.2 应用场景

一个系统中存在很多模板，但是业务的优先级有先有后。部分重要业务的资源需要优先保证。

一个系统中存在很多用户，但部分用户由于业务特殊性，例如长期面向客户，需要优先保证该用户的业务资源。

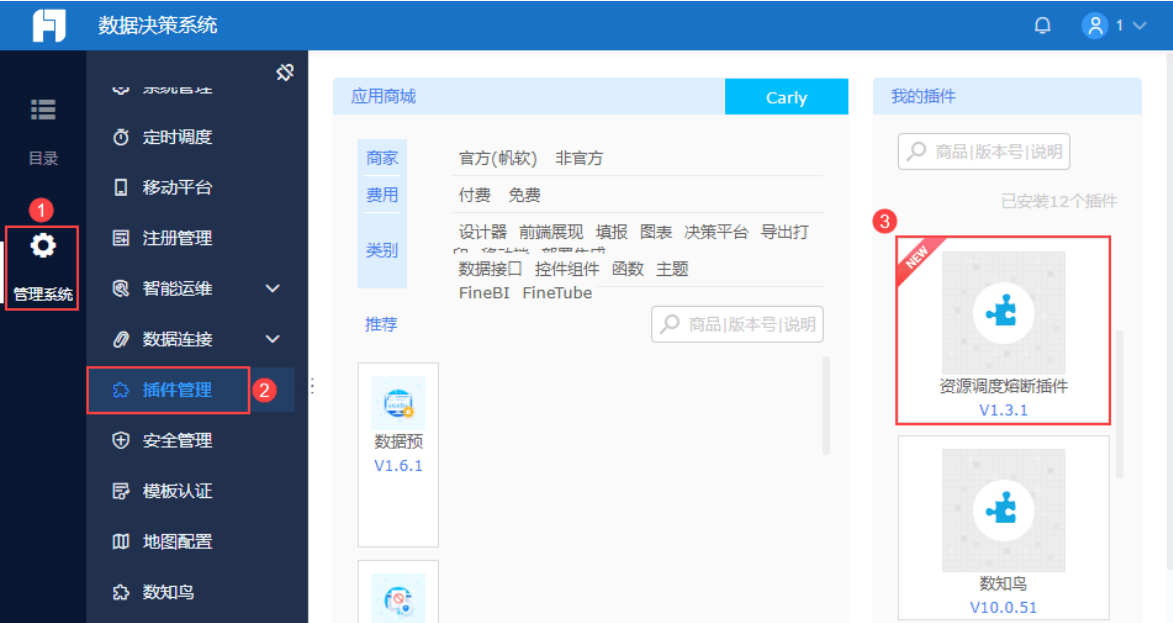
### 2. 使用前提

资源优先级功能依赖「资源调度熔断插件」。

管理员登录待配置资源优先级的运维项目，点击「管理系统>插件管理」，查看是否已安装资源调度熔断插件。

如未安装该插件：

- 点击下载插件：[资源调度熔断插件](#)
- 设计器插件安装方法参照：[设计器插件管理](#)
- 服务器安装插件方法参照：[服务器插件管理](#)



### 3. 设置资源优先级

#### 3.1 开启资源优先级

管理员登录运维平台，点击「业务工具>资源优先级管理」。选择需要配置优先级的项目，开启「启动资源优先级」按钮。

开启资源优先级功能后，支持按照模板、用户等维度定义不同优先级来更合理的调度资源。



#### 3.2 配置高优先级模板

开启资源优先级功能后，选择「高优先级模板」Tab，点击「编辑」按钮，支持选择「平台目录」、「模板文件」、BI「公共链接」作为优先级资源。如下图所示：

当线程队列中同时有高优先级和低优先级的业务存在时，优先放行高优先级业务再放行低优先级业务。

且资源紧张时会优先保障高优先级业务，低优先级业务直接熔断或限行。



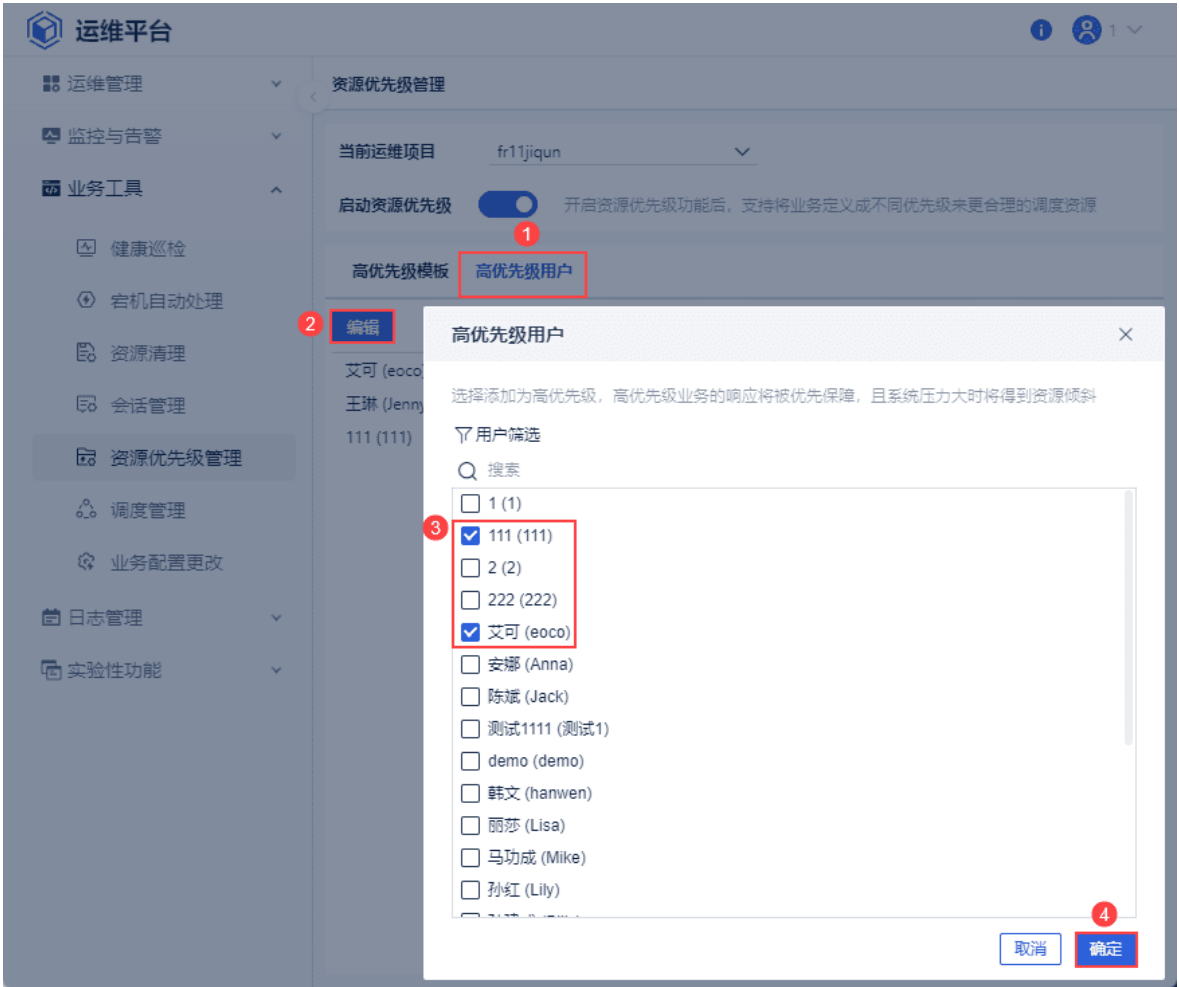
### 3.3 配置高优先级用户

开启资源优先级功能后，选择「高优先级用户」Tab，点击「编辑」按钮，支持选择用户作为优先级资源。如下图所示：

当高优先级和低优先级的用户同时发送请求时，优先满足高优先级用户再放行低优先级用户。

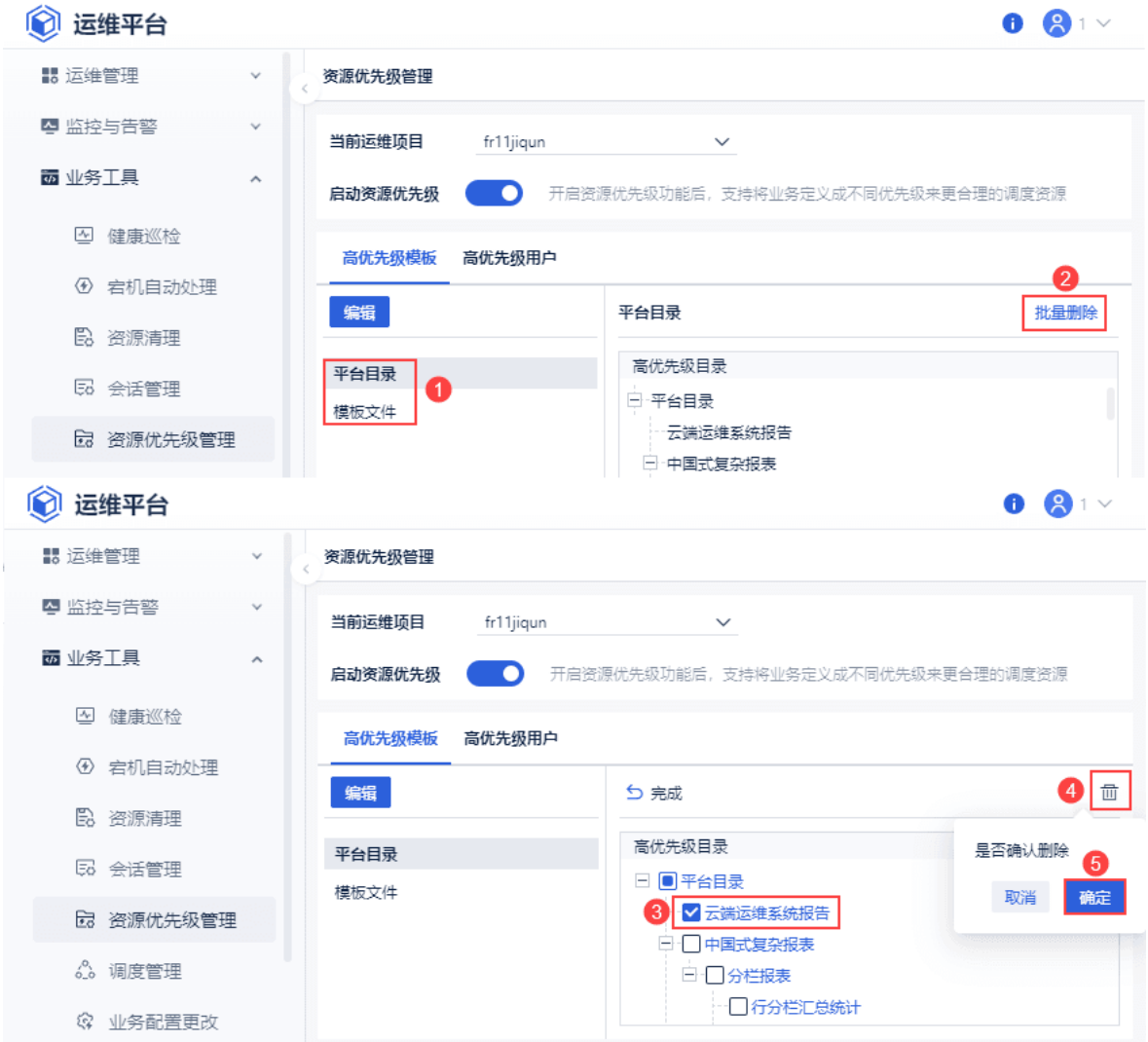
且资源紧张时会优先保障高优先级用户，低优先级用户直接熔断或限行。

注：可点击「用户筛选」按钮，根据「部门」/「角色」筛选用户。



#### 4. 删除资源优先级

对于优先级模板/用户，可批量删除。点击「批量删除」按钮，选择需要删除优先级的资源，确认删除，如下图所示：





## 7.8 调度管理

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.5.5	-

#### 1.2 应用场景

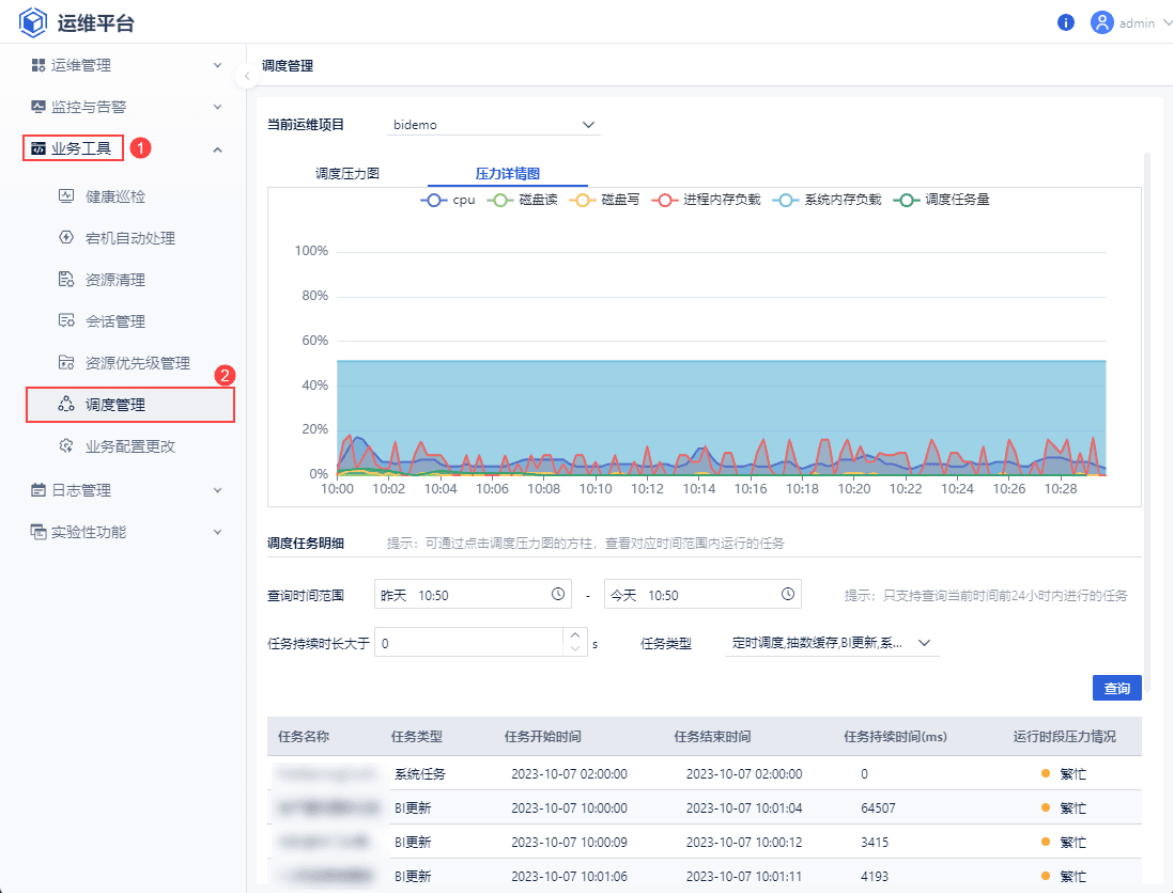
在 FineReport、FineBI 中，存在各种各样的后台调度任务，例如 BI 表更新、定时调度任务、系统任务等。

这些调度缺乏系统管理：

- 调度任务堆积在同一时间段执行，系统压力大，甚至导致调度失败。
- 系统任务与业务任务时间冲突，影响正常的业务运行。
- FineBI 抽取表更新压力大，用户不清楚如何优化抽取设置。

#### 1.3 功能简介

运维平台提供「调度管理」功能，帮助用户了解系统中的调度任务和系统压力，促进任务配置优化。如下图所示：



## 2. 调度压力一览

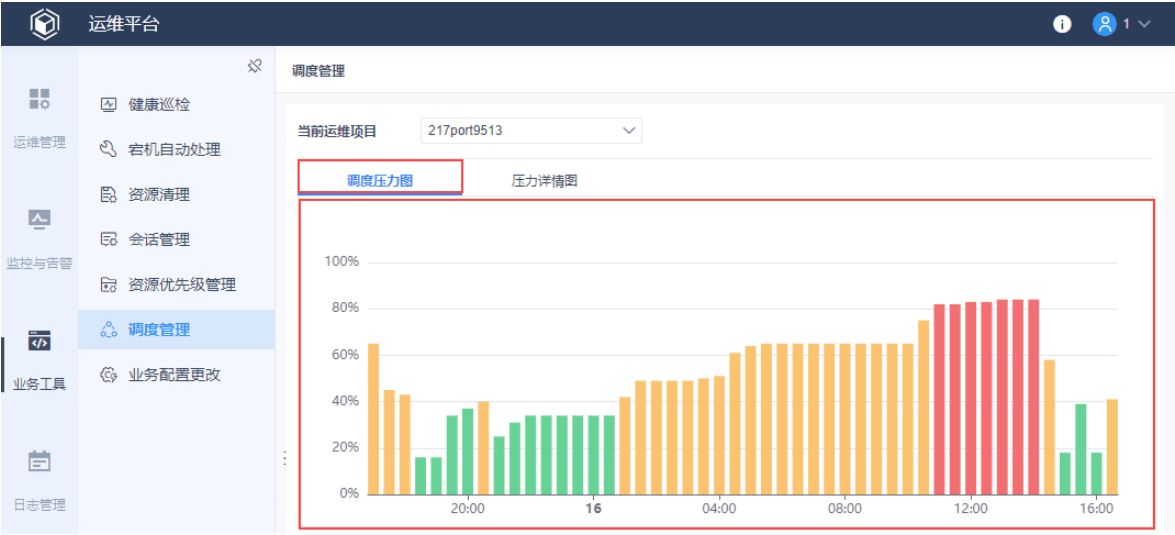
调度压力一览，是根据系统过去 24 小时调度的综合执行情况，评估出的系统调度压力分值。包括：调度压力图和压力详情图。

### 2.1 功能简介

#### 1) 调度压力图

调度压力图展示过去 24 小时中，每半小时的计算压力总分，以柱形图展示。

柱子颜色代表不同的调度压力：0 ~ 39%空闲（显示绿色）、40% ~ 79%繁忙（显示黄色）、80% ~ 紧张（显示红色）



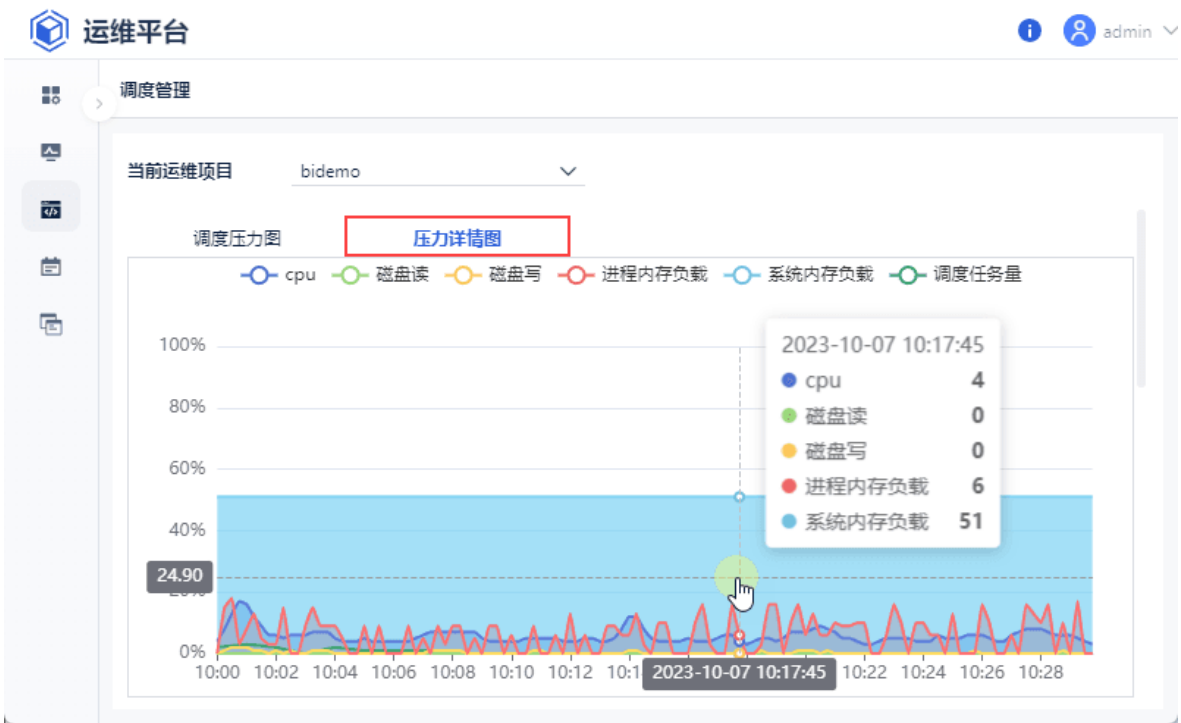
#### 2) 压力详情图

压力详情图展示过去半小时中，每 15 秒的各项压力维度的分值，以面积图展示。

点击调度压力图的方柱，可与压力详情图联动，切换压力详情图的展示时间段。

压力维度包括：CPU、磁盘读、磁盘写、进程内存负载、系统内存负载、调度任务量。

重合面积越大，表示系统压力越大。



## 2.2 计算逻辑

调度压力计算评估维度包括：任务量维度和压力维度，几者综合得出调度压力。

注：集群环境下，不对单个节点进行分析，而是综合所有节点，来计算系统的调度压力。

### 1) 任务量维度

对应时间段内的调度任务量（注意：即使）

### 2) 压力维度

- CPU 高占用的持续时间占总时间的比例
- 磁盘高读写的持续时间占总时间的比例
- 内存高占用的持续时间占总时间的比例

## 3. 调度任务明细

调度任务明细，以明细表形式展示调度任务明细，支持筛选查看不同任务。



1) 筛选维度

筛选维度	说明
查询时间	1) 支持查询过去 24 小时内的调度任务明细。 2) 默认展示过去一小时内的调度任务明细。 3) 调度任务明细可与调度压力图联动。点击调度压力图的方柱，调度任务明细切换查询时间范围，查看对应时间范围内运行的任务。
任务持续时长	筛选持续时长大于 x 秒的任务
任务类型	支持自定义勾选需要查看的任务类型 包括：定时调度、抽数缓存、BI 更新、系统任务

2) 展示内容

展示内容	说明
任务名称	任务名称，包括： 定时调度任务的名称、BI 更新任务的名称、抽数缓存任务的名称、后台调度任务的名称
任务类型	任务类型，包括： 定时调度、抽数缓存、BI 更新、系统任务

任务开始时间	调度任务开始的时间
任务结束时间	调度任务结束的时间 若任务未完成，则为空
任务持续时间	调度任务从开始到结束，花费的时间 若任务未完成，则为空
运行时段压力情况	调度任务所涉及时间段的最高压力情况 绿色：空闲；黄色：繁忙；红色：紧张

# 7.9 业务配置更改

## 1. 概述

### 1.1 版本

运维平台版本	功能变更
V1.5.5	-

### 1.2 应用场景

帆软应用中，有些配置信息记录配置库 FineDB 的 fine\_conf\_entity 表中，只能通过修改表字段来进行更改。

用户可通过「业务配置更改」方式安全地修改相关配置。

## 2. 参数配置方法

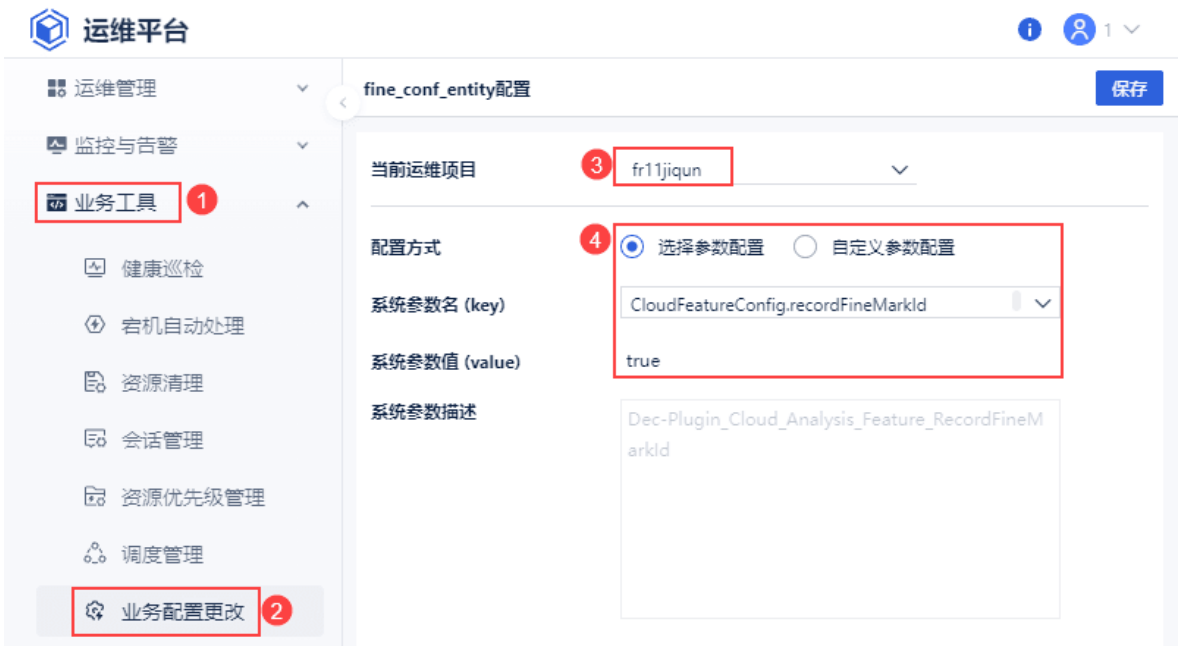
管理员登录运维平台，点击「业务工具>业务配置更改」，选择需要修改配置的项目。

配置方法支持以下两种：

- 选择参数配置：**系统参数名（key）下拉框中罗列了支持配置的 fine\_conf\_entity 参数，用户可直接在下拉框中选择参数，配置参数值（value）。
- 自定义参数配置：**用户可自行输入支持配置的 fine\_conf\_entity 参数名（key），并自动跳出参数值（value），用户可修改并保存参数值。

按照参数规则修改配置后，点击「保存」，跳出提示框：您修改的参数在重启后生效，请重启服务器。

重启帆软应用服务器后，配置生效。



3. 参数配置内容

可修改的 fine\_conf\_entity 中的配置参数和参数值如下表所示：

参数名	参数描述	参数值
BackupConfig.customKeyLength	备份 conf 表，自定义 id 长度	参数值需为正整型 默认值为 1500
BackupConfig.customValueLength	备份 conf 表，自定义 value 长度	参数值需为正整型 默认值为 65536
CacheConfig.maxMemSize	共享数据集缓存空间上限	参数值需为非负整型 默认值为 4194304
ClusterTimeMonitorConfig.permittCount	集群单次时间误差检测任务中，多次误差分析中最小成功次数	参数值需为正整型 默认值为 1
ClusterTimeMonitorConfig.permittError	集群单次时间误差检测任务中，允许时间误差值，单位 ms	参数值需为正长整型 默认值为 10000
ClusterTimeMonitorConfig.sumCount	集群单次时间误差检测任务中，尝试误差分析的总次数	参数值需为正整型 默认值为 3
ConfigVersion.versionOrdinal	配置版本号	默认值为当前版本号
CustomConfig.absPositionOffset	悬浮元素是否以绝对位置展示（不受单元格扩展的影响）	参数值需为布尔型，默认为 false false：不以绝对位置展示（不受单元格扩展的影响） true：以绝对位置展示（受单元格

		扩展的影响)
		参数值如下，默认为 case_ignore
CustomConfig.columnMatchRuleForWidgetValue	数据字段是否区分大小写	case_ignore: 不区分大小写
		case_sensitive: 区分大小写
	定义填报提交,校验请	参数值如下，默认为 xml
CustomConfig.dataTransType	求中前端模板数据的传输形式	xml: 以 xml 字符串格式传输
		json: 以 json 字符串格式传输
CustomConfig.excelAdjustHeight	Excel 导出行高系数	参数值需为非负单精度浮点型
		默认值为 20.1
CustomConfig.excelAdjustWidth	Excel 导出列宽系数	参数值需为非负单精度浮点型
		默认值为 34.742
		参数值需为布尔型，默认为 false
		false: Excel 导出时，不导出完整
CustomConfig.exportFullImg4Excel	Excel 导出的时候是否导出完整图片	图片
		true: Excel 导出时，导出完整图片
		参数值需为布尔型，默认为 false
CustomConfig.printWidget	是否导出/打印控件(目前支持打印复选框与复选框组控件)	false: 不导出/打印控件
		true: 导出/打印控件
		参数值需为布尔型，默认为 true
CustomConfig.screenScaleEnabled	控制设计器缩放	false: 关闭设计器缩放
		true: 开启设计器缩放
CustomConfig.wordAdjustHeight	Word 导出用于调整行高的数值，防止内容高度在某些临界值时出现空白页	参数值需为非负整型
		默认值为 2
		参数值需为布尔型，默认为 false
FSConfig.authorizeAttr.postAuthority	按职位分配权限的开关	false: 关闭按职位分配权限
		true: 开启按职位分配权限



FSConfig.loginConfig.forceRedirectAfterLogin	登录后强制跳转至：  http://ip:port/webroot/decision/	参数值需为布尔型，默认为 false  false：登录后不强制跳转  true：登录后强制跳转
FileServerMonitorConfig.messageInterval	ftp 异常消息通知频率，单位 h	参数值需为正长整型  默认值为 1
FineFocusPointConfig.allowRecordFitInfo	是否允许记录自适应埋点	参数值需为布尔型，默认为 true  false：不允许记录自适应埋点  true：允许记录自适应埋点
GcConfig.gcThreshold	模板版本控制存储优化条件阈值，单位为文件个数	参数值需为非负整型  默认值为 300
HotBackConf.master	热备 master 节点配置	主节点的 hostname
HotBackConf.slave	备节点配置	备节点的 hostname  参数值需为长整型
IntelliLogConfig.expired	模板缓存失效时间	默认值为 18000
IntelliReleaseConfig.defaultCellCount	待杀会话格子数的默认基数	参数值需为长整型  默认值为 1000000
IntelliReleaseConfig.highInterruptAgainRate	高负载时第二次释放超过（默认值格子数*此比例）的会话	参数值需为非负双精度浮点型  默认值为 0.6
IntelliReleaseConfig.highInterruptRate	高负载时第一次释放超过（默认值格子数*此比例）的会话	参数值需为非负双精度浮点型  默认值为 0.8
IntelliReleaseConfig.highKillRate	高负载第一次释放的会话比例超过此值不触发第二次	参数值需为非负双精度浮点型  默认值为 0.1
IntelliReleaseConfig.interruptRate	待杀会话格子数的默认比例	参数值需为非负双精度浮点型  默认值为 0.3
IntelliReleaseConfig.jvmSupport	当前版本 jdk 是否支持智能释放	参数值需为布尔型，默认为 true  false：当前版本 jdk 不支持智能释放  true：当前版本 jdk 支持智能释放
IntelliReleaseConfig.releaseSessionInterval	释放触发的时间间隔	参数值需为正整型  默认值为 20
IntelliReleaseConfig.terribleInterruptAgainRate	超高负载时第二次释放超过（默认值格子数*此比例）的会话	参数值需为非负双精度浮点型  默认为 0.4
IntelliReleaseConfig.terribleInterrupt	超高负载时第一次释放	参数值需为非负双精度浮点型

New Help		
uptRate	超过(默认值格子数*此比例)的会话	默认为 0.6
IntelliReleaseConfig.terribleKillRate	超高负载第一次释放的会话比例超过此值不触发第二次	参数值需为非负双精度浮点型 默认为 0.2
IntelliReleaseConfig.waitInLineRate	每个会话在高负载有几率排队	参数值需为双精度浮点型 默认为 0.9
JarConsistenceConfig.messageInterval	jar 不一致异常消息通知频率, 单位 h	参数值需为正长整型 默认值为 6
LoadConfig.majorExtremeTerribleThreshold	majorgc 后, 残留在老年代对象大小与老年代大小的比值超过这个值算是极限负载	参数值需为非负双精度浮点型 默认为 0.95
LoadConfig.majorHighThreshold	majorgc 后, 残留在老年代对象大小与老年代大小的比值超过这个值算是高负载	参数值需为非负双精度浮点型 默认为 0.7
LoadConfig.majorTerribleThreshold	majorgc 后, 残留在老年代对象大小与老年代大小的比值超过这个值算是超高负载	参数值需为非负双精度浮点型 默认为 0.85
LoadConfig.minorHighThreshold	minorgc 后, 晋升到老年代的对象速率与 eden 区的比值大小超过这个值算是高负载	参数值需为非负双精度浮点型 默认为 0.04
LoadConfig.minorMidThreshold	minorgc 后, 晋升到老年代的对象速率与 eden 区的比值大小超过这个值算是中等负载	参数值需为非负双精度浮点型 默认为 0.02
LoadConfig.minorTerribleThreshold	minorgc 后, 晋升到老年代的对象速率与 eden 区的比值大小超过这个值算是超高负载	参数值需为非负双精度浮点型 默认为 0.06
		参数值为非空字符串
MobileConfig.appMsgProxy	App 消息代理	默认值为__EMPTY__
注: 仅对移动端生效		代理服务器地址格式为:  http://ip:port/mobile/push/message/send
MobileConfig.disableEncryptPassword	控制移动端登录的密码是否加密	参数值需为布尔型, 默认为 false false: 移动端登录的密码不加密 true: 移动端登录的密码加密
注: 仅对移动端生效		
MobileConfig.persistedAttach	是否在启动服务器的时候持久化定时调度	参数值需为布尔型 默认第一次启动为 false, 后为 true

	图片	false：在启动服务器的时候持久化定时调度图片
	注：仅对移动端生效	true：在启动服务器的时候不持久化定时调度图片
		参数值需为布尔型，默认为 true
MobileConfig.productPlan	是否加入开发者计划	false：不加入开发者计划
	注：仅对移动端生效	true：加入开发者计划
		参数值需为整型
		默认值为-1
		值小于 0：填报请求、表单数据、请求组件数据请求永不超时
		值大于 0（值为 n）：超时时间为 n 秒；超过 ns 后，请求（填报请求、表单数据、请求组件数据）没有返回结果则取消加载；其中填报请求、表单数据超时会弹出提示「请求超时，请联系管理员增加请求」
MobileConfig.templateFetchTimeout	模板超时时间	参数值如下，默认值为 default
	注：仅对移动端生效	default：手机展示手机，pad 展示 pad
MobileConfig.requiredDisplayDeviceType	设置移动端设备展示类型	phone：手机 & pad 都展示手机效果
QuartzConfig.maxConnections	定时调度模块最大连接数（2019 年 8 月 16 之后版本支持）	参数值需为正整型 默认值为 50
QuartzConfig.threadCount	定时调度模块最大线程数（2019 年 8 月 16 之后版本支持）	参数值需为正整型 默认值为 100
RedisClusterConfig.maxConnections	Redis Cluster 连接池最大实例总数	参数值需为正整型 默认值为 200

New Help		
RedisClusterMonitorConfig.messageInterval	Redis 节点异常消息通知频率，单位 h	参数值需为正长整型 默认值为 6
RedisConfig.database	Redis 指定数据库	参数值需为整型 默认值为 0
RedisConfig.expireStateRemoveInterval	状态服务器过期 key 删除间隔时间，单位 ms	参数值需为正整型 默认值为 300000
RedisConfig.maxConnection	Redis 连接池最大实例总数	参数值需为正整型 默认值为 200
RemoteDesignConfig.avoidTempAuthValid	远程设计支持数字签名预览	参数值需为布尔型，默认为 true false：远程设计不支持数字签名
		true：远程设计支持数字签名预览
		预览
ScheduleSettingConfig.taskTimeout	定时任务超过时间，默认 5 分钟	参数值需为正长整型 默认值为 300000  参数值需为布尔型，默认为 false false：定时调度任务不开启监控
ScheduleSettingConfig.timeoutRemind	定时调度任务是否开启监控日志超时提醒	日志超时提醒  true：定时调度任务开启监控日志超时提醒
SecurityConfig.frontSeed	用于给前端加密提供密钥	参数值为 16 位大小写字母 初始是随机的 16 位大小写字母字符串，可自行修改
ServerConfig.cookiePath	cookie 路径  注：请在部署工程时配置。否则会造成浏览器存储多个 cookie，需要每个用户手动清除 cookie 才能登录成功	参数值需为非空字符串 初始值为/
ServerConfig.tokenFromCookie	开启后后台校验 token 时 可从 cookie 中取，解决 h	参数值需为布尔型，默认为 false

New Help		
	ttponly 下后台单点和跨域单点登录失败的问题	false: 后台校验 token 时不可从 cookie 中取  true: 后台校验 token 时可从 cookie 中取  参数值需为布尔型，默认为 false
ServerPreferenceConfig.useOptimizedUPM	配置是否开启新插件管理	false: 不开启新插件管理  true: 开启新插件管理  参数值需为布尔型，默认为 false  false: 不隐藏 system info 中的版本信息
SystemConfig.hideVersion	是否隐藏 system info 中的版本信息	true: 隐藏 system info 中的版本信息  参数值如下，默认为 success
SystemConfig.serverInit	平台是否初始化完成	success: 平台初始化完成  fail: 平台初始化未完成  参数值需为字符串
SystemOptimizationConfig.ClientMasterId	Web 集群有数据的节点	是 Web 集群, 指定的数据请求, 更新, 有数据的那台机器的 Id
SystemOptimizationConfig.biClusterMasterNodeHostName	BI 集群的主节点的 host name	参数值需为字符串  初始值为__EMPTY__  参数值需为非负整型
SystemOptimizationConfig.cacheHeap	缓存个数: 除去关联维表缓存和分页缓存的其他缓存	默认不配置时, 大数据集缓存 100 个, 小数据集缓存 10000 个  设置成 0, 表示缓存个数无限制, 修改后需要重启生效
SystemOptimizationConfig.cacheIdleSeconds	缓存过期时间	参数值需为正长整型  默认值为 300
SystemOptimizationConfig.cacheLargeMemPercent	大缓存池内存占比	参数值需为非负双精度浮点型  默认为 0.2

New Help		
SystemOptimizationConfig.cacheMiddleMemPercent	中缓存池内存占比	参数值需为非负双精度浮点型 默认为 0.05
SystemOptimizationConfig.cacheSmallMemPercent	小缓存池内存占比	参数值需为非负双精度浮点型 默认为 0.01
SystemOptimizationConfig.cacheStrategy	缓存策略	参数值需为字符串 默认为 mem
SystemOptimizationConfig.chartBigDataNum	限制大数据图表的数量	参数值需为整型 默认值为 3
SystemOptimizationConfig.chartDataThreadPoolNum	图表线程池数量	参数值需为非负整型
		默认值为 8
		0 代表关闭线程池，关闭后，BI 的线程栈将不会出现 chart-group-data-pool 开头的线程 直连时该参数无意义
SystemOptimizationConfig.configLockTimeoutTime	配置锁超时时间设置，单位为秒	参数值需为正长整型 默认值为 30
SystemOptimizationConfig.dashboardEditLimit	仪表板编辑时的行数限制（不选中全部数据）	参数值需为整型 默认值为 10000
SystemOptimizationConfig.detailUseColumnarCompress	明细表是否启用列压缩	参数值需为布尔型，默认为 true
		false：明细表不启用列压缩 true：明细表启用列压缩
SystemOptimizationConfig.directMemoryEngineLimitRows	直连新的内存引擎缓存数据的行数限制	参数值需为整型 默认值为 100000
SystemOptimizationConfig.directUseAllData	直连全部数据是否可用，默认不可用	参数值需为布尔型，默认为 true
		false：直连全部数据可用
		true：直连全部数据不可用
SystemOptimizationConfig.excelExtractDataBase	Excel 数据抽取 直连的关联可选表存在限制，如果系统参数为 Excel 数据抽取的话，Excel 表才可以配置关联	参数值需为布尔型，默认为 false
		false：直连的 Excel 表不可以配置关联 true：直连的 Excel 表可以配置

		关联
SystemOptimizationConfig.excelExtractMaxLimitRows	直连 excel 生成临时表时的行数限制	参数值需为整型 默认值为 10000
SystemOptimizationConfig.fieldInfoTimeoutSeconds	从数据源取字段信息的超时时间	参数值需为长整型 默认值为-1
SystemOptimizationConfig.firstDayofWeek	每周的第一次是星期几	参数值需为整型 默认值为 0
SystemOptimizationConfig.groupUseColumnarCompress	分组表是否启用列压缩	参数值需为布尔型，默认为 true false：分组表不启用列压缩 true：分组表启用列压缩 参数值需为布尔型，默认为 true
SystemOptimizationConfig.inheritPermissionAndRelation	设置权限继承默认开关状态	false：权限继承默认关闭 true：权限继承默认开启
SystemOptimizationConfig.memoryManagerPercent	全局内存管理的大小占比	参数值需为非负双精度浮点型 默认为 0.6
SystemOptimizationConfig.memoryWarningSize	sql 数据集数据量告警值	参数值需为整型 默认值为 100000000 参数值需为布尔型，默认为 false
SystemOptimizationConfig.needTreatRedMark	是否修复标红的自助数据集	false：不修复标红的自助数据集 true：修复标红的自助数据集 参数值需为布尔型，默认为 true
SystemOptimizationConfig.optimizeSqlAlias	是否优化 SQL 别名 (开启会把 SQL 中的别名全部重设一遍)	false：不优化 SQL 别名 true：优化 SQL 别名
SystemOptimizationConfig.queryConditionCountRestriction	明细过滤条件个数限制	参数值需为整型 默认值为-1 参数值需为布尔型，默认为 true false: 添加 db 表时不读取数据库
SystemOptimizationConfig.readRelationFromDbSource	添加 db 表时，是否要读取数据库中的关联	中的关联 true：添加 db 表时读取数据库中的关联
SystemOptimizationConfig.reportShareType	模板分享是否使用创建者权限	参数值为整型 默认值为 0

SystemOptimizationConfig.schedulerMergeTimeSeconds	多少秒进行一次定时任务的合并	参数值需为整型 默认值为 5
SystemOptimizationConfig.tableLoadThreadPoolNum	更新线程数量	参数值需为整型 默认值为 20
SystemOptimizationConfig.threadPoolStrategy	更新线程池策略	参数值如下，默认为 dynamic dynamic：根据内存来算，1G=1 个线程 static：固定线程数量，和 TableLoadThreadPoolNum 一起使用
SystemOptimizationConfig.treeCacheHeap	树结构的缓存个数	参数值需为整型 默认值为-1
SystemOptimizationConfig.treeCacheSeconds	树结构的缓存时长	参数值需为长整型 默认值为-1
SystemOptimizationConfig.useColumnarCompress	分组表是否启用列压缩	参数值需为布尔型，默认为 true false：分组表不启用列压缩 true：分组表启用列压缩
SystemOptimizationConfig.usePagingCache	是否开启分页缓存	参数值需为布尔型，默认为 true false：不开启分页缓存 true：开启分页缓存
TemplateAuthConfig.digitalAuthAvailable	模板认证-数字签名功能是否可使用	5.1.11 及之后版本的 BI 工程支持该参数 参数值如下，默认为 true false：管理员无法使用模板认证-数字签名功能 true：管理员可使用模板认证-数字签名功能
WebSecurityConfig.cacheControlExpiresHeader	浏览器缓存禁用 header expires 的值	参数值如下，默认值为 0 0：代表着过去的日期，即该资源已经过期 <http-date>：在指定日期后，响



		应过期
		参数值如下，默认值为 no-cache
		no-cache: 在发布缓存副本之前，
		强制要求缓存把请求提交给原
WebSecurityConfig.cacheControlHeader	浏览器缓存禁用 header cache-control 的值	始服务器进行验证(协商缓存验证)
		max-age=<seconds>: 设置缓存存储的最大周期，超过这个时间缓存被认为过期(单位秒)
WebSecurityConfig.cacheControlPragmaHeader	浏览器缓存禁用 header pragma 的值	参数值为 no-cache
		参数值如下，默认值为 object-src 'self'
		1) object-src 'self':
		限制<object>、<embed>、<applet>标签的源地址
		且仅允许与当前来源（而不是其子域）匹配
WebSecurityConfig.contentSecurityPolicyHeader	csp 内容安全策略 header content-security-policy 的值	2) object-src uri
		限制<object>、<embed>、<applet>标签的源地址
		且上述标签中地址仅指定 URL 允许加载
		3) default-src https:
		禁用不安全的内联/动态执行, 只允许通过 https 加载这些资源 (images, fonts, scripts, etc.)
WebSecurityConfig.contentTypeOptionsHeader	内容嗅探攻击防护 header x-content-type-options 的值	参数值为 nosniff

		参数值如下，默认为 1
		0：综合，后缀不在白名单内放行，否则校验头
WebSecurityConfig.fileInspectorType	开启了安全管理的文件校验后，用此配置文件上传校验类型	1：白名单，后缀在白名单且头匹配的才放行
		2：黑名单，后缀不在黑名单内放行
		参数值如下，默认值为 SAMEORIGIN
		DENY：浏览器会拒绝当前页面加载任何 frame 页面
WebSecurityConfig.frameOptionsHeader	点击劫持攻击防护 header x-frame-options 的值	SAMEORIGIN：frame 页面的地址只能为同源域名下的页面
		ALLOW-FROM origin：允许 frame 加载的页面地址
		参数值为如下，默认值为 max-age=31536000; includeSubDomains
		1) max-age=<expire-time>：设置在浏览器收到这个请求后的 XXX 秒的时间内凡是访问这个域名下的请求都使用 HTTPS 请求
WebSecurityConfig.hstsHeader	hsts header strict-transport-security 的值	2) max-age=<expire-time>; includeSubDomains：设置在浏览器收到这个请求后的 XXX 秒的时间内凡是访问这个域名下的请求都使用 HTTPS

		请求。此规则也适用于该网站的所有子域名
		3) max-age=<expire-time>; preloaded:
		设置在浏览器收到这个请求后的 XXX 秒的时间内凡是访问这个域名下的请求都使用 HTTPS 请求。预加载 HSTS
		参数值需为布尔型，默认为 false
		false: 不限制跨域登录接口和 cas 后台登录接口的使用
WebSecurityConfig.loginValidateStrict	用于限制跨域登录接口，以及 cas 后台登录接口的使用，如果配置 true，则不能使用这俩接口	true: 限制跨域登录接口和 cas 后台登录接口的使用
		参数值如下，默认值为 1; mode=block
		0: 禁止 XSS 过滤
		1: 启用 XSS 过滤，如果检测到跨站脚本攻击，浏览器将清除页面，删除不安全的部分
WebSecurityConfig.xssProtectionHeader	xss 攻击防护 header x-xss-protection 的值	1; mode=block: 启用 XSS 过滤，如果检测到攻击，浏览器将不会清除页面，而是阻止页面加载
		1; report=<reporting-uri>: 启用 XSS 过滤如果检测到跨站脚本攻击，浏览器将清除页面，并使用 CSP reporting-uri 的功能发送违规报告
WebSocketConfig.port	websocket 服务器监听端口	参数值为端口数组["port1","port2"]

		2"]
		port 均属于区间(1024,65535]
		默认值为["38888","39888"]
		参数值为非空字符串，尽量使用
WebSocketConfig.trustStore	信任证书的文件存放路径	绝对路径
		默认值为__EMPTY__
		参数值为 JKS 或 PKCS12
WebSocketConfig.trustStoreFormat	信任证书的类型	如果是 pfx 证书，改成 PKCS12
		默认值为 JKS
WebSocketConfig.trustStorePassword	信任证书使用时的申请密钥	参数值为非空字符串，尽量使用绝对路径
		默认值为__EMPTY__
		参数值需为布尔型，默认为 false
ClusterRedirectConfig.enableSessionIDGenerationStrategy	是否启用集群下的 sessionID 生成策略	false: 不启用集群下的 sessionID 生成策略
		true: 集群下的 sessionID 的尾部会拼接上集群的 nodeID
		参数值需为布尔型，默认为 false
FineClusterConfig.params.encrypt	集群通信加密开关，默认为 false	false: 集群通信不加密
		true: 集群通信加密
FineClusterConfig.params.ipStackType	集群通信方式	参数值默认为 ipv4，可配置 ipv6
	HTML5 允许关闭开发者计划，用户可以手动关闭	参数值需为布尔型，默认为 true
Html5Config.isJoinProductPlan	是否加入产品改良计划，进一步的保护用户的隐私安全	false: 关闭开发者计划
		true: 开启开发者计划
		默认值为 zh_CN
LanguageConfig.locale	平台全局的语言设置	zh_CN: 简体中文
		zh_TW: 繁体中文
		en_US: 英文

New Help		
ja_JP: 日文		
ko_KR: 韩文		
MarketConfig.cloudOperationMaintenanceId	重置云端运维应用 ID	参数值默认为您的云端运维应用 ID
MobileConfig.emptyDirVisible	控制在移动端（APP、HTML5）是否显示空目录	参数值需为布尔型，默认为 false
	目录	true: 显示空目录
	注：仅对移动端生效。	false: 不显示空目录
ParseCheckConfig.allowDrawing	当 html 解析报错时(各个导出场景遇到不支持的标签和属性)，是否允许将 html 画成图	参数值需为布尔型，默认为 true
		false: 当 html 解析报错时，不允许将 html 画成图
		true: 当 html 解析报错时，允许将 html 画成图
ServerPreferenceConfig.errorTemplate	FR 模板报错页面，可参考帆软官方帮助文档，自定义模板报错页面	参数值为自定义报错页面链接默认为__EMPTY__
参数默认值为__EMPTY__		
参数格式为：{策略序号}:{清理几月数据},{清理时间（时）},{清理时间（分）},{清理周期（秒）};		
SystemOptimizationConfig.clearEntityStrategy	获取自定义的 finedb 表清理策略	例如：
		策略 1，清理 3 月前数据，每 2 天执行一次，4 点 20 分执行
		策略 2，清理 1 月前的数据，每 30 天执行一次，2 点 50 执行。
则该参数为：1:3,4,20,172800;2:1,2,50,2592000		
参数值为服务器内置网卡或对外 IP		
WebSocketConfig.hostName	Websocket 监听 IP 设置	默认为 0.0.0.0

		参数值需为布尔型，默认为 false
WebSocketConfig.webSocketTokenInHeader	socket 通信 token 传递方式	<div>参数值需为布尔型，默认为 false</div> <div>true: token 从请求头传输</div> <div>false: token 从 url 传输</div>
	多级上报的独立链接，首次登录时，页面会弹出弹窗提示：目录权限中有关于上报页面的权限配置，是否从目录权限中同步权限配置	<div>参数值需为布尔型，默认为 false</div> <div>true: 不弹出</div> <div>false: 弹出</div>
WorkflowAuthorityConfig.compatible	该参数用于控制之后是否弹出提示	
DistributedOptimizationConfig.schedulerConfig.local_file_delete_retry_mode	本地文件删除重试机制	<div>参数值需为布尔型，默认为 true</div> <div>true: 本地文件删除重试机制</div> <div>false: 本地文件不删除重试机制</div>
DistributedOptimizationConfig.schedulerConfig.local_file_delete_retry_period	本地文件删除重试周期	参数值需为正整数，默认为 10
DistributedOptimizationConfig.schedulerConfig.local_file_delete_retry_time	本地文件删除重试次数	参数值需为正整数，默认为 3
DistributedOptimizationConfig.schedulerConfig.max_slice_memory_size	slice 最大内存设置	参数值需为正整数，默认为 5242880
DistributedOptimizationConfig.schedulerConfig.spark_blockManagerPort	spark blockManager 端口	<div>参数值需为[1001, 65535]区间内的正整数</div> <div>默认值为 17778</div>
DistributedOptimizationConfig.schedulerConfig.spark_driver_host	spark driver 主机名	参数值需为字符串，默认为空
DistributedOptimizationConfig.schedulerConfig.spark_driver_maxResultSize	spark 返回 driver 端最大结果集大小	参数值需为字符串，默认为 1g
DistributedOptimizationConfig.schedulerConfig.spark_driver_port	spark driver 端口	参数值需为[1001, 65535]区间

		内的正整数
		默认值为 17777
DistributedOptimizationConfig.s piderConfig.spark_executor_core s	spark executor 核心	参数值需为正整数，默认为 12
DistributedOptimizationConfig.s piderConfig.spark_executor_me mory	spark executor 内存	参数值需为字符串，默认为 6g
DistributedOptimizationConfi g.spiderConfig.spark_local_di r	spark 临时文件目录, 计算 吐磁盘目录	参数值需为字符串，默认为空
DistributedOptimizationConfig.s piderConfig.spark_master_host	spark master 主机名	参数值需为字符串，默认为空
		参数值需为[1001, 65535]区间
DistributedOptimizationConfig.s piderConfig.spark_master_port	spark master 端口	内的正整数
		默认值为 7077
DistributedOptimizationConfig.s piderConfig.spark_scheduler_allo cation_file	spark 调度分配文件	参数值需为字符串，默认为空
DistributedOptimizationConfig.s piderConfig.spark_scheduler_mo de	spark 调度模式	参数值需为字符串，默认为 CAPACIT Y
		参数值需为布尔型，默认为 true
DistributedOptimizationConfig.s piderConfig.spark_shuffle_servic e_enabled	shuffle 过程数据外排	true： shuffle 过程数据外排  false： shuffle 过程数据不外排
		参数值需为布尔型，默认为 true
DistributedOptimizationConfig.s piderConfig.spark_sql_adaptive_ enabled	计算自适应功能	true： 计算自适应功能  false： 不计算自适应功能
DistributedOptimizationConfig.s piderConfig.spark_sql_adaptive_ shuffle_targetPostShuffleInputSiz e	计算自适应目标块大 小	参数值需为字符串，默认为 200M
DistributedOptimizationConfig.s piderConfig.spark_sql_broadcast Timeout	broadcast 的超时时间	参数值需为字符串，默认为 12000
DistributedOptimizationConfig.s piderConfig.spark_sql_shuffle_pa rtitions	spark shuffle 的并发块 数	参数值需为字符串，默认为空
DistributedOptimizationConfig.s piderConfig.spark_sql_sortMerge JoinExec_buffer_in_memory_thre shold	sortMergeJoin 算子内 存最大数据量	参数值需为字符串，默认为 100000

DistributedOptimizationConfig.s piderConfig.spark_sql_sortMerge JoinExec_buffer_spill_threshold	sortMergeJoin 算子吞 吐阈值	参数值需为字符串，默认为 100000
DistributedOptimizationConfig.s piderConfig.spider_array_group_ map_function_threshold	数组分组算法阈值	参数值需为正整数，默认为 5000000
DistributedOptimizationConfig.s piderConfig.spider_base_update _thread_proportion	基础表线程占比	参数值需为正整数，默认为 30
DistributedOptimizationConfig.s piderConfig.spider_build_field_in dex_cores	构建字段索引核心数	参数值需为正整数，默认为 8
DistributedOptimizationConfig.s piderConfig.spider_build_index_c ores	构建索引核心数	参数值需为正整数，默认为 4
DistributedOptimizationConfig.s piderConfig.spider_column_max _load_dictionary_key_size	数据列最大载入字典 数量	参数值需为正整数，默认为 10000
DistributedOptimizationConfig.s piderConfig.spider_compress_sli ce_size	抽数时，每个分片的行 数	参数值需为正整数，默认为 100000
DistributedOptimizationConfig.s piderConfig.spider_core_limit_fra ction	spark 限制核心任务评 分	参数值需为非负双精度浮点型 默认为 0.67
DistributedOptimizationConfig.s piderConfig.spider_create_satelli te_async	异步创建卫星表	参数值需为布尔型，默认为 true true： 异步创建卫星表 false： 异步不创建卫星表
DistributedOptimizationConfig.s piderConfig.spider_deletion_crite ria_size_per_execution	增量删除每次最大计 算大小	参数值需为字符串，默认为 128m
DistributedOptimizationConfig.s piderConfig.spider_detail_formul a_use_page	明细公式使用分页	参数值需为布尔型，默认为 true true： 明细公式使用分页 false： 明细公式不使用分页 参数值需为正整数
DistributedOptimizationConfig.s piderConfig.spider_distinct_medi an_threshold	去重计数/中位数内存 阈值	5.1.12 之前的版本，默认值为 10 0 5.1.12 及之后的版本，默认值为 25
DistributedOptimizationConfig.s piderConfig.spider_double_dicti onary_max_key_size	double 类型最大字典	参数值需为正整数，默认为 10000



数量	
DistributedOptimizationConfig.spiderConfig.spider_engine_use_page	参数值需为布尔型，默认为 true true：使用分页引擎 false：不使用分页引擎
DistributedOptimizationConfig.spiderConfig.spider_etl_coalesce_size	参数值需为正整数，默认为 10
DistributedOptimizationConfig.spiderConfig.spider_etl_compute_limit_unit	参数值需为正整数，默认为 5000000
DistributedOptimizationConfig.spiderConfig.spider_etl_core_limit	参数值需为布尔型，默认为 false true：etl 限制核心 false：etl 不限制核心
DistributedOptimizationConfig.spiderConfig.spider_etl_iterator_repartition_size	参数值需为正整数，默认为 20
DistributedOptimizationConfig.spiderConfig.spider_etl_no_order	参数值需为布尔型，默认为 false true：etl 无序 false：etl 有序
DistributedOptimizationConfig.spiderConfig.spider_etl_output_use_bitmap	参数值需为布尔型，默认为 true true：ETL 生成索引 false：ETL 不生成索引
DistributedOptimizationConfig.spiderConfig.spider_etl_preview_row_count	参数值需为字符串，默认为 5000
DistributedOptimizationConfig.spiderConfig.spider_etl_writer_lock_wait	参数值需为大于等于-1 的整数 默认为 60000
DistributedOptimizationConfig.spiderConfig.spider_fast_compute_limit_memory	快速计算内存限制字节大小 参数值需为正长整型 默认为 500000000
DistributedOptimizationConfig.spiderConfig.spider_fast_compute_limit_row	快速计算行数限制 参数值需为正整数，默认为 1000000
DistributedOptimizationConfig.spiderConfig.spider_fast_compute_limit_unit	快速计算单元格限制 参数值需为正整数，默认为 1000000
DistributedOptimizationConfig.spiderConfig.spider_fast_etl_cache	快速计算使用缓存 参数值需为布尔型，默认为 true true：快速计算使用缓存

		false：快速计算不使用缓存
DistributedOptimizationConfig.spiderConfig.spider_file_max_mmap_size	文件最大 mmap 大小	参数值需为正整数，默认为 2147483647
DistributedOptimizationConfig.spiderConfig.spider_file_mmap_size	文件 mmap 大小	参数值需为正整数，默认为 1073741823
DistributedOptimizationConfig.spiderConfig.spider_first_day_of_week	星期的第一天	参数值需为[0, 6]区间内的整数 默认值为 1
DistributedOptimizationConfig.spiderConfig.spider_float_dictionary_max_key_size	float 类型最大字典数量	参数值需为正整数，默认为 10000
DistributedOptimizationConfig.spiderConfig.spider_generate_join_cache_in_query	在查询时生成关联缓存	参数值需为布尔型，默认为 true true：在查询时生成关联缓存 false：在查询时不生成关联缓存
DistributedOptimizationConfig.spiderConfig.spider_high_performance_pool_size	不保证计算性能时，关联和分析表资源池大小	参数值需为正整数，默认为 1000
DistributedOptimizationConfig.spiderConfig.spider_high_performance_resource_time	高性能资源占用时间	参数值需为字符串，默认为空
DistributedOptimizationConfig.spiderConfig.spider_ignore_base_update	是否跳过基础表更新	参数值需为布尔型，默认为 false true：跳过基础表更新 false：不跳过基础表更新
DistributedOptimizationConfig.spiderConfig.spider_index_build_bitmap_dataset	索引生成使用 dataset	参数值需为布尔型，默认为 false true：索引生成使用 dataset false：索引不生成使用 dataset
DistributedOptimizationConfig.spiderConfig.spider_index_build_bitmap_external	索引生成使用外排	参数值需为布尔型，默认为 true true：索引生成使用外排 false：索引不生成使用外排
DistributedOptimizationConfig.spiderConfig.spider_index_build_bitmap_external_buffer_path	索引缓存文件路径	参数值需为字符串 默认为 distributed/external
DistributedOptimizationConfig.spiderConfig.spider_index_build_bitmap_external_buffer_size	索引缓冲大小	参数值需为正整数，默认为 100000

DistributedOptimizationConfig.spiderConfig.spider_int_dictionary_max_key_size	int 类型最大字典数量	参数值需为正整数，默认为 10000
		参数值为指定值，默认值为 0
	5.1.12 及之后的版本	0：表示数据结构为 KolobokeIntSet，速度快但内存占用大（平均一个 int key 占用 8 字节）
DistributedOptimizationConfig.spiderConfig.spider_int_distinct_structure	新增该参数 控制去重记录数的数据结构	1：表示数据结构为 RoaringBitmap，速度慢但内存占用小（平均一个 int key 占用 1 字节）
DistributedOptimizationConfig.spiderConfig.spider_join_always_use_sql	关联始终使用 SQL 计算	参数值需为布尔型，默认为 false true：关联始终使用 SQL 计算 false：关联始终不使用 SQL 计算
DistributedOptimizationConfig.spiderConfig.spider_join_build_index	关联生成索引	参数值需为布尔型，默认为 true true：关联生成索引 false：关联不生成索引
DistributedOptimizationConfig.spiderConfig.spider_join_cache_delete_direct	关联缓存立刻删除	参数值需为布尔型，默认为 true true：关联缓存立刻删除 false：关联缓存不立刻删除
DistributedOptimizationConfig.spiderConfig.spider_join_dot_constant	关联点常量	参数值需为字符串，默认为.
DistributedOptimizationConfig.spiderConfig.spider_join_equal_constant	关联等号常量	参数值需为字符串，默认为=
DistributedOptimizationConfig.spiderConfig.spider_join_huge_table_size	关联大表大小	参数值需为非负整数，默认为 0
DistributedOptimizationConfig.spiderConfig.spider_join_left_bracket_constant	关联左括号常量	参数值需为字符串，默认为(
DistributedOptimizationConfig.spiderConfig.spider_join_left_square_bracket_constant	关联左中括号常量	参数值需为字符串，默认为[
DistributedOptimizationConfig.spiderConfig.spider_join_name_simple	关联使用简单命名	参数值需为布尔型，默认为 false true：关联使用简单命名 false：关联不使用简单命名

DistributedOptimizationConfig.spiderConfig.spider_join_right_bracket_constant	关联右括号常量	参数值需为字符串，默认为)
DistributedOptimizationConfig.spiderConfig.spider_join_right_square_bracket_constant	关联右中括号常量	参数值需为字符串，默认为]
DistributedOptimizationConfig.spiderConfig.spider_join_use_relation	使用关联缓存	参数值需为布尔型，默认为 true true：使用关联缓存 false：不使用关联缓存
DistributedOptimizationConfig.spiderConfig.spider_load_data_buffer_queue_size	抽数的缓冲队列大小	参数值需为正整数，默认为 100
DistributedOptimizationConfig.spiderConfig.spider_load_data_buffer_row_size	抽数的缓冲行大小	参数值需为正整数，默认为 1000 参数值需为布尔型，默认为 true
DistributedOptimizationConfig.spiderConfig.spider_load_use_bitmap	抽数生成索引	true：抽数生成索引 false：抽数不生成索引
DistributedOptimizationConfig.spiderConfig.spider_local_compute_cores	本地计算核心数	参数值需为正整数，默认为空
DistributedOptimizationConfig.spiderConfig.spider_local_default_db	数据默认 DB	参数值需为字符串，默认为 db
DistributedOptimizationConfig.spiderConfig.spider_local_root_path	数据抽取存放路径	参数值需为字符串，默认为 ../spider
DistributedOptimizationConfig.spiderConfig.spider_lock_wait_time	锁的默认等待时间	参数值需为正整数，默认为 1000
DistributedOptimizationConfig.spiderConfig.spider_long_dictionary_max_key_size	long 类型最大字典数量	参数值需为正整数，默认为 10000
DistributedOptimizationConfig.spiderConfig.spider_low_performance_small_pool_size	保证计算性能时，关联生成资源池大小	参数值需为正整数，默认为 4
DistributedOptimizationConfig.spiderConfig.spider_max_rowstream_memory_size	etl rowstream 内存限制	参数值需为正整数，默认为 5242880
DistributedOptimizationConfig.spiderConfig.spider_mem_filtration_model_ratio	过滤后内存评估比率	参数值需为非负双精度浮点型，默认为 0.3
DistributedOptimizationConfig.spiderConfig.spider_memory_check	分组算法内存统计频率	参数值需为正整数，默认为 100000

ck_frequency_group	率	
DistributedOptimizationConfig.spiderConfig.spider_merge_forbid_time	合并禁止时间	参数值需为字符串，默认为空
DistributedOptimizationConfig.spiderConfig.spider_number_set_dictionary_max_key_size	数值类型字典编码是否设置最大字典数量	参数值需为布尔型，默认为 false true：设置最大字典数量 false：不设置最大字典数量
DistributedOptimizationConfig.spiderConfig.spider_page_summary_concurrent_threads	分页指标计算多线程	参数值需为布尔型，默认为空 true：分页指标计算多线程 false：分页指标不计算多线程
DistributedOptimizationConfig.spiderConfig.spider_partition_limitation	Partition 行限制	参数值需为正整数，默认为 1000
DistributedOptimizationConfig.spiderConfig.spider_partition_merge_policy	partition 合并策略	参数值需为字符串，默认为空
DistributedOptimizationConfig.spiderConfig.spider_partition_row_size	分区行数	参数值需为正整数，默认为 10000000
DistributedOptimizationConfig.spiderConfig.spider_partition_slice_size	分区抽数时，每个分片的行数	参数值需为正整数，默认为 5000
DistributedOptimizationConfig.spiderConfig.spider_query_condition_count_restriction	引擎查询 condition 数量限制	参数值需为大于等于-1 的整数 默认为-1
DistributedOptimizationConfig.spiderConfig.spider_reduce_row_key_ratio	reduce 使用字典的比率	参数值需为非负双精度浮点型，默认为 1.0
DistributedOptimizationConfig.spiderConfig.spider_relation_column_strategy	关联列的生成策略	参数值需为字符串，默认为 SectionShrink
DistributedOptimizationConfig.spiderConfig.spider_relation_matcher_policy	关联匹配规则	参数值需为字符串，默认为 HeadTail
DistributedOptimizationConfig.spiderConfig.spider_restrictions_column_long_text	最长本文限制	参数值需为正整数，默认为 1000
DistributedOptimizationConfig.spiderConfig.spider_retry_max_times	重试最大次数	参数值需为正整数，默认为 2
DistributedOptimizationConfig.spiderConfig.spider_row_index_strategy	分析表行索引优化	参数值需为布尔型，默认为 false true：分析表行索引优化

	false: 分析表行索引不优化
DistributedOptimizationConfig.spiderConfig.spider_satellite_creation_step 卫星表创建步长	参数值需为正整数，默认为 10000000
DistributedOptimizationConfig.spiderConfig.spider_section_merge_large_section_count section 合并大块数量	参数值需为正整数，默认为 6
DistributedOptimizationConfig.spiderConfig.spider_section_merge_policy section 合并策略	参数值需为字符串，默认为 grouping
DistributedOptimizationConfig.spiderConfig.spider_section_merge_row_threshold_factor section 合并行数阈值	参数值需为正整数，默认为 1000000
DistributedOptimizationConfig.spiderConfig.spider_section_merge_small_section_count section 合并小块数量	参数值需为正整数，默认为 4
	参数值需为布尔型，默认为 false
DistributedOptimizationConfig.spiderConfig.spider_spark_driver_use_random_port spark driver 使用随机端口	true: spark driver 使用随机端口 false: spark driver 不使用随机端口
DistributedOptimizationConfig.spiderConfig.spider_spark_local_threads spark 本地线程数	参数值需为正整数，默认为空
DistributedOptimizationConfig.spiderConfig.spider_sparksql_filter_pushdown sparkSQL 过滤下推	参数值需为布尔型，默认为 true true: sparkSQL 过滤下推 false: sparkSQL 过滤不下推
DistributedOptimizationConfig.spiderConfig.spider_sql_merge_table 使用 sql 合并表	参数值需为布尔型，默认为 true true: 使用 sql 合并表 false: 不使用 sql 合并表
DistributedOptimizationConfig.spiderConfig.spider_stream_partition_estimated_byte 流式分区预估大小	参数值需为正整数，默认为 200000000
DistributedOptimizationConfig.spiderConfig.spider_submit_spark_thread_count 提交 spark 线程数	参数值需为正整数，默认为 5
DistributedOptimizationConfig.spiderConfig.spider_table_estimate_strategy 估算表大小的方式	参数值需为字符串，默认为 SAMPLE
DistributedOptimizationConfig.spiderConfig.spider_temp_folder_path_for_spark spark 临时文件路径	参数值需为字符串 默认为/root/temp/spark
DistributedOptimizationConfig.spiderConfig.spider_update_fast_analysis_user自助数据集	参数值需为正整数，默认为 10000000

compute_limit_cell	磁盘占用大小	参数值需为布尔型，默认为 false
DistributedOptimizationConfig.spiderConfig.spider_updated_background	增量更新合并是否后台执行	true：增量更新合并后台执行 false：增量更新合并不在后台执行
DistributedOptimizationConfig.spiderConfig.spider_updated_partition_num	触发合并的更新分区数	参数值需为正整数，默认为 2
DistributedOptimizationConfig.spiderConfig.spider_use_fast_etl_local_optimize	快速计算使用本地优化	参数值需为布尔型，默认为 false true：快速计算使用本地优化 false：快速计算不使用本地优化
DistributedOptimizationConfig.spiderConfig.spider_use_pin_yin_sort	使用中文拼音排序	参数值需为布尔型，默认为 false true：使用中文拼音排序 false：不使用中文拼音排序
TemplateIOErrorContextConfig.showTemplateMissingPlugin	插件缺失时报错记录在日志中的功能是否开启	参数值需为布尔型，默认为 false false：不开启 true：开启
FSConfig.loginConfig.fWords	自定义配置 LDAP 认证用户名登录搜索字段	参数值格式为：["值 1","值 2"] 参数值不允许为空，参数长度不允许为 0
UpdateConfig.launchSwitchMap.plugins	插件自动更新的启动更新是否开启	参数默认值为["sAMAccountName","cn","userPrincipalName","uid","displayName","name","sn"] 如需关闭工程启动后的插件自动更新，请将参数值修改为 false，否则无需改动
CronUpdateConfig.typeMap.plugins	插件自动更新的定时更新是否开启	如需关闭工程启动后每晚 11 点的外网更新插件，请将参数值修改为 false，否则无需改动
DelayUpdateConfig.delaySwitchMap.plugins	插件自动更新的延迟更新是否开启	如需关闭工程启动后延迟一段时间的插件自动更新，请将参数值修改为 false，否则无需改动
DistributedOptimizationConfig.spiderConfig.spider_high_performance_resource_time	设置更新可占用高性能资源的时间段	如需设置晚七点-早八点，请将参数值修改为 19:00:00-8:00:00





## 8. 日志管理

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.5.5	-
V1.5.6	1) 新增「运维平台日志」，支持下载运维平台自身的日志文件 2) 不再提供对出错日志、业务分析日志进行分析的相关功能
V1.5.7	1) 日志管理功能拆分，划分为三个模块：运维日志、性能堆栈、dump文件 2) 集群多节点工程，支持单次下载多节点日志文件 3) 各类日志下载时，支持选择服务类型，按需下载日志

#### 1.2 功能简介

运维平台提供「日志管理」功能。



### 2. 索引

功能模块	简介	帮助文档
运维日志	当用户遇到需要使用日志的场景时，部分公司对机房管控较严	<a href="#">运维日志</a>

格，申请进机房流程繁琐，日志获取费时费力。

运维平台功能出现问题时，需要通过运维平台自身的 fanruan.log 日志定位问题。

FineOps 运维平台，支持下载运维项目的 debug 日志，支持下载运维平台自身的 fanruan.log 日志。

当工程出现宕机或异常时：

- 1) 在工程宕机时，管理员需要快速生成堆栈日志，以确保事后有办法复盘定位问题。
- 2) 管理员需要对堆栈日志进行分析，追踪问题产生的根本原因，但日志分析对管理员的技术水平要求较高。

性能堆栈

[性能堆栈](#)

- 3) 管理员需要将堆栈日志拿到本地自行分析，或提供给帆软技术支持人员，帮助确认工程存在的问题，但很多公司日志获取费时费力。

FineOps 运维平台提供「性能堆栈」功能，支持对堆栈日志进行快速生成、智能分析、便捷下载。

FineOps 运维平台支持宕机自动/手动快速生成 dump 文件

- 1) 当工程非正常退出时，可以使用 dump 日志来了解崩溃的原因。这些原因可能包括堆栈溢出、内存泄漏、代码错误等。
- 2) 当工程出现内存泄漏时，dump 日志可以帮助开发人员查找内存泄漏的源头并修复问题。
- 3) 当工程卡顿或死锁时，dump 日志可以提供有关卡顿或死锁的线程信息以及导致问题的原因。
- 4) 当工程运行时出现错误时，dump 日志可以提供有关错误的详细信息，如发生错误的代码行、变量值等。

dump 文件

[dump 文件](#)

## 8.1 运维日志

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.5.5	-
	1) 运维平台日志管理功能拆分优化
V1.5.7	2) 支持单次下载集群多节点日志
	3) 支持选择指定服务类型的日志
V1.5.9	1) 「运维日志」支持查看「日志基础信息」
	2) 「运维日志」下载时，支持选择下载日志类型
	1) 运维日志新增四种日志，包括：
	ops-agent 组件日志：agent.log
V1.13.0	FineBI 工程日志：eventlog、polars.log、operator_calculate.log
	2) 运维平台日志下载，支持下载更多运维组件相关日志

#### 1.2 功能简介

当用户遇到需要使用日志的场景时，部分公司对机房管控较严格，申请进机房流程繁琐，日志获取费时费力。

运维平台功能出现问题时，需要通过运维平台自身的 fanruan.log 日志定位问题。

FineOps 运维平台，支持下载运维项目的 debug 日志，支持下载运维平台自身的 fanruan.log 日志。

### 2. 运维日志

功能简介：

当用户遇到需要使用日志的场景时，部分公司对机房管控较严格，申请进机房流程繁琐，日志获取费时费力。

FineOps 运维平台，支持查看运维日志存储位置，支持下载运维日志，包括以下类型的 debug 日志：

日志	说明
----	----

fanruan.log	帆软应用运行日志
	仅 FineBI 项目支持查看和下载该日志
standard.log	FineBI 集群日志信息，一般默认存放在 logs/cluster 路径下
	仅 FineBI 项目支持查看和下载该日志
eventlog	记录了数据集更新在 spark 引擎中计算的具体信息，用于分析 spark 执行慢问题
	仅 FineBI 项目支持查看和下载该日志
polars.log	仪表板和数据集编辑涉及的计算日志
	仅 FineBI 项目支持查看和下载该日志
operator_calculate.log	仪表板和数据集编辑涉及的计算日志
access.log	Nginx 访问日志的访问日志文件，用于记录用户访问网站的请求信息，如访问时间、IP 地址、请求方法、访问路径、HTTP 状态码等
catalina.out	Tomcat 的标准输出和标准出错，所有输出到这两个位置的都会进入 catalina.out，这里包含 tomcat 运行自己输出的日志以及应用里向 console 输出的日志
catalina.log	Tomcat 自己运行的一些日志，主要记录 Tomcat 在启动和暂停时的运行内容
localhost_access_log.txt	Tomcat 的请求访问日志，请求的时间，请求的类型，请求的资源 and 返回的状态码都有记录
messages.txt	Linux 系统中的系统日志文件，用于记录系统中发生的各种事件和错误信息，如启动、关机、硬件故障、软件崩溃等。这些信息可以帮助管理员及时发现和解决系统问题，保障系统的稳定运行。
agent.log	ops-agent 组件日志

2.1 查看日志基础信息

操作步骤：

- 1) 管理员登录运维平台，点击「日志管理>运维日志>运维日志」。选择需要下载日志的「运维项目」。
- 2) 在「日志基础信息」中，点击「查看」按钮，即可查看该运维项目中各个工程节点的运维日志存储位置。

注意事项：

- 1) 容器化部署的运维项目，如安装 elasticsearch 组件，用于日志存储，则此处不显示储存位置，提示「elasticsearch 日志服务」
- 2) 如不存在某种类型的日志，则不展示该日志的存储位置。



2.2 下载运维日志

序号	操作步骤	注意事项
	管理员登录运维平台	
1	点击「日志管理>运维日志>运维日志」  选择需要下载日志的「运维项目」  <div>可选范围：15 天前~当前日期，所选时间跨度不可超过 2 天（4</div>	
2	选择时间范围  <div>默认时间：2 小时前~当前时间</div> <div>支持类型：BI、FR</div> <div>可选范围：所选时间内存在的服务类型，默认全选</div>	
3	选择服务类型	注意事项：  1) 非容器化运维项目不支持选择服务类型  2) 如不存在任一类型，则不出现该选项
4	选择节点	可选范围：所选运维项目中的所有节点，默认全选

支持类型：

fanruan.log、standard.log、access.log、catalina.out、catalina.log、localhost\_access\_log.txt、messages.txt

5 选择需下载的日志

注意事项：

- 1) 如不存在某种类型的日志，则不支持下载该类型日志
- 2) 仅 FineBI 项目，支持下载 standard.log 日志
- 3) 容器化部署的运维项目，如安装 elasticsearch 组件用于日志存储，则不显示该项，直接全量下载

6 点击「下载」，生成运维日志文件到本地电脑

- 1) 若文件大小超过 5G，则无法导出该日志，提示「以下日志大小超过上限（5GB），不支持下载：xxx」
- 2) 若该类型日志文件不存在，则无法导出该日志，提示「以下日志不存在：xxx」
- 3) 日志下载时，文件需要进行压缩，若等待时间过长，用户可点击「取消下载」按钮停止下载
- 4) 下载的文件解压后，按照「节点-日期-服务类型-日志类型」结构存放



3. 运维平台日志

功能简介：

运维平台功能出现问题时，需要通过运维平台自身的 fanruan.log 日志定位问题。

FineOps 运维平台，支持下载 ops 和相关运维组件日志，包括以下：

日志	说明
----	----

docker 内容器的日志，包括以下容器：

dockerLogs.log	• alertmanager
	• elasticsearch
	• elasticsearch_exporter
	• grafana
	• nginx
	• nginx_exporter
	• node_exporter
	• ops
	• ops_agent
	• prometheus
access.log	• registry
	• skywalking_oap
access.log	Nginx 访问日志的访问日志文件，用于记录用户访问网站的请求信息，如访问时间、IP 地址、请求方法、访问路径、HTTP 状态码等
error.log	记录 FineOps 运维平台预期外的报错
fanruan.log	记录 FineOps 运维平台运行日志
agent.log	ops-agent 日志

操作步骤：

1) 管理员登录运维平台，点击「日志管理>运维日志>运维平台日志」。选择需要下载日志的「运维项目」。

2) 选择操作时间。

- 可选范围：任意值，若日期超出当前存在的日志文件范围，则仅下载现有文件。
- 默认时间：前一天~当天

3) 点击「下载」，生成运维平台日志文件到本地电脑。

- 若文件大小超过 5G，则无法导出该日志，提示「以下日志大小超过上限（5GB），不支持下载：xxx」
- 若该类型日志文件不存在，则无法导出该日志，提示「以下日志不存在：xxx」
- 日志下载时，文件需要进行压缩，请耐心等待。
- 下载的文件解压后，按照「组件类型-日志类型」存放。

运维平台

运维管理

监控与告警

业务工具

日志管理

运维日志

性能堆栈

dump文件

运维日志

运维平台日志

操作时间

2023-09-18

→

2023-09-19

下载

708



## 8.2 性能堆栈

### 1. 概述

#### 1.1 版本

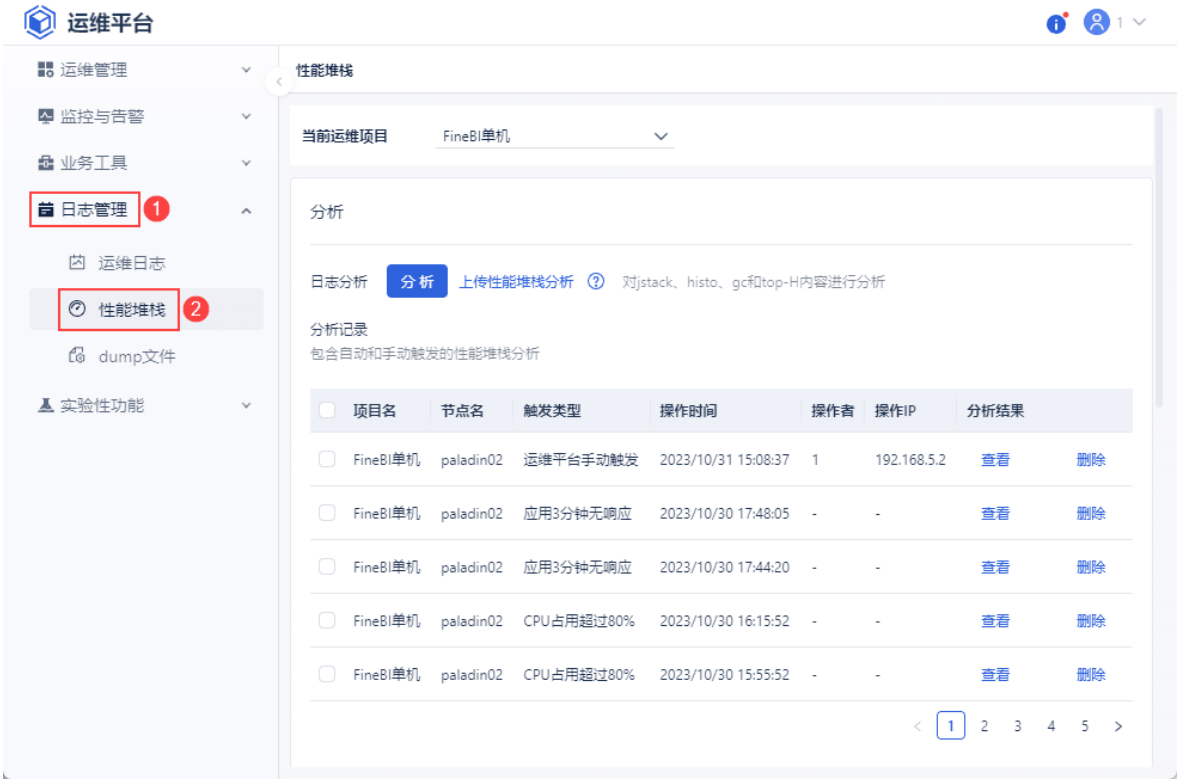
运维平台版本	功能变更
V1.5.5	-
	1) 运维平台日志管理功能拆分优化
V1.5.7	2) 对于集群多节点工程，支持单次下载多节点日志文件
	3) 各类日志下载时，支持选择服务类型，按需下载日志
V1.8.0	分析日志中，分析记录列表新增展示「触发类型」
	1) 支持手动上传本地性能堆栈，至运维平台进行分析
V1.12.0	2) 支持展示历史性能堆栈生成记录
	3) 性能堆栈使用顺序优化，变更为：分析>下载>生成
	运维平台监测宕机，自动触发生成日志条件优化
	1) 应用无响应触发时间由 3 分钟优化为 2 分钟
V1.14.0	2) 新增 FullGC 触发条件：5min 内 FullGC 次数总和大于 3 次，且时间总和超过 50s
V1.15.0	下载日志时，支持按需选择日志类型，无需全量下载，降低下载成本

#### 1.2 应用场景

当工程出现宕机或异常时：

- 1) 在工程宕机时，管理员需要快速生成堆栈日志，以确保事后有办法复盘定位问题。
- 2) 管理员需要对堆栈日志进行分析，追踪问题产生的根本原因，但日志分析对管理员的技术水平要求较高。
- 3) 管理员需要将堆栈日志拿到本地自行分析，或提供给帆软技术支持人员，帮助确认工程存在的问题，但很多公司日志获取费时费力。

FineOps 运维平台提供「性能堆栈」功能，支持对堆栈日志进行快速生成、智能分析、便捷下载。



1.3 日志简介

性能堆栈功能，支持对以下堆栈日志进行分析、下载和生成：

日志	说明
fanruan.gc.log	帆软宕机日志
thread.txt	线程信息统计报告，该报告包含了所有线程的状态、堆栈跟踪信息等，可以帮助了解应用程序的线程使用情况，及时发现和解决线程问题。
histo.txt	内存使用情况统计报告，该报告包含了对象的数量、大小、占用空间等信息，可以帮助了解应用程序的内存使用情况，及时发现和解决内存泄漏等问题。
hs_err.txt	Java 在运行过程中遇到错误时自动生成的错误报告文件，包含了详细的错误信息、堆栈跟踪信息、线程信息等，可以帮助定位和解决代码问题。
top.txt	Linux 系统下的 top 命令的输出结果，用于显示当前系统中运行进程的相关信息，包括 top-H.txt、top-sortedByCPU.txt、top-sortedByMEM.txt 1) top-H.txt：该文件是 top 命令输出结果中的 "H" 模式，用于显示线程信息，包括线程 ID、CPU 占用率、内存占用率等 2) top-sortedByCPU.txt：该文件是 top 命令输出结果按照 CPU 占用率从高到低排序的信息，包括进程 ID、CPU 占用率、内存占用率等 3) top-sortedByMEM.txt：该文件是 top 命令输出结果按照内存占用率从高到低排序的信息，包括进程 ID、CPU 占用率、内存占用率等

2. 分析日志

应用场景：

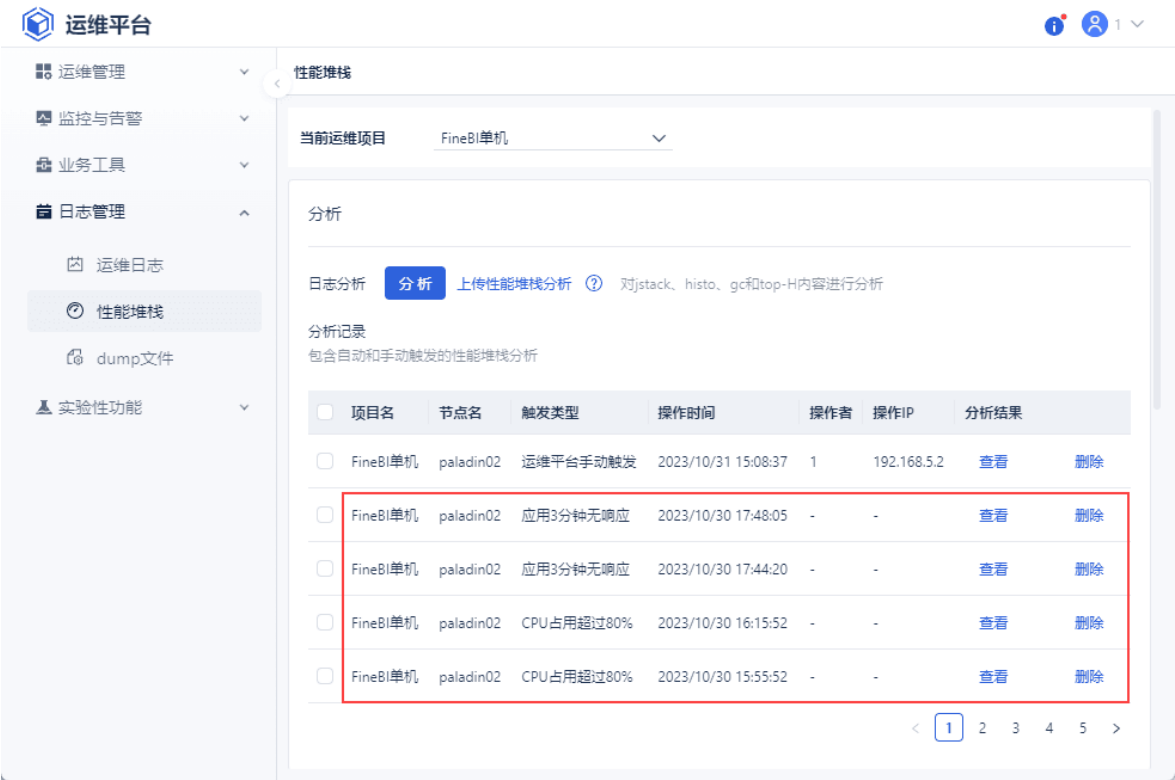
当工程宕机后，管理员需要对宕机日志进行分析，修复工程存在的问题，防止再次发

生。

FineOps 运维平台支持对 jstack、histo、gc 和 top-H 内容进行分析，帮助用户分析错误堆栈并给出相关改进意见。

2.1 自动分析运维项目

当满足 4.1 节自动生成日志后，FineOps 会对每个节点自动生成的堆栈日志进行分析，自动分析结果生成在分析记录中。



2.2 手动分析运维项目

操作步骤：

- 1) 管理员登录运维平台，点击「日志管理>性能堆栈」。选择需要分析日志的「运维项目」。
- 2) 在「分析」模块下，点击「分析」按钮。触发该运维项目下每个节点生成堆栈日志。
- 3) FineOps 会对每个节点生成的堆栈日志进行分析，分析结果生成在分析记录中。

注意事项：

每 1 分钟仅支持进行 1 次分析，以防频繁触发，导致系统性能差。若分析请求过于频繁，则提示「分析请求太频繁啦，请稍等一分钟后再重新尝试」。



### 2.3 手动上传日志分析

用户可手动上传本地堆栈日志到运维平台进行分析。

#### 操作步骤：

##### 1) 准备本地堆栈日志包

仅支持.zip/.tar/.gz 格式压缩包，将根据压缩包中 jstack、histo、gc 和 top-H 生成分析报告，仅支持同时分析一个节点。

##### 2) 上传本地堆栈日志包

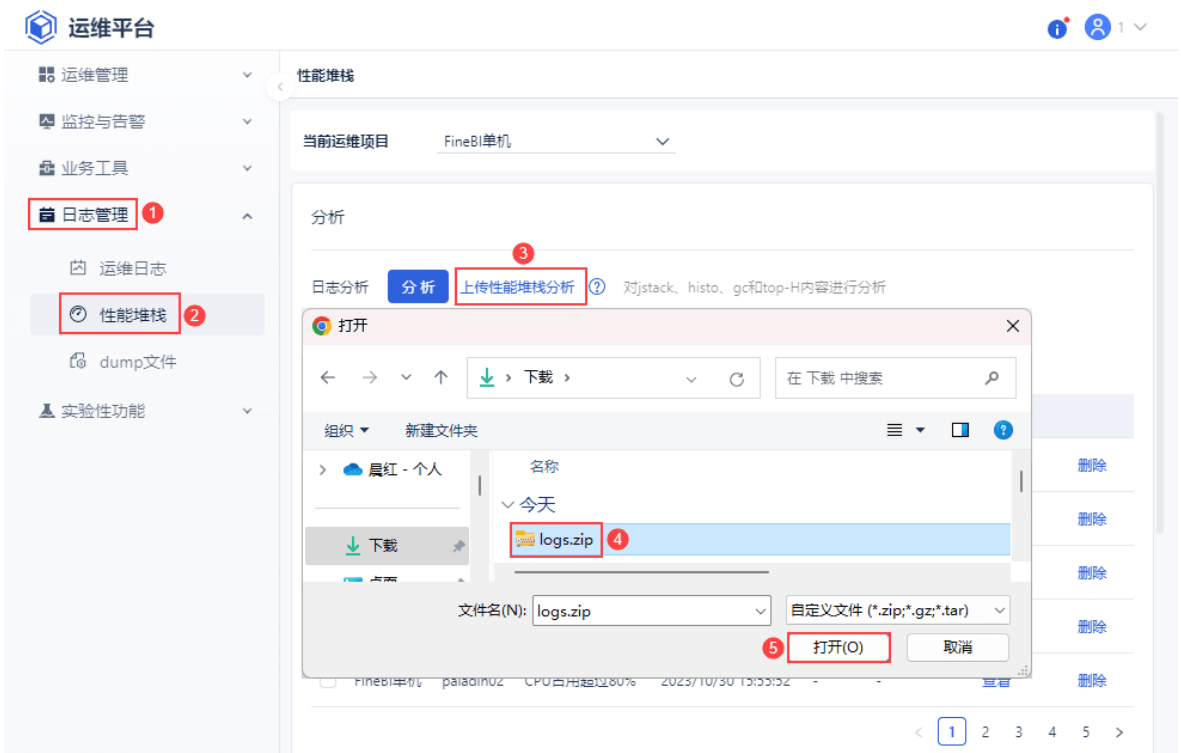
管理员登录运维平台，点击「日志管理>性能堆栈」。

在「分析」模块下，点击「上传性能堆栈分析」按钮，选择本地日志包，点击「打开」。

##### 3) 生成分析报告

FineOps 将根据压缩包中 jstack、histo、gc 和 top-H 生成分析报告，仅支持同时分析一个节点。

分析结果在新标签页打开，不记录在分析记录中。



## 2.4 查看分析记录

管理员登录运维平台，点击「日志管理>性能堆栈」。选择「运维项目」。

在「分析>分析记录」模块下，以明细表形式显示该运维项目下所有手动/自动触发的堆栈分析结果（不包括手动上传的日志分析记录）。

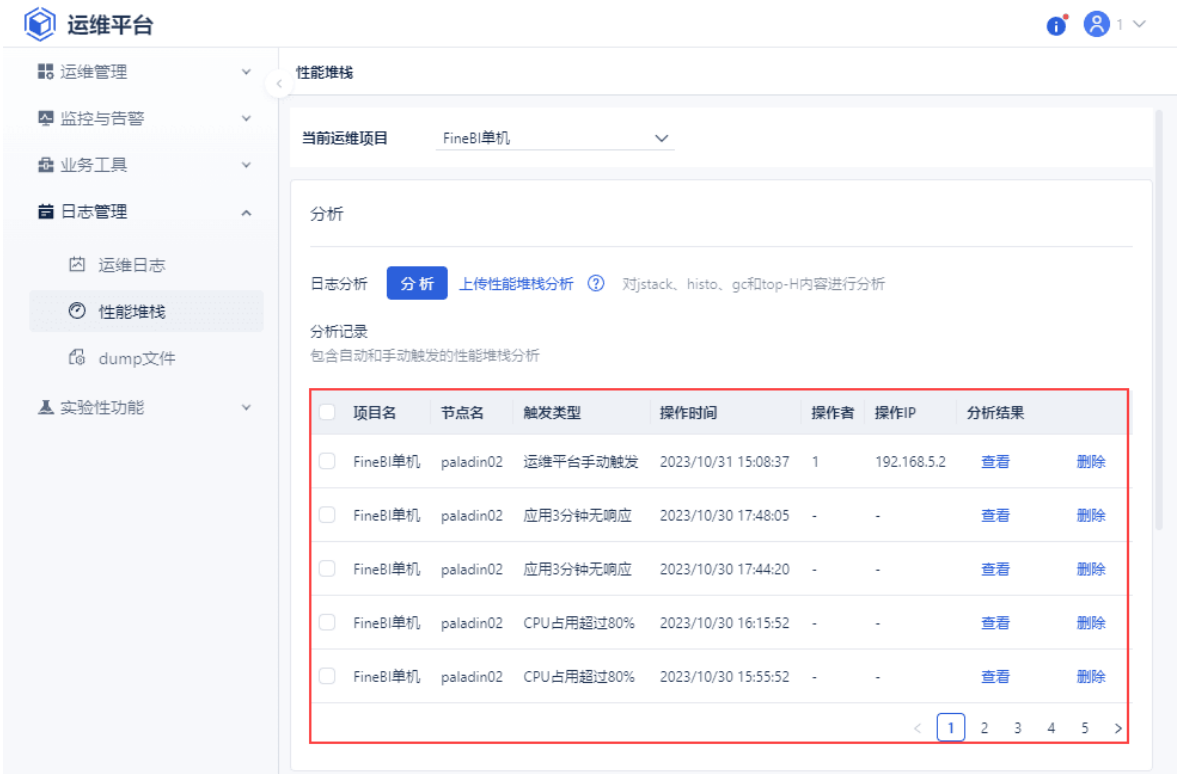
1) 分析记录以明细表形式展示，展示内容包括：项目名、节点名、触发类型、操作时间、操作者、操作 IP、分析结果。

若手动触发，操作者和操作 IP 为空。

2) 若运维项目为集群环境，每个节点都会生成一份分析结果。

3) 日志分析历史记录最多只保留最近 50 条记录。支持选中单个分析记录删除，支持批量删除多个分析记录，支持清空全部历史记录。

4) 点击分析记录中的「查看」按钮，即可在新标签页打开具体的分析结果。



点击对应分析结果后的「查看」按钮，即可查看本次分析的具体结果。

性能堆栈分析结果示例如下图所示，包括分析建议、内存状态和明细、GC 状态和耗时。

性能堆栈

分析建议

原因：  
2023-10-30 17:48:18：数据源取数耗时过长导致线程阻塞  
2023-10-30 17:48:29：数据源取数耗时过长导致线程阻塞  
2023-10-30 17:48:33：数据源取数耗时过长导致线程阻塞  
解决方案：  
2023-10-30 17:48:18：建议使用抽数缓存、SQL优化、数据预处理等方式提升取数性能  
2023-10-30 17:48:29：建议使用抽数缓存、SQL优化、数据预处理等方式提升取数性能  
2023-10-30 17:48:33：建议使用抽数缓存、SQL优化、数据预处理等方式提升取数性能

[分析过程详情](#)

[2023-10-30 17:48:18](#)    2023-10-30 17:48:29    2023-10-30 17:48:33

线程分组汇总

同类线程	数量
GC task thread	53
direct-chart-group-data-pool	48
interrupt-pool	41
spider-group-summary-pool	32
http-nio	15
async-cache-insert-thread	12

<< < 1 / 7 > >>

受阻塞耗时线程

线程名	线程cpu占用情况	堆栈深度	线程堆栈
-----	-----------	------	------

<< < 1 / 1 > >>

GC状态：

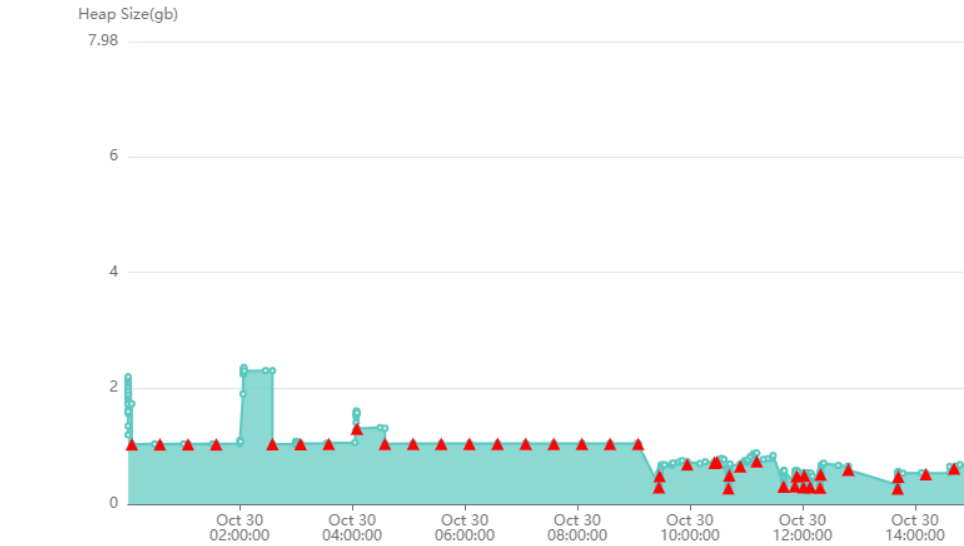
内存是否过载：否

GC状态是否异常：否

[分析过程详情](#)

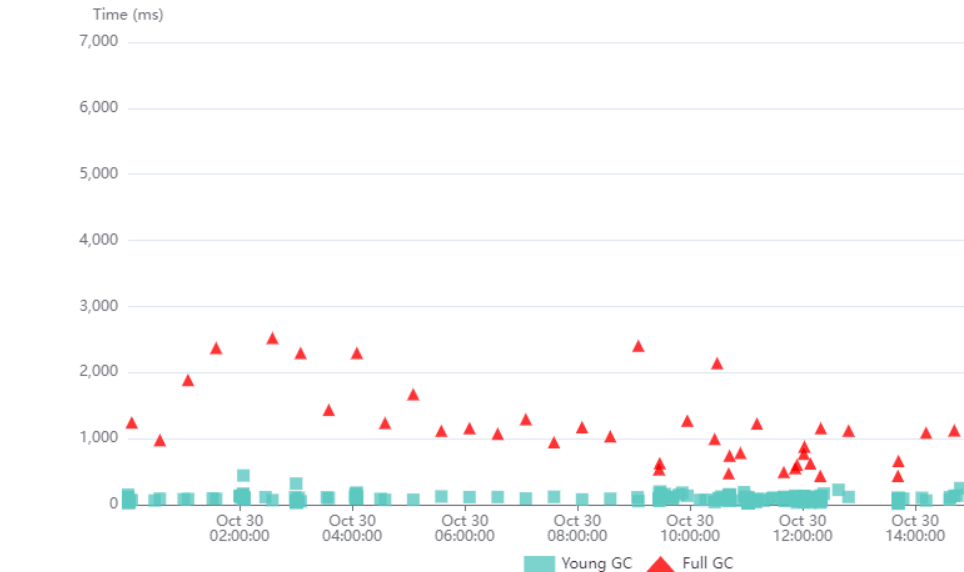
内存使用情况

Heap Usage (after GC)



gc耗时情况

GC Duration Time



内存使用

### 3. 下载日志

应用场景：

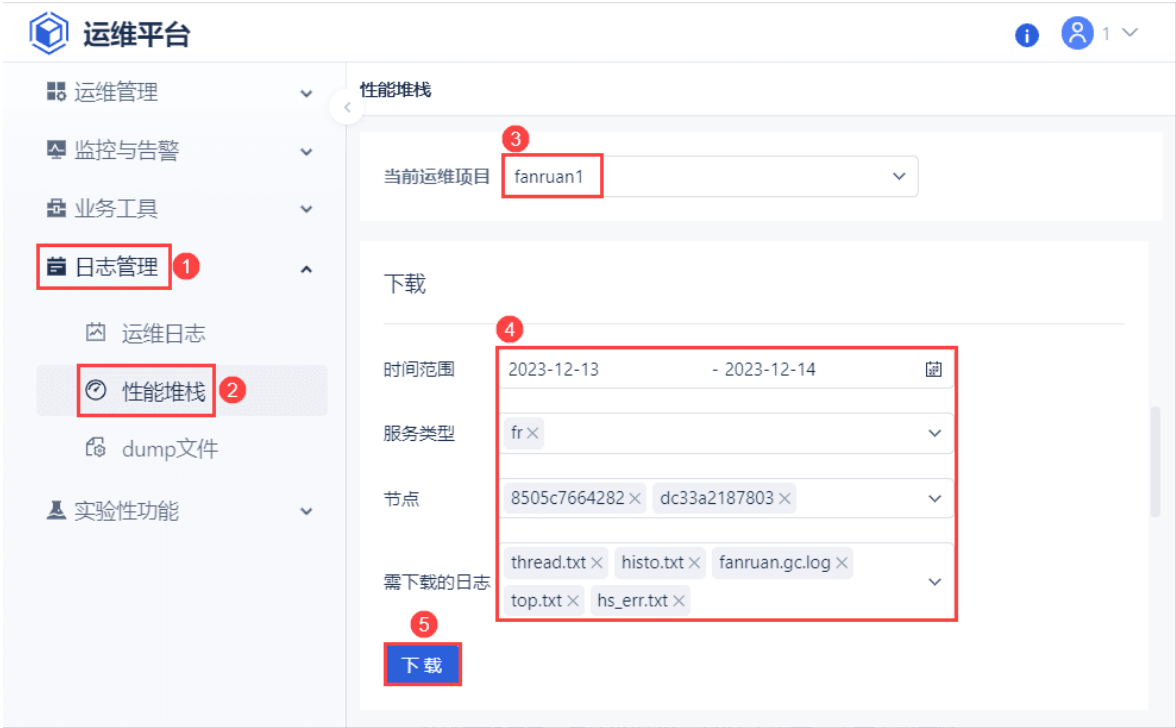
当工程宕机后，管理员需要将宕机日志拿到本地自行分析，或提供给帆软技术支持人员，帮助确认工程存在的问题，防止再次发生。

操作步骤：

序号	操作步骤	注意事项
	管理员登录运维平台	
1	点击「日志管理>性能堆栈」	
	选择需要下载日志的「运维项目」	
		可选范围：7 天前~当前日期，所选时间跨度不可超过 2 天（48 小时）
2	选择时间范围	默认时间：2 小时前~当前时间
		支持类型：BI、FR
		可选范围：所选时间内存在的服务类型，默认全选
3	选择服务类型	注意事项： 1) 非容器化运维项目不支持选择服务类型 2) 如不存在任一类型，则不出现该选项
4	选择节点	可选范围：所选运维项目中的所有节点，默认全选
		支持类型： fanruan.gc.log、thread.txt、histo.txt、hs_err.txt、top.txt
5	选择需下载的日志	注意事项： 1) 如不存在某种类型的日志，则不支持下载该类型日志 2) 容器化部署的运维项目，如安装 elasticsearch 组件用于日志存储，则不支持该项，直接全量下载
6	点击「下载」，生成运维日志文件到本地电脑	1) 若文件大小超过 5G，则无法导出该日志，提示「以下日志大小超过上限（5GB），不支持下载：xxx」 2) 若该类型日志文件不存在，则无法导出该日志，提示「以下日志不存在：xxx」 3) 日志下载时，文件需要进行压缩，若等待时间过长，用户可点击「取消下载」按钮停止下载



4) 下载的文件解压后，按照「节点-日期-服务类型-日志类型」结构存放



## 4. 生成日志

### 应用场景：

当工程宕机或出现异常时，堆栈日志可以帮助管理员快速定位问题所在的代码。

当工程出现性能问题（如响应时间过长或 CPU 占用过高）时，堆栈日志可以帮助管理员理解程序的执行路径，找到可能的性能瓶颈。

当工程出现问题时，histo 日志可以帮助管理员追踪使用者历史输入的命令，保持运维人员对用户行为的监控，追踪问题产生的根本原因

运维平台支持手动/自动生成堆栈和 histo 日志。

### 4.1 自动生成

#### 1) 触发自动生成日志

以下三种情况，均会触发自动生成日志。

情况	说明
运维平台	1) 开启「宕机自动处理」
监测宕机	管理员登录运维平台，点击「业务工具>宕机自动处理>宕机处理策略」。
触发	选择需要生成日志的「运维项目」，开启「宕机自动处理」，点击「保存」。



2) 配置「自动生成」内容

管理员登录运维平台，点击「日志管理>性能堆栈」。选择需要生成日志的「运维项目」。

在「生成>自动生成」模块下：

支持选择生成类型：包括堆栈、histo，默认全选

支持选择服务类型：包括 BI、FR、FDL，默认全选（非容器化运维项目不支持选择服务类型；如不存在任一类型，则不出现该选项）



3) 当该运维项目出现以下情况时，自动触发生成日志

CPU 占用超过 80%

发生超过 30S 的 GC

应用 2 分钟无响应

线程池用量达到 85%

连续 FullGC 后依旧负载较高（连续 5 次 FullGC 且负载分大于 100）

5min 内 FullGC 次数总和大于 3 次，且时间总和超过 50s

项目  
自身  
监测  
宕机

1) 开启「宕机自动处理」

管理员登录运维项目工程，点击「管理系统>智能运维>宕机处理>宕机自动处

触发「  
理工具」。

开启「宕机自动处理」。开启「自动导出内存堆栈」，选择生成类型：包括堆栈、  
histo、dump。

点击「保存」。



2) 当该运维项目出现以下情况时，自动触发生成日志

CPU 占用超过 80%

发生超过 30S 的 GC

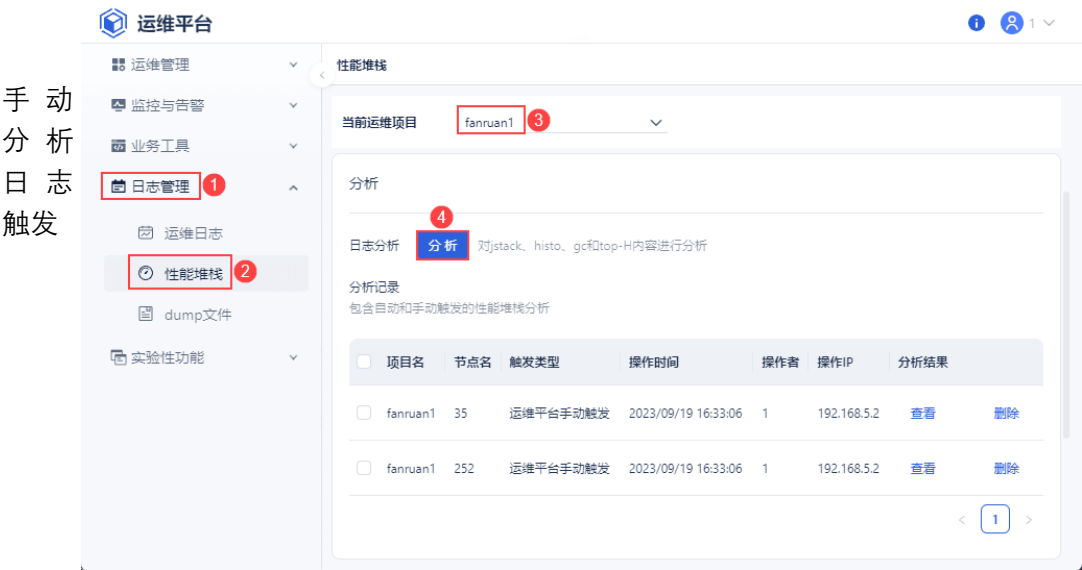
应用 3 分钟无响应

线程池用量达到 95%

连续 FullGC 后依旧负载较高（连续 5 次 FullGC 且负载分大于 100）

管理员登录运维平台，点击「日志管理>性能堆栈」。选择需要生成日志的「运  
维项目」。

当手动点击「日志分析>分析」按钮时，会先自动触发生成一份堆栈日志。



2) 「自动生成」的日志

当满足上文条件后，自动触发该运维项目下每个节点生成堆栈日志。

生成的文件位置为：

- 容器化项目：项目每个工程节点挂载目录的 logs\FineLog\日期文件夹内。
- 非容器化项目：项目每个节点所在服务器下%Tomcat%\logs\FineLog\日期文件夹内。

4.2 手动生成

使用前提：

如需生成堆栈日志，必须先开启「宕机自动处理」功能。否则提示「功能尚未启动，如需启动请前往业务工具>宕机自动处理>宕机处理策略，开启【宕机自动处理】开关」。  
管理员登录运维平台，点击「业务工具>宕机自动处理>宕机处理策略」，选择需要生成日志的「运维项目」，开启「宕机自动处理」开关，点击「保存」即可。



操作步骤：

- 1) 管理员登录运维平台，点击「日志管理>性能堆栈」。选择需要生成日志的「运维项目」。
- 2) 在「生成>手动生成」模块下，选择服务类型。
  - 支持类型：BI、FR
  - 可选范围：当前运维项目所有节点中存在的服务类型；非容器化运维项目不支持选择服务类型；如不存在任一类型，则不出现该选项。
- 3) 点击「生成」，触发该运维项目下每个节点生成堆栈日志。
- 4) 生成的文件位置为：

- 容器化项目：项目每个工程节点挂载目录的 logs\FineLog\日期文件夹内。
- 非容器化项目：项目每个节点所在服务器下%Tomcat%\logs\FineLog\日期文件夹内。

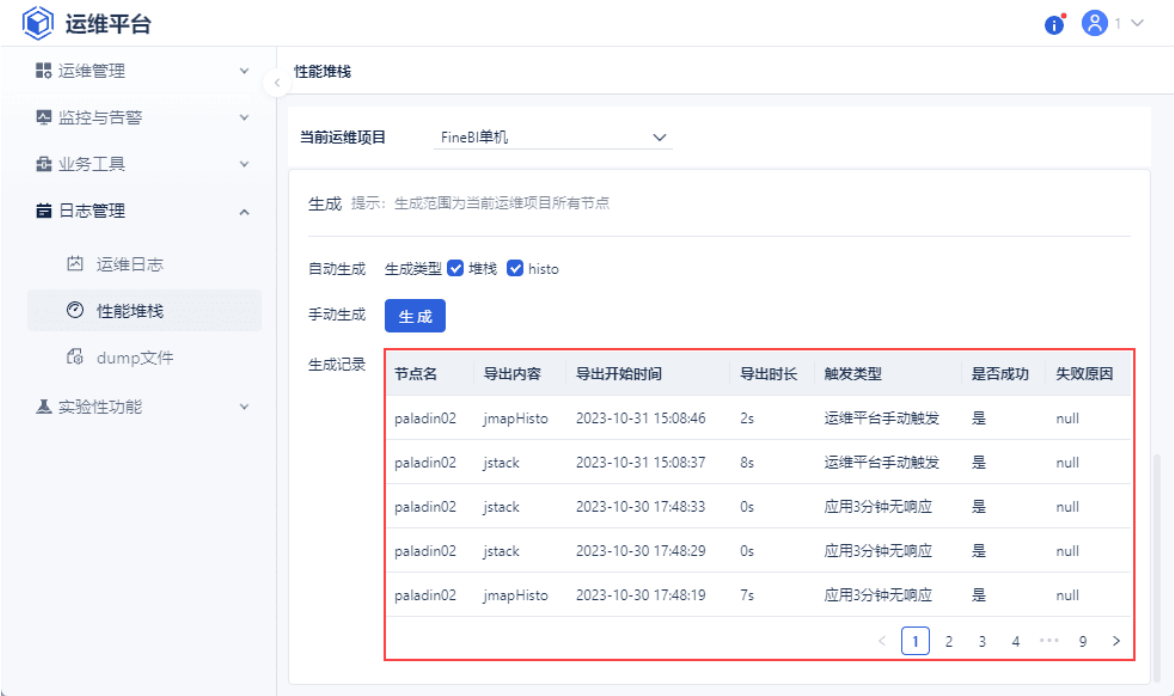
注意事项：

每 1 分钟仅支持生成 1 次堆栈，以防频繁触发，导致系统性能差。



4.3 生成记录

在性能堆栈中，展示历史生成记录，包含所有自动/手动触发生成的情况。如下图所示：以明细表形式展示：节点名、导出内容、导出开始时间、导出时长、触发类型、是否成功、失败原因。



## 8.3 dump 文件

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.5.5	-
V1.5.7	运维平台日志管理功能拆分优化

#### 1.2 应用场景

- 1) 当工程非正常退出时，可以使用 dump 日志来了解崩溃的原因。这些原因可能包括堆栈溢出、内存泄漏、代码错误等。
- 2) 当工程出现内存泄漏时，dump 日志可以帮助开发人员查找内存泄漏的源头并修复问题。
- 3) 当工程卡顿或死锁时，dump 日志可以提供有关卡顿或死锁的线程信息以及导致问题的原因。
- 4) 当工程运行时出现错误时，dump 日志可以提供有关错误的详细信息，如发生错误的代码行、变量值等。

#### 1.3 功能简介

FineOps 运维平台提供「dump 文件」功能，支持宕机自动/手动快速生成 dump 文件。



### 2. 手动生成

#### 操作步骤：

- 1) 管理员登录运维平台，点击「日志管理>dump 文件」。选择需要生成日志的「运维

- 项目」。
- 2) 选择节点。默认不选，仅支持单选。
- 3) 点击「生成」，触发该运维项目下该工程节点生成 dump 文件。
- 4) 生成的文件位置为：该节点所在服务器下%Tomcat%\logs\FineLog\日期文件夹内。

注意事项：

每 20 分钟仅支持生成 1 次 dump，以防频繁触发，导致系统性能差。



3. 自动生成

1) 开启「宕机自动处理」

管理员登录运维平台，点击「业务工具>宕机自动处理>宕机处理策略」。

选择需要生成日志的「运维项目」，开启「宕机自动处理」，点击「保存」。



2) 配置「自动生成」内容

管理员登录运维平台，点击「日志管理>dump 文件」。选择需要生成日志的「运维项目」。

开启「自动生成」按钮。



### 3) 触发「自动生成」日志

当工程宕机自动重启后，自动触发该运维项目有问题的节点生成 dump 日志。

生成的文件位置为：该节点所在服务器下%Tomcat%\logs\FineLog\日期文件夹内。



## 9. 实验性功能

## 9.1 链路追踪

## 9.1.1 场景案例-健康观测

## 9.1.2 首屏链路

## 9.1.3 调用链路

本文涉及模块属于「实验性功能」，此类功能使用门槛较高，功能仍在进行打磨优化中。

若您有相关实际使用场景，或需获取使用帮助、反馈功能相关建议和评价，付费用户请联系技术支持。

技术支持联系方式：前往「[服务](#)」，选择「在线支持」、电话「400-811-8890」

### 1. 概述

#### 1.1 版本

运维平台 版本	功能变更
V1.5.5	-
	1) 查询界面新增「重置」按钮，方便重置查询
	2) 调用链路列表展示优化，支持展开收起，方便放大记录明细区
V1.5.7	3) 调用链路明细新增耗时查询按钮，支持使用控件筛选满足耗时范围的链路
	1) 优化「收起侧边列表」按钮交互
V1.6.0	2) 支持点击复制 traceID
	对于使用 elasticsearch 套件获取和存储日志的运维项目，在查看链路详情时，支持
V1.7.0	获取日志到本地
V1.8.0	查询界面新增筛选项：端点、状态、实例
	1) 调用链路详情新增「分析汇总」视图，帮助用户快速洞察耗时分布，汇总展示链路 span 信息
V1.11.0	2) 调用链路详情新增「链路信息」按钮，汇总展示链路关键信息，减少查询时间
	3) 调用链路列表新增「FGC」标识，标记链路时间范围内是否发生 FullGC
V1.12.0	若某个调用链路所在 session 有相关首屏链路数据，则支持联动跳转「查看首屏链路」
V1.14.0	1) 调用链路详情「列表」视图下，链路信息中，新增「自助数据集」字段，若有即

可查看

2) 调用链路详情「列表」视图，新增「span 关键字查询」功能，支持搜索包含指定关键字的 span

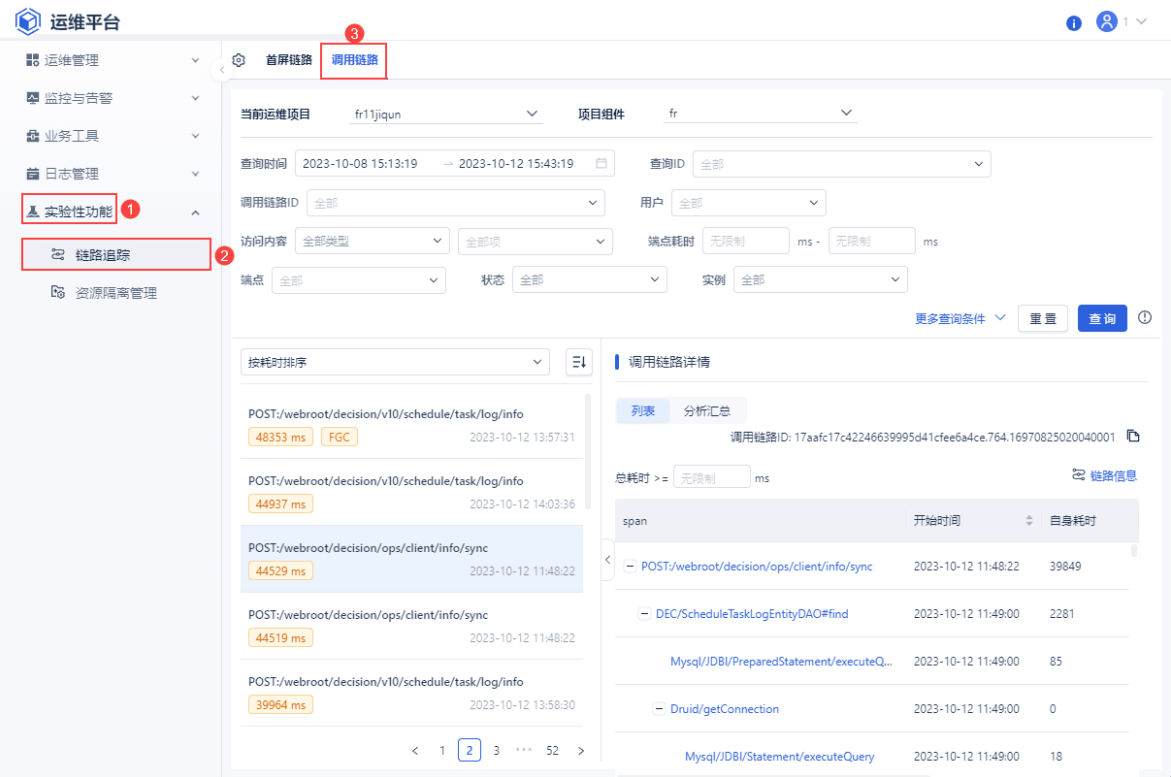
3) 调用链路详情「分析汇总」视图下，支持点击 span 名称跳转至「列表」视图下，以名称作为关键词进行搜索

- V1.15.0
- 1) 支持自定义链路追踪数据留存天数，满足个性化分析诉求

2) 支持手动清空运维项目链路数据，防止磁盘不足引起异常

1.2 功能简介

运维平台提供「调用链路」功能，帮助用户对查询请求中的链路明细进行采集和分析。支持根据 ID、时间等，筛选请求，界面化展示相关链路，帮助管理员监控系统性能异常点。



2. 使用前提

2.1 启用运维组件

链路追踪功能，由这些运维组件提供：elasticSearch、skywalking\_oap。

1.5.7 及之后版本，这些组件默认启动，用户可手动关闭组件。1.5.6 及之前版本，这些组件默认不启动。

如需使用「链路追踪」功能，必须先确保已启动这些组件。否则提示「链路追踪组件未启用，请在运维组件目录中启动 elasticsearch 以及 skywalking\_oap」。如下图所示：



管理员登录运维平台，点击「运维管理>运维组件」，先启动 elasticSearch 组件，再启动 skywalking\_oap 组件。



## 2.2 磁盘可用空间

请确保 FineOps 运维平台所在分区，已使用磁盘空间不足 90%，否则无法启用链路追踪功能。

```
[root@iZ28x3egfhlZ ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/vda1        79G   43G   33G   58% /
devtmpfs         32G    0    32G    0% /dev
tmpfs            32G    0    32G    0% /dev/shm
tmpfs            32G  2.4M   32G    1% /run
tmpfs            32G    0    32G    0% /sys/fs/cgroup
/dev/vdb1        246G   89G  146G   38% /data_bak
/dev/vdc1        246G  224G   11G   96% /data
tmpfs            6.3G    0   6.3G    0% /run/user/0
```

## 2.3 工程部署方式

- 1) 非容器化部署的运维项目默认不支持启用链路追踪，如需使用，请先参考该文档调整：[非 finekey 部署应用接入链路追踪](#)。
- 2) 运维项目与运维平台，是同一个容器化部署工具 finekey 进行部署的。该运维项目可默认启用链路追踪。

3) 运维项目是容器化部署的, 但与运维平台并非同一容器化部署工具 finekey 部署的。

该运维项目需要手动「启用链路追踪」, 启用链路追踪时提醒「启用链路追踪需要重启应用, 是否启用」。

点击「重启项目并继续」按钮, 即会自动重启项目并启用链路追踪功能。



2.4 全局设置

管理员登录运维平台, 点击「实验室功能>链路追踪>全局设置」, 支持对链路追踪功能进行配置。

建议在使用链路追踪相关功能前, 配置好全局设置项。

设置项	说明
数据留存天数	链路数据留存的天数, 默认 5 天
	增加天数需要确保磁盘有充足的空间, 当磁盘空间不足时会自动停止数据采集
	建议按照「项目数*增加的数据留存天数*10G」来评估磁盘增量
	注: 磁盘空间不足时, 可点击「全局设置>关闭链路追踪」中的「清空」按钮清空链路数据。
采集次数上限	3 秒内同一端点的采集次数限制
	默认值-1, 表示不限制
	输入范围为-1 ~ 10000
不采集的端点	不进行采集的端点, 如有多个端点, 使用逗号分割
	默认不进行采集的端点包括:



/webroot/decision/metrics/schedule  
/webroot/decision/system/health  
/webroot/decision/metrics/once

采集单个端点的最大跨度（span）数量，超出的将被舍弃

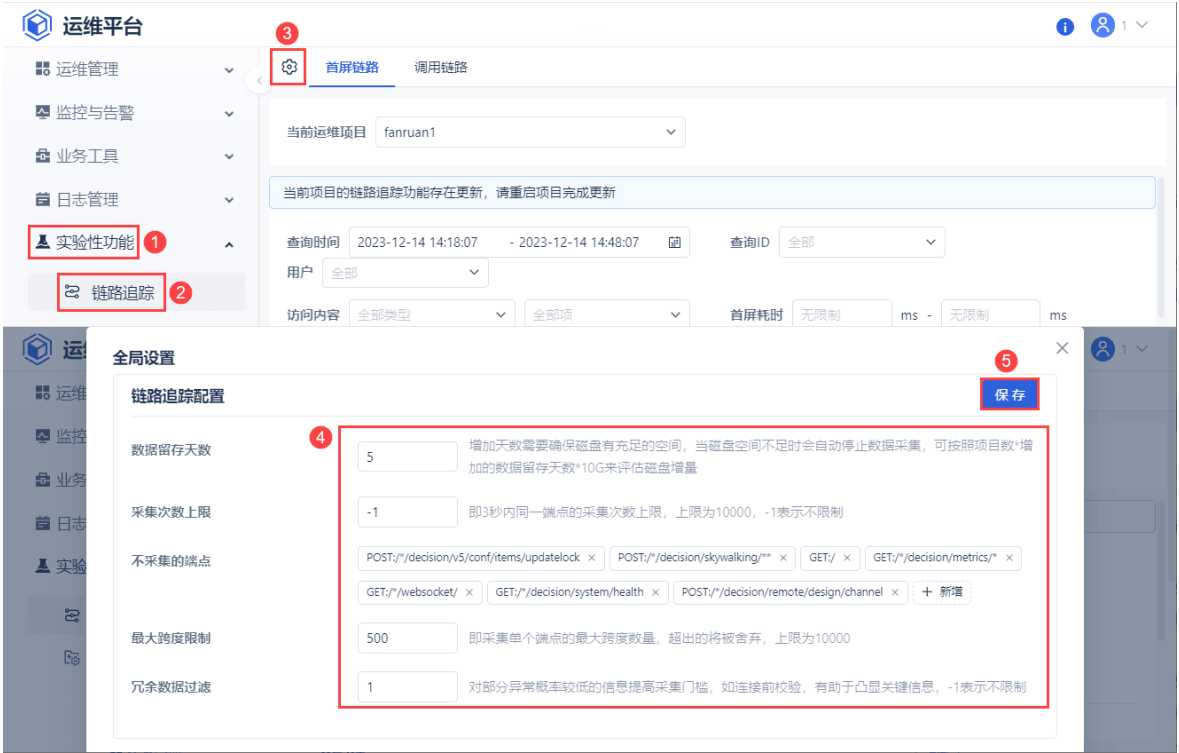
最大跨度限制 默认值 300

输入范围为 0 ~ 10000

对部分异常概率较低的信息提高采集门槛，如连接前校验，有助于凸显关键信息

冗余数据过滤 默认值 5ms

-1 表示不限制



3. 使用示例

3.1 配置查询条件

- 1) 管理员登录运维平台，点击「实验性功能>链路追踪>调用链路」，选择需要进行查询的运维项目和项目组件。
- 2) 配置查询条件，点击「查询」。支持的查询条件如下表所示：

查询条件	简介
说明：	筛选所有在此时间范围内开始的查询
可选项：	支持选择当前日期过去 5 天范围内，但查询区间跨度不得大于 3 天
查询时间	默认值：当前时间前半小时内

	说明：查找对应 session 相关记录
查询 ID	<p>可选项：时间范围内的所有 sessionID</p> <p>默认值：为空，即展示全部</p> <p>说明：查找对应 traceID 的调用链路</p>
调用链路 ID	<p>可选项：时间范围内的所有 traceID</p> <p>默认值：为空，即展示全部</p> <p>说明：筛选该用户触发的所有查询</p>
用户	<p>可选项：所有有记录的用户</p> <p>默认值：为空，即展示全部</p> <p>说明：筛选查询对象对应的查询记录</p> <p>可选项：</p>
访问内容	<ul style="list-style-type: none"><li>FineReport：模板实际路径</li><li>FineBI：仪表板名称、组件名称</li></ul> <p>默认值：为空，即展示全部</p> <p>说明：筛选调用链路耗时在区间内的查询记录</p>
端点耗时	<p>可选项：非负整数，为空则表示不限制上限/下限</p> <p>默认值：为空，即展示全部</p> <p>说明：筛选调用链路类型</p>
端点	<p>可选项：时间范围内的所有端点</p> <p>默认值：为空，即展示全部</p> <p>说明：筛选调用链路成功/失败的情况</p> <p>可选项：</p>
状态	<ul style="list-style-type: none"><li>成功：调用链路成功</li><li>失败：调用链路失败</li></ul> <p>默认值：为空，即展示全部</p>
实例	说明：筛选运维项目中的节点对应的链路，仅集群工程支持该筛选项

可选项：选择集群中的各个节点

默认值：为空，即展示全部

说明：筛选所有标签等于所选值条件的查询

可选项：sessionId、reportId、dec.user.name、fr.domain、fr.cmd、db.type

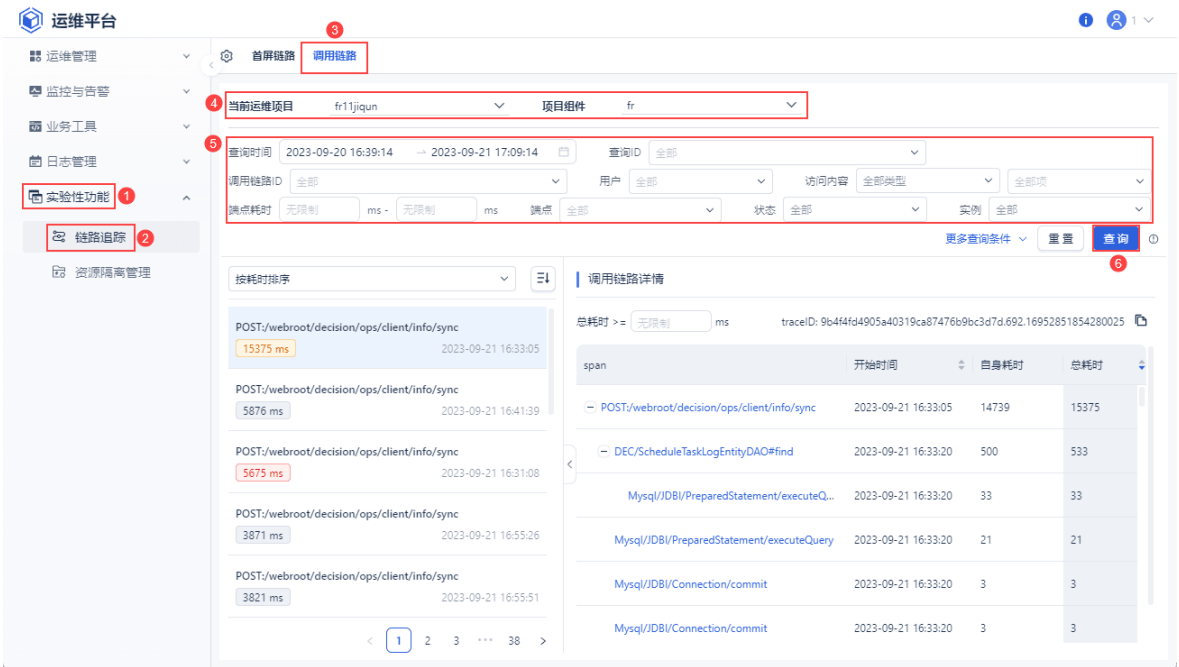
更多

e、http.method、http.status\_code、db.instance（后五项不可同时选中）

默认值：为空，即展示全部

重置

说明：点击该按钮，即可一键清空以上所有查询条件



3.2 链路结果展示

查询后的界面如下图所示：

序号	说明
	支持两种排序方法，支持降序/升序：
1	调用链路排序方式 按耗时排序：按照查询的首屏耗时时间排序
	按开始时间排序：按照查询触发的时间排序
1)	展示每一条调用链路的缩略信息：
	访问内容：post/get 请求、URL
	链路耗时：黄色代表请求超过 10s，红色代表请求报错，其他为灰色
	FGC 标识：标记链路时间范围内是否发生 FullGC
	时间：链路发生时间
2)	点击右侧折叠按钮，支持折叠展开左侧调用链路列表区域

1) 视图切换

以列表/分析汇总视图展示每条 span 的详情，左上角支持切换视图

2) 调用链路 ID

右上角显示链路 ID，点击即可复制

3) 获取日志

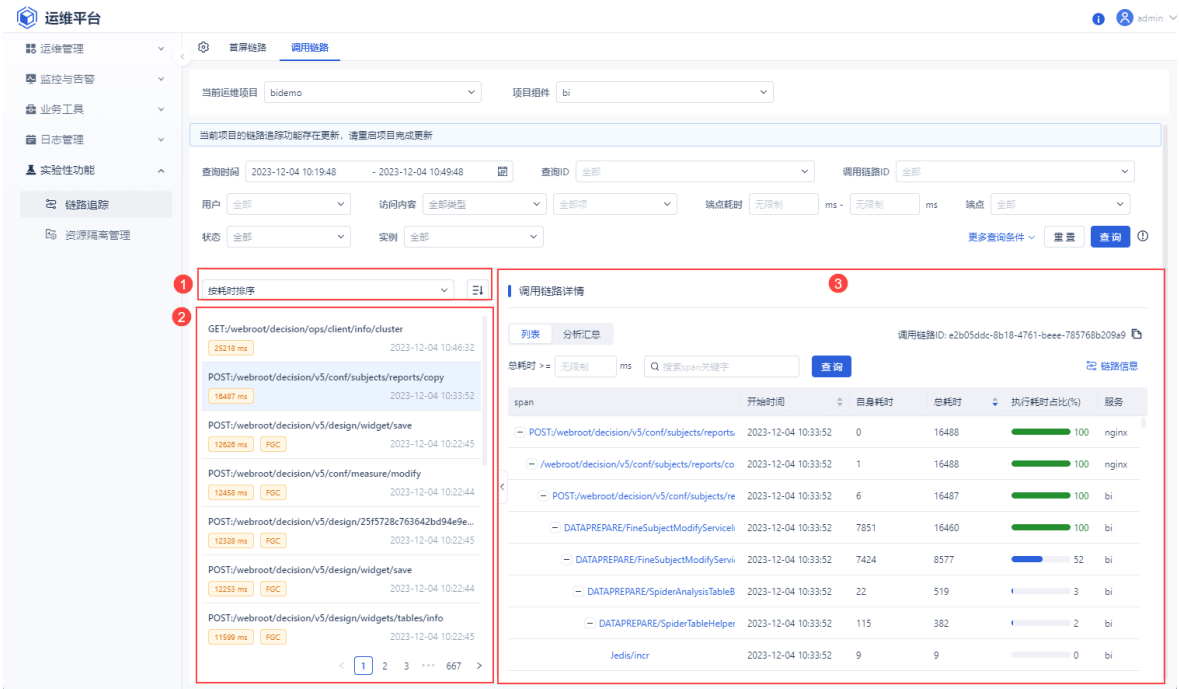
3 调用链路详情

使用前提：该运维项目使用 elasticsearch 套件获取和存储日志

使用说明：点击按钮将下载包含对应 TraceID 的运维日志到本地

4) 查看首屏链路

若某个调用链路所在 session 有相关首屏链路数据，则支持联动跳转查看首屏链路



3.2.1 列表

1) traceID 复制按钮：点击可快速复制 traceID，用于日志查询。

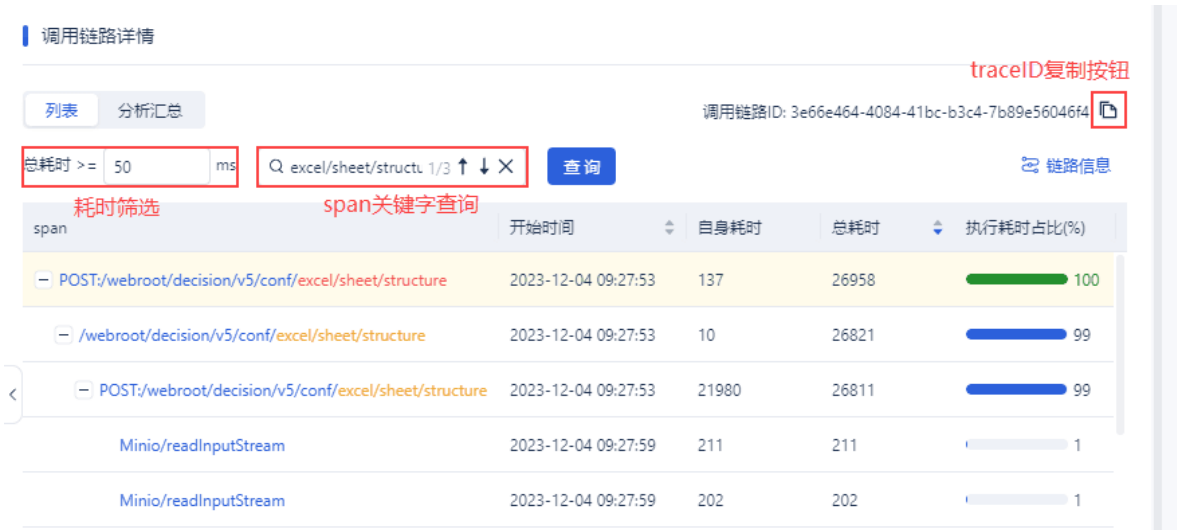
2) 获取日志按钮

使用前提：该运维项目使用 elasticsearch 套件获取和存储日志

使用说明：点击按钮将下载包含对应 TraceID 的运维日志到本地

3) 耗时筛选：支持使用控件筛选满足耗时范围的链路

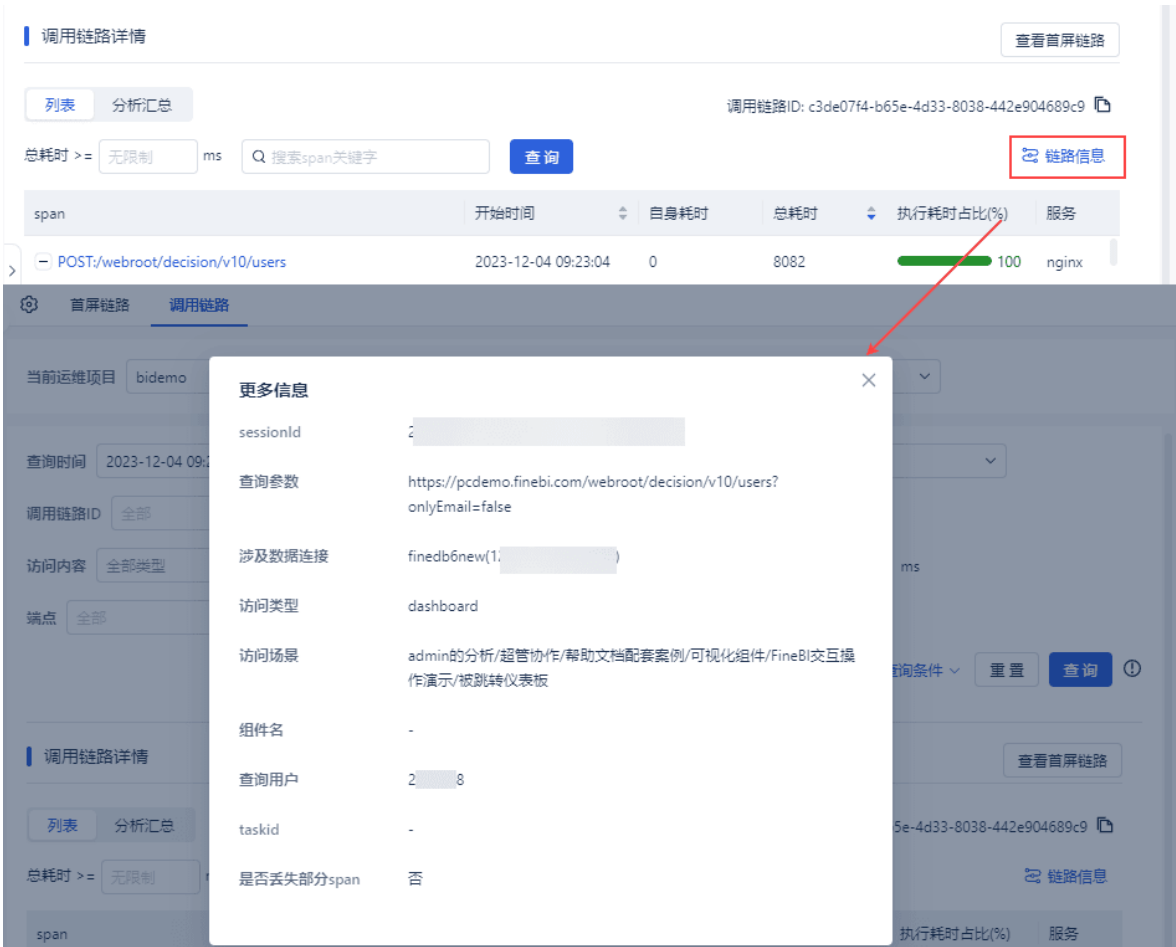
4) span 关键字查询：支持搜索包含指定关键字的 span



5) 以明细表形式展示该链路的明细 span 信息，点击可展示每条 span 的更详细信息。



6) 点击「链路信息」按钮，无需逐条查看 span，即可展示系统自动提炼的调用链路的最关键信息。



3.2.2 分析汇总

分析汇总界面分为两个部分：

1) 耗时分布：帮助用户一眼看到耗时分布的情况

分为两类：按 peer 取数耗时、其他

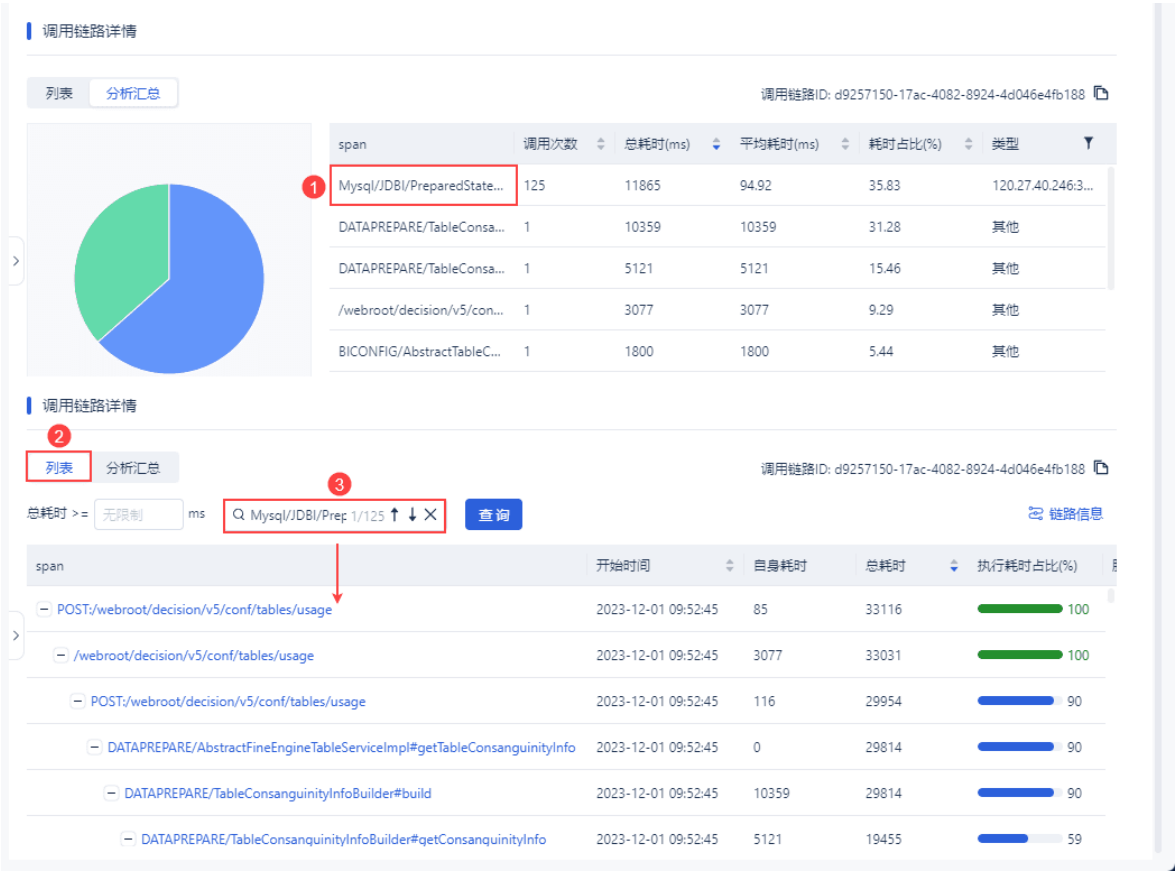
点击具体类别可以联动筛选下面的 span 汇总

2) span 汇总：以汇总维度展示链路 span 的汇总信息

内容	说明
span	span 信息
调用次数	调用的次数
总耗时	所有调用的耗时总和，单位 ms
平均耗时	总耗时/调用次数，单位 ms
耗时占比	在所有耗时中，该 span 的耗时占比
	按照链路耗时占比降序排列
类型	peer 取数耗时/其他



3) 支持点击 span 名称跳转至「列表」视图下，以名称作为关键词进行搜索。



## 9.1.4 非容器化部署项目接入链路追踪

本文涉及模块属于「实验性功能」，此类功能使用门槛较高，功能仍在进行打磨优化中。

若您有相关实际使用场景，或需获取使用帮助、反馈功能相关建议和评价，付费用户请联系技术支持。

技术支持联系方式：前往「[服务](#)」，选择「在线支持」、电话「400-811-8890」

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.5.5	-
V1.5.7	1) 非容器化项目，启用链路追踪功能步骤简化 2) 支持关闭链路追踪功能

#### 1.2 功能简介

运维平台提供「[链路追踪](#)」功能，但该功能默认仅对容器化部署的应用生效。

本文将介绍如何将非容器化部署的 FineReport/FineBI 项目，接入链路追踪功能。

### 2. 操作步骤

#### 2.1 启用运维组件

链路追踪功能，由这些运维组件提供：elasticSearch、skywalking\_oap。

1.5.7 及之后版本，这些组件默认启动，用户可手动关闭组件。1.5.6 及之前版本，这些组件默认不启动。

如需使用「链路追踪」功能，必须先确保已启动这些组件。否则提示「链路追踪组件未启用，请在运维组件目录中启动 elasticsearch 以及 skywalking\_oap」。如下图所示：





管理员登录运维平台，点击「运维管理>运维组件」，先启动 elasticsearch 组件，再启动 skywalking\_oap 组件。



## 2.2 确认端口互通

运维平台的 skywalking\_oap 组件，负责链路追踪的数据上报和分析整理服务。因此必须确保 FineReport/FineBI 项目所在服务器，支持访问运维平台的 skywalking\_oap 组件端口。

skywalking\_oap 组件默认端口为：{11800:11800}、{12800:12800}

## 2.3 新增项目

管理员需要先将待接入链路追踪的非容器化部署的 FineReport/FineBI 项目，新增到运维平台的「项目管理」中。如下图所示：

新增项目的操作方法请参见：[接入已有非容器化项目](#)

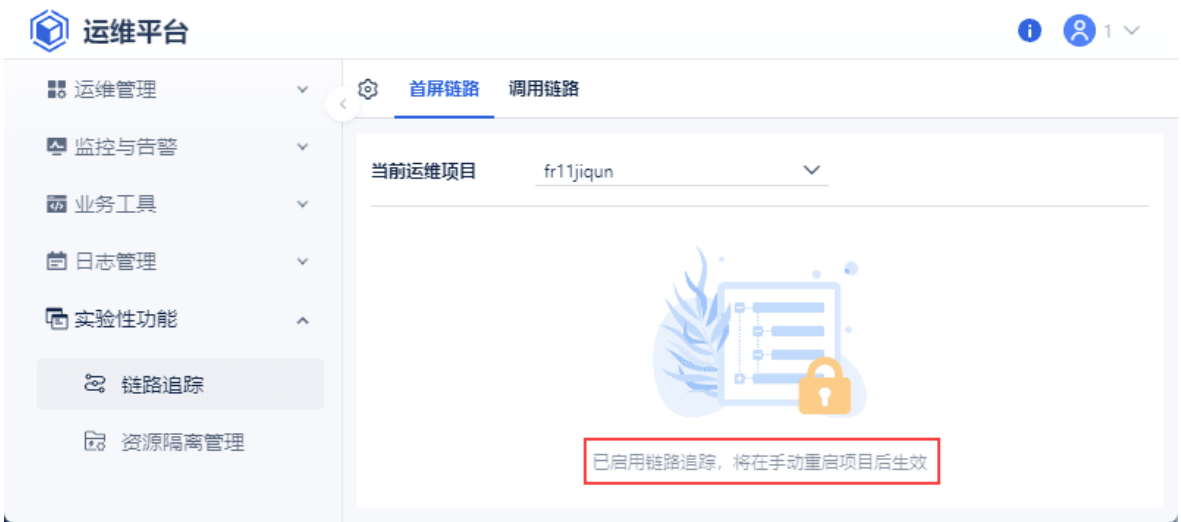


2.4 启用链路追踪

- 1) 管理员访问「链路追踪」相关界面，选择待启用链路追踪的非容器化项目。
- 2) 界面显示「当前未启用链路追踪」，点击「启用链路追踪」。
- 3) 跳出提示「请手动重启项目，该项目在开启链路追踪后需手动重启」，点击「我知道了」，开始自动启用链路追踪。



- 4) FineOps 运维平台会自动执行启用步骤  
自动传输 skywalking-java-agent 文件到 FineReport/FineBI 工程的 WEB-INF 文件夹下  
自动为 FineReport/FineBI 工程添加 skywalkingAgent 相关的启动参数
- 5) 启用步骤执行完成后，提示「已启用链路追踪，将在手动重启项目后生效」。



## 2.5 重启项目

参考 [关闭或重启 FineBI 工程](#) / [关闭或重启 FineReport 工程](#) 文档，重启 FineReport/FineBI 项目。

即可将该非容器化部署的 FineReport/FineBI 项目，接入链路追踪功能。

注：FineReport/FineBI 工程下的所有节点，都需要依次进行以上操作。

## 3. 注意事项

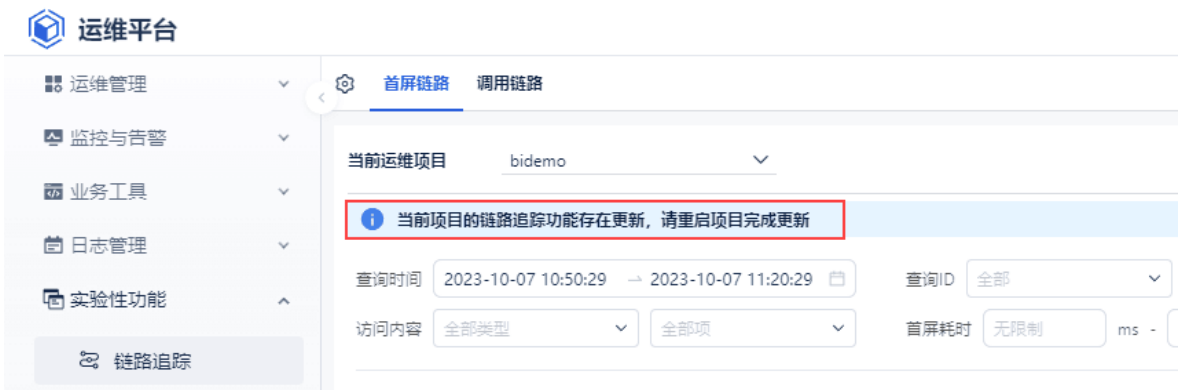
### 3.1 升级链路追踪依赖文件

skywalking-java-agent 文件会持续更新升级。

管理员登录 FineOps 运维平台时，会自动检测非容器化项目是否存在 skywalking-java-agent 文件更新。

若需要更新，会自动进行更新。更新成功后，仍需要重启 FineReport/FineBI 项目才能生效。

若更新了文件但未重启项目，使用该容器化项目的链路追踪功能时，提示「当前项目的链路追踪功能存在更新，请重启项目完成更新」。



### 3.2 关闭链路追踪

FineOps 运维平台支持对项目停用链路追踪功能。

- 1) 管理员登录运维平台， 点击「实验性功能>链路追踪>全局设置」。
  - 2) 选择待关停链路追踪的项目， 点击「关闭链路追踪」。
  - 3) 参考 [关闭或重启 FineBI 工程](#) / [关闭或重启 FineReport 工程](#) 文档， 重启 FineReport/FineBI 项目。
- 即可成功关闭当前运维项目的链路追踪功能。



## 9.2 资源隔离管理

本文涉及模块属于「实验性功能」，此类功能使用门槛较高，功能仍在进行打磨优化中。

若您有相关实际使用场景，或需获取使用帮助、反馈功能相关建议和评价，付费用户请联系技术支持。

技术支持联系方式：前往「[服务](#)」，选择「在线支持」、电话「400-811-8890」

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.5.5	-

#### 1.2 功能简介

集群工程存在多个节点，每个节点下都有很多资源模板。

运维平台提供「资源隔离管理」功能，支持将模板与节点绑定，实现资源隔离，以此来减少不同业务间的影响。



#### 1.3 名词说明

「业务组」：专门用于执行某一种或某多种业务的虚拟组织，组织内以应用节点为最小单位，可以多个节点一个业务组。

「公用节点」：帆软应用集群中，没有包含在任何「业务组」内的节点，主要用于通用业务或者应急保障等场景。

## 2. 开启业务组

### 2.1 使用前提

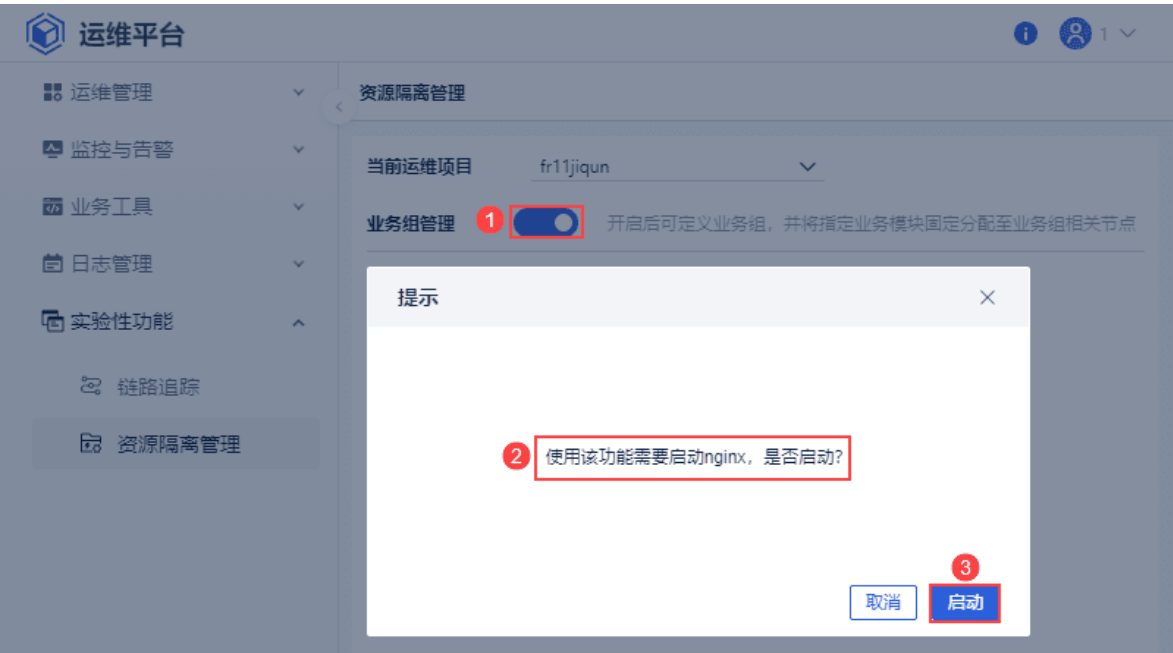
1) 节点数小于 2 时，不可开启业务组管理，因为此时无法设置「公用节点」。

提示「节点数小于 2，不可添加业务组」



2) 运维项目环境下不存在 Nginx，或存在的 Nginx 与运维项目前缀不一致。

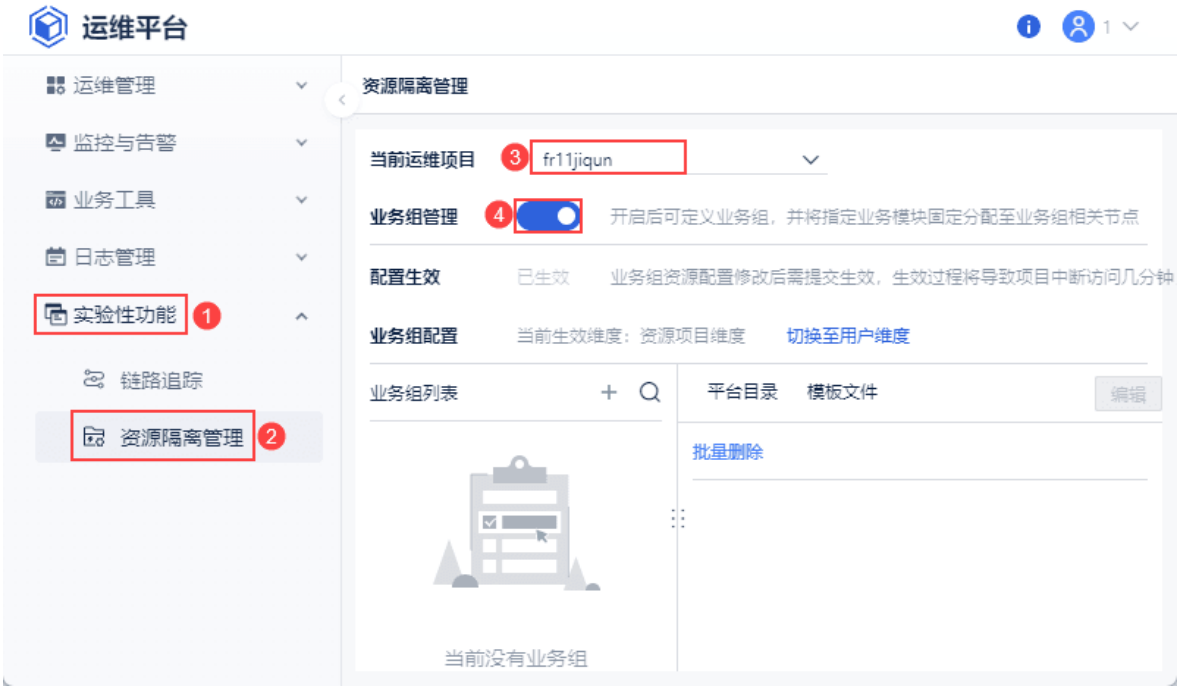
可快速完成 Nginx 启动与接入后，方可正常使用业务组管理功能。详情请参见：[业务组管理配置 Nginx 组件](#)



### 2.2 开启业务组管理

管理员登录运维平台，点击「实验性功能>资源优先级管理>资源隔离管理」。选择项目，开启「业务组管理」按钮。

开启后可定义业务组，并将指定业务模块固定分配至业务组相关节点。

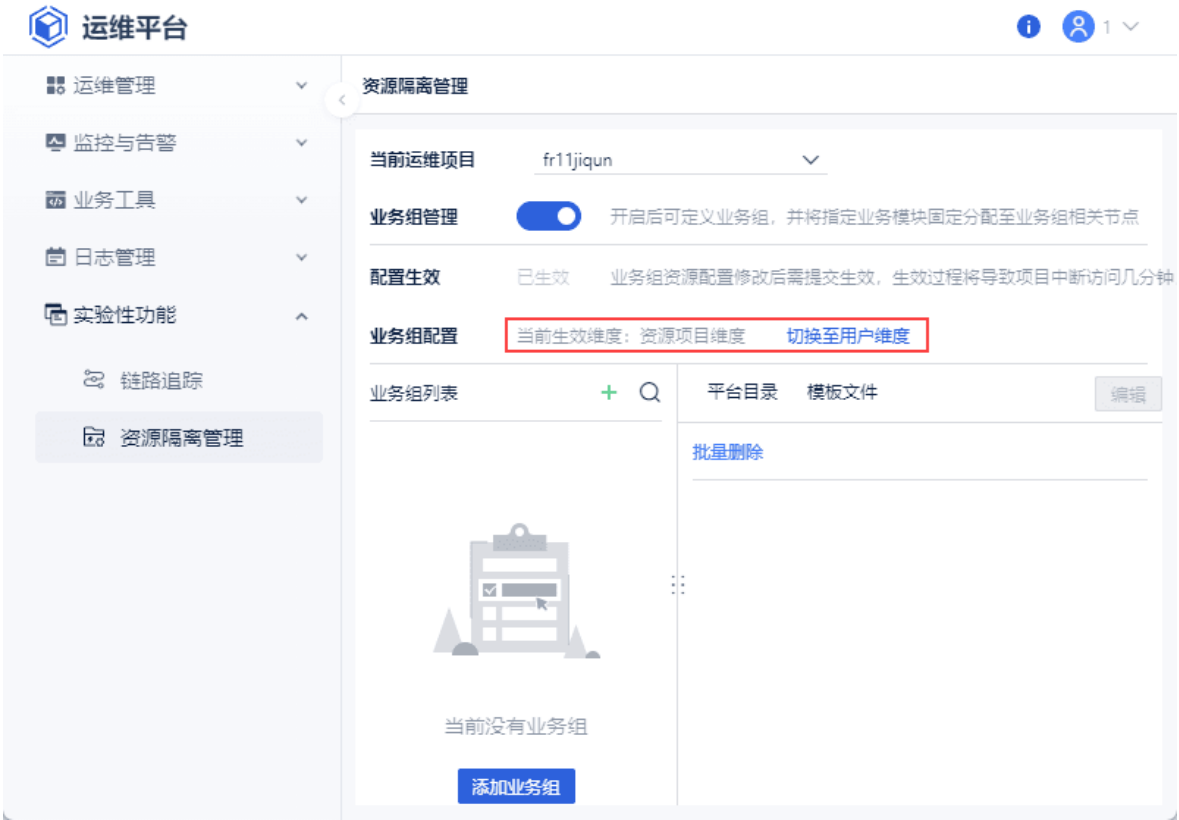


### 3. 配置业务组

#### 3.1 选择生效维度

业务组配置支持两种维度，同时只能有一种维度生效。（切换维度后，原维度配置失效）

- 资源项目维度：是将平台目录、模板文件，放置到业务组中。
- 用户维度：是选择「平台用户」作为业务组中的资源。



#### 3.2 新增业务组

点击「+」号/「添加业务组」按钮，配置业务组信息后，点击「确定」，即可新增业务

组。如下图所示：

配置项	说明
业务组名称	自定义业务组名称
节点	选择节点，支持多选，但不可超出节点数上限
	指定节点资源，单选
节点资源	1) 业务组独占：表示其他资源/用户访问不会进入此业务组
	2) 共享资源：表示其他资源/用户访问也会使用此业务组
	设置当业务组节点非存活状态后、业务组的处理策略，单选
	1) 转入共享节点：业务组节点全部非存活状态后，该业务组的资源/用户访问，转入其他共享节点中
异常策略	2) 阻止访问：业务组节点全部非存活状态后，该业务组的资源/用户访问不转入其他节点，直接无法访问



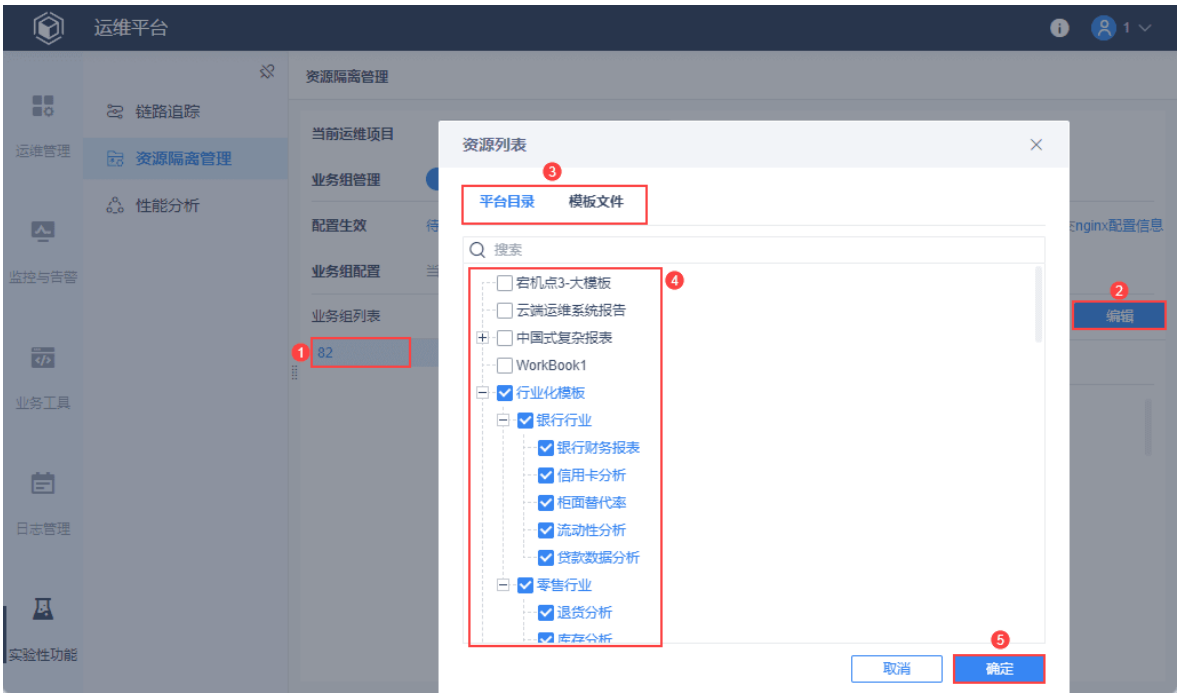
对于已有的业务组，支持查看业务组信息、修改业务组、删除业务组。如下图所示：





### 3.3 配置业务组中的资源

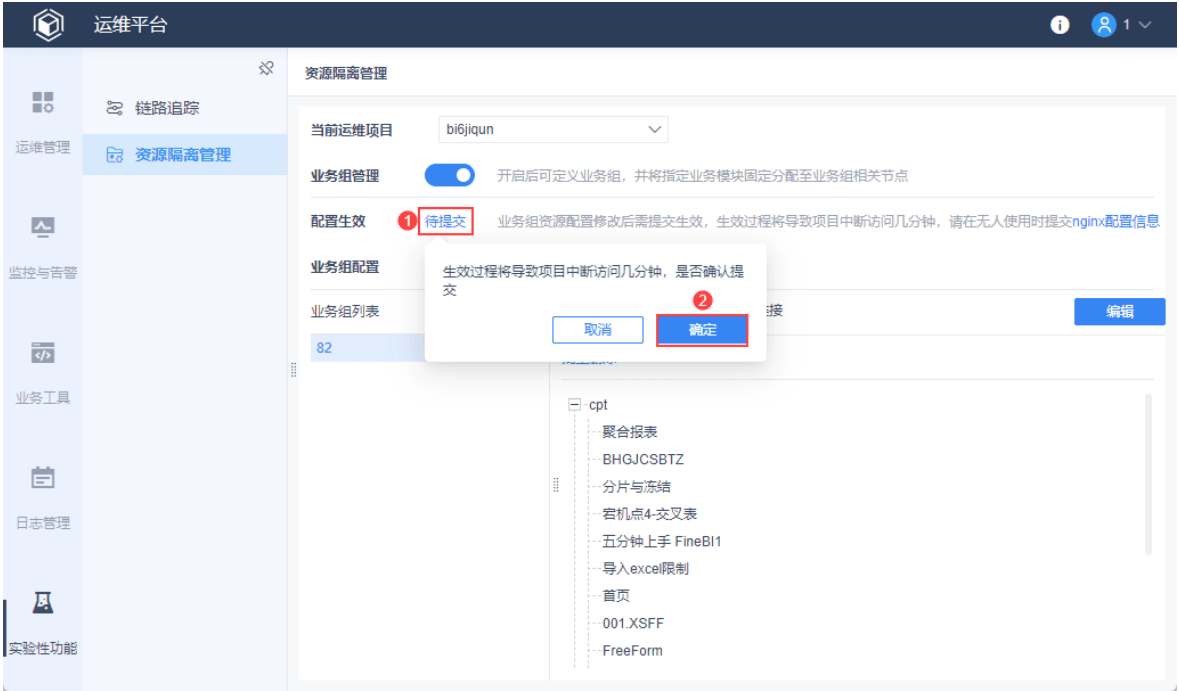
选中业务组，点击「编辑」按钮，支持选择「平台目录、模板文件/用户」作为该业务组中的资源。



### 3.4 提交配置

业务组资源配置修改后需提交方可生效，生效过程将导致应用中中断访问几分钟，请在无人使用时提交。

资源与业务组绑定后，指定模板/用户分发到指定节点，以保证重要业务不受影响。



## 9.2.1 业务组管理配置 Nginx 组件

本文涉及模块属于「实验性功能」，此类功能使用门槛较高，功能仍在进行打磨优化中。

若您有相关实际使用场景，或需获取使用帮助、反馈功能相关建议和评价，付费用户请联系技术支持。

技术支持联系方式：前往「[服务](#)」，选择「在线支持」、电话「400-811-8890」

### 1. 概述

#### 1.1 版本

运维平台版本	功能变更
V1.5.5	-

#### 1.2 功能简介

运维平台提供「业务组管理」功能，帮助用户对工程资源进行管理。

但使用该功能的前提，是运维项目下必须存在与运维项目前缀一致的 nginx。

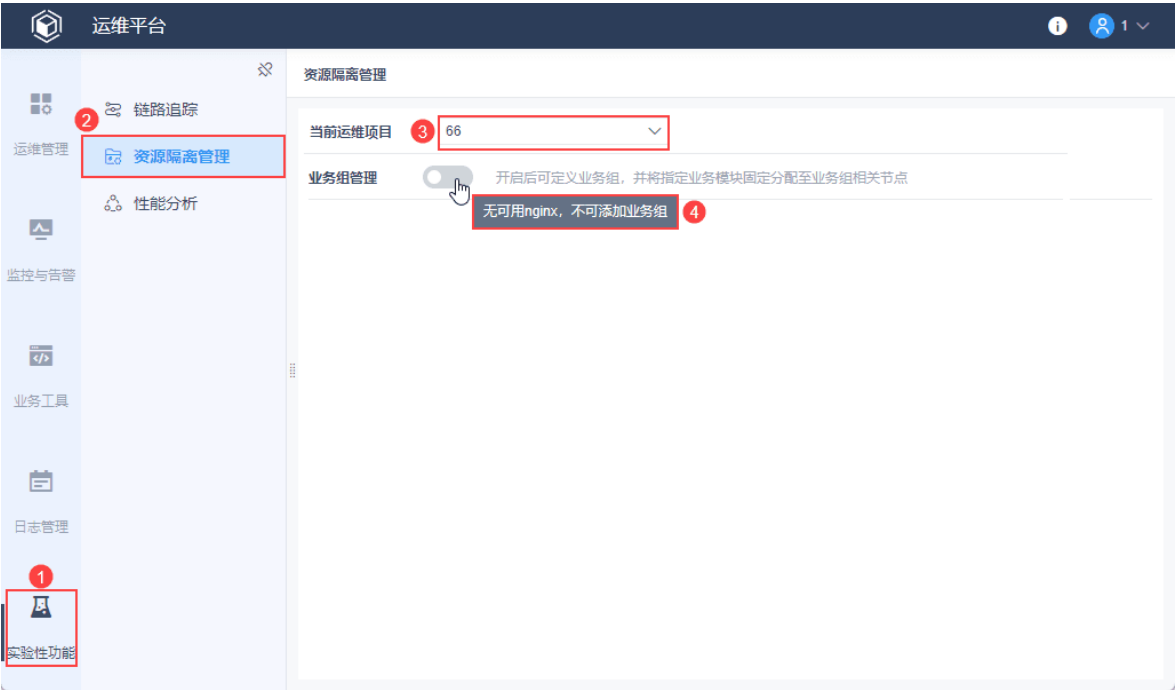
本文将介绍如何快速完成运维项目下的 Nginx 启动和接入。

### 2. 操作步骤

#### 2.1 查看是否存在可用 Nginx

管理员登录运维平台，点击「实验性功能>资源隔离管理」，选择运维项目。

鼠标悬浮于「业务组管理」按钮上，提示「无可用 Nginx，不可添加业务组」。



## 2.2 启动 Nginx

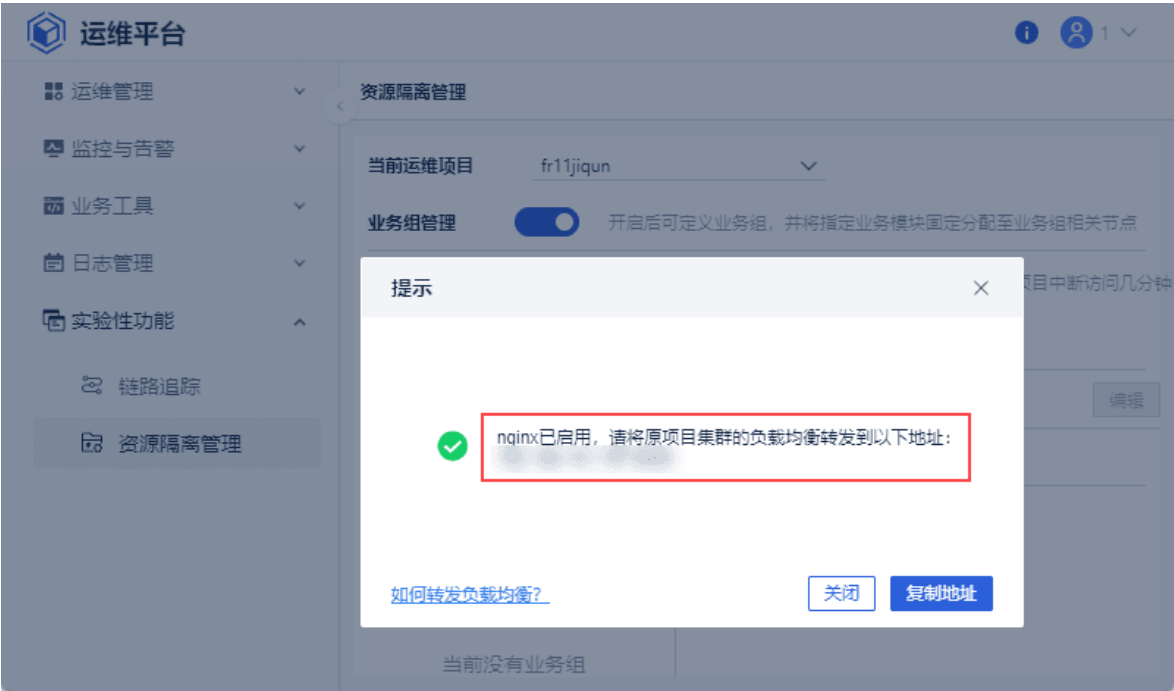
点击「业务组管理」后，弹出提示框「使用该功能需要启动 Nginx，是否启动？」。

点击「启动」，将在运维平台所在节点启动一个新的 Nginx 容器，并自动完成接入当前项目各个集群节点的动作，如下图所示：



新 Nginx 启动完成后，返回新 Nginx 的应用访问地址。

弹出提示框「Nginx 已启用，请将原项目集群的负载均衡转发到以下地址：ip:port」，如下图所示：



## 2.3 获取新 Nginx 配置信息

新 Nginx 启动完成后，可点击「Nginx 配置信息」，获取到原项目集群 Nginx 配置路径和新 Nginx 应用访问地址。



## 2.4 修改原集群 nginx.conf

根据 Nginx 配置路径，找到原项目集群 Nginx 配置下的 nginx.conf 文件并修改。

- 1) 注释掉原有的随机转发请求。
- 2) 注释掉原有的健康检查。
- 3) 新增上一节获取的 Nginx 应用访问地址为随机转发节点。
- 4) 重启原项目集群 Nginx 组件。

```
32
33     upstream FR.com {
34         #server 192.168.61.250:8080 max_fails=15 fail_timeout=300s;
35         #server 192.168.61.251:8080 max_fails=15 fail_timeout=300s;
36         server 192.168.61.251:9049
37
38     #<-----ngx_healthcheck_module----->#
39         #check interval=2000 rise=5 fall=10 timeout=10000 type=http;
40         #check_http_send "GET /webroot/decision/system/health HTTP/1.0\r\n\r\n";
41         #check_http_expect_alive http_2xx http_3xx;
42     #<-----ngx_healthcheck_module----->#
43     }
44
45     upstream WBS.com {
46         server 192.168.61.250:38888 max_fails=15 fail_timeout=300s;
47         server 192.168.61.251:38888 max_fails=15 fail_timeout=300s;
48         ip_hash;
49     }
50     server {
51         listen      80;
52         server_name localhost;
53         underscores_in_headers on;
54         #charset koi8-r;
55         #access_log logs/host.access.log main;
56
57     #<-----Nginx upstream status monitor----->#
58         #location /status {
59             #    healthcheck_status html;
60             #}
61     #<-----Nginx upstream status monitor----->#
```

## 2.5 重启 Nginx

nginx.conf 文件修改完成后，需要重启原项目集群 Nginx 组件，配置方可生效。

重启成功后，运维项目集群原本的负载均衡，即可接在新 Nginx 容器上。