

# 品牌简介

腾讯蓝鲸智云，简称蓝鲸，是腾讯互动娱乐事业群（Interactive Entertainment Group，简称 IEG）自研自用的一套用于构建企业研发运营一体化体系的 PaaS 开发框架，提供了 aPaaS（DevOps 流水线、运行环境托管、前后台框架）和 iPaaS（持续集成、CMDB、作业平台、容器管理、数据平台、AI 等原子平台）等模块，帮助企业技术人员快速构建基础运营 PaaS。

传统的 Linux 等单机操作系统已发展数十年，随着云时代的到来，企业所需资源数暴增，操作节点（物理或虚拟服务器及容器）数量普遍达到数千个，大型互联网公司甚至达到百万级别，混合云模式成为常态，虽然 IaaS 供应商的出现从一定程度上解决了资源切割调度问题，但并未很好的解决资源与应用的融合，企业需要一种介于 IaaS 与应用（SaaS）之间的层级，用于屏蔽及控制 IaaS，快速开发及托管 SaaS，我们将其称之为基础 PaaS 层，并着重发展用于研发及托管企业内技术运营类 SaaS 的基础运营 PaaS，并将其作为区别于传统 OS 的下一代企业级分布式运营操作系统。

企业 IT 应用的全生命周期可划分为研发、运维、运营三段，在各行业进行互联网化转型的过程中，融入敏捷思维，即形成持续集成、持续部署、持续运营的概念（CI-CD-CO）。

为降低转型成本，不以增加人力数量为转型前提，腾讯 IEG 以运维团队作为转型起点，充分利用这一群体低价值重复性工作占比高的特点，从 CD 领域切入，以 PaaS 技术进行运维自动化领域的烟囱治理，形成运维 PaaS 体系。将自动化所释放的人力资源，转型为运维开发团队，利用 PaaS 的自增长属性，将运维 PaaS 逐步向 CI 及 CO 拓展，最终完成企业级研发 - 运维 - 运营基础 PaaS 构建，落地企业研发运营一体化。

腾讯蓝鲸智云秉承开放共赢的理念，招募及联合国内优质合作伙伴，以改变中国运维行业为起点，致力于推动国内企业借助研发运营一体化，低成本实现企业 IT 经营管理模式升级和自主化。

# 版本类别

腾讯蓝鲸智云有 2 个主要版本线：社区版和企业版，根据软件包的功能和技术服务提供方的不同，细分为软件基础包、软件增强包、合作商软件包。各版本的描述和产品信息详见下表：

版本类别	详细描述
社区版	可以从腾讯蓝鲸智云官方网站下载，并免费使用，用户以运维社区的个人/企业为主，通过社区力量，协助腾讯蓝鲸更好地验证产品的功能和性能。目前该版本的软件由基础平台（管控平台、配置平台、作业平台、PaaS 平台）和官方提供的 SaaS（节点管理、日志检索、蓝鲸监控、故障自愈、标准运维等）组成，且 PaaS 平台上可运行基于腾讯蓝鲸统一开发框架研发的各种场景 SaaS。
企业版	企业版是付费软件，通过购买 License 的方式永久授权。用户以企业客户为主。其软件产品在功能、性能、服务等方面均优于社区版。蓝鲸的“企业版技术合作伙伴”为软件销售和实施的授权方。

## 软件包类别

软件包类别	详细描述
基础包	基础包是企业内落地自动化运维解决方案的基础软件，包含基础平台（管控平台、配置平台、作业平台、PaaS 平台）和官方提供的 SaaS（如：节点管理、蓝鲸监控、故障自愈、标准运维、日志检索 等）。
增强包	增强包是更高级的运维产品的集合，必须基于基础包一起使用。有蓝鲸数据平台、容器管理平台、蓝盾等。根据企业需求，可以提供给客户使用。在社区版中，目前增强包有容器管理平台。
合作商软件包	插件包是由腾讯蓝鲸合作伙伴提供的运维解决方案场景类的产品，如 Wman、OSKit、网络管理、CICDKit 等，该类产品可在已有基础包的前提下，以“插件”的方式，面向用户提供更多功能场景。由于开发者/技术服务提供方均非腾讯蓝鲸团队，涉及到合作、销售、推广等事宜需要与软件提供方商榷。

- 社区版中的部分产品已开源（如：[配置平台](#)、[PaaS 平台](#)、[标准运维](#)等），其 GitHub 项目的 release 包与社区版最新的版本包保持一致，使用过程中，请遵循《蓝鲸智云软件用户服务协议》和开源项目协议。
- [蓝鲸企业版在线体验申请信息](#) | 蓝鲸智云团队面向企业出售技术解决方案，将由腾讯蓝鲸的技术合作伙伴提供专属的企业服务。请填写在线体验申请信息，工作人员将在 3-5 个工作日内与您取得联系。

## 部署脚本

### bkcec 说明

bkcec 是蓝鲸社区版的安装维护主脚本。

“

bk\_install 是封装了 bkcec 的集成安装脚本。

”

bkcec 的调用语法为：`bkcec <command> <module> [project] [1]`

其中可用的 command 有：

- sync:** 从中控机的 src/ 同步安装 `<module>` 依赖的文件和目录到对应机器的 src/ 下。
- install:** 安装 `<module>` 到 `$INSTALL_PATH`(默认 /data/bkce)下。
- install 1:** 等同于先执行 `sync <module>`，再执行 `install <module>`，合二为一。
- initdata:** 安装后初始化，常用于模块的用户创建，sql 导入，权限设置等操作。

- **render:** 渲染模块的模板文件，install 过程中会调用它，做模板变量替换。
- **status:** 检查模块的进程是否运行。
- **start:** 启动模块的进程。
- **stop:** 停止模块的进程。
- **upgrade:** 升级更新模块。

其中模块名，对应 install.config 里的名称，当前社区版包含以下模块，按模块特点分为两类：

- 开源外部组件。
- 蓝鲸自研产品。

## 开源外部组件

### 基础部分：

- **consul:** 分布式服务发现和域名服务。
- **mysql:** 关系数据库管理系统。
- **redis:** 键值对存储数据库。
- **nginx:** web 服务器和反向代理。
- **mongodb:** 面向文档的数据库管理系统。
- **zk:** ZooKeeper 分布式配置服务。
- **rabbitmq:** 消息队列服务。
- **es:** ElasticSearch 分布式搜索和数据分析引擎。
- **kafka:** 分布式数据流处理服务。
- **beanstalk:** 消息队列服务。

### BCS (容器管理平台)部分：

- **web\_console:** 是容器服务提供快捷查看集群状态的命令行服务。
- **cc:** 容器管理平台配置中心。
- **monitor:** 容器监控后台-用于产生告警。
- **thanos(rule):** 全局监控告警策略。
- **devops(navigator):** 提供产品导航统一框架。
- **grafana:** 容器监控后台-用于可视化展示。

- **devops(pm):** 项目管理后台模块。
- **harbor(api):** 是蓝盾 bcs 容器服务查询 docker 镜像信息的中间件，目前只支持 harbor 镜像查询，后面有计划支持其他 docker 仓库(jfrog /腾讯云等)。
- **harbor(server):** Harbor 是 Vmware 公司开源的企业级 Docker Registry 管理项目。
- **bcs(api):** apiserver 是 bcs 容器管理平台的总接入口，是 bcs 对外暴露的唯一入口。主要功能包括：路由，服务发现，负载均衡，熔断，限流，鉴权等功能。
- **bcs(storage):** bcs-storage 作为 bcs 的动态数据管理，负责存储来自 mesos 和 k8s 集群上报的资源动态数据，并提供相应的查询 API 接口。
- **bcs(health-master):** health-master 负责整个监控服务的源数据拉取，为 health-slave 提供任务分发功能。同时负责数据的分析、聚合、处理等。
- **bcs(dns-service):** 为 bcs service 下的 mesos 集群提供 DNS 服务。
- **bcs(health-slave):** health-slave 负责：任务获取->执行->上报执行结果。
- **bcs(ops):** 容器服务调用标准运维作业。

## 蓝鲸自研产品

- **paas:** 蓝鲸 PaaS 平台。
- **cmdb:** 蓝鲸配置平台。
- **job:** 蓝鲸作业平台。
- **gse:** 管控平台后台。
- **license:** 证书服务。
- **appo:** SaaS 部署的正式环境。
- **appt:** SaaS 部署的测试环境。
- **bkdata:** 数据平台基础服务。
- **dataapi:** 数据平台 API 接口服务。
- **databus:** 数据平台总线服务。
- **monitor:** 监控后台。
- **fta:** 故障自愈后台。

启动蓝鲸监控后台服务示例：`./bkcec start bkdata monitor`。

## 文件用途

## 脚本包的文件说明

- **agent\_setup/**: 存放安装 Agent 相关的脚本以及脚本模版。
- **appmgr/**: 存放构建 SaaS 和运行 SaaS 所需的脚本，社区版采用 virtualenv 方式。
- **bcs/**: 存放 bcs 构建所需的脚本。
- **bk\_install**: 封装 bkcec 用来做集成安装部署。
- **bkcec**: 安装和维护的主脚本。
- **bkco.env bkco.fc bkco\_install**: 网络管理模块所用的安装部署相关环境变量和函数定义。
- **clean.fc**: 卸载清理相关函数。
- **configure\_ssh\_without\_pass**: 一键配置免密 ssh 登陆用的脚本。
- **control.rc**: 定义进程启停的函数库。
- **crontab.rc**: crontab 新增和删除相关的函数定义。
- **deck/saas.py**: 命令行部署 SaaS 应用的工具。
- **deliver.rc**: 定义从中控机同步文件目录到对应模块机器的函数。
- **dependences.env**: 各模块依赖的命令和 rpm 包定义。
- **errors.env**: 错误码定义。
- **functions**: 通用的 bash 函数。
- **globals.env**: 定义蓝鲸组件用到的全局变量配置。
- **health\_check/check\_proc\_exists**: Consul 使用的健康检查脚本。
- **install.config**: 定义模块在主机的分布部署方式。
- **install.config.3ip.sample**: 部署基础不含bcs时使用。
- **install.config.bcs.sample**: 完整的基础平台+bcs 方案分布。
- **install.config.bcs.single.sample**: 单机部署 bcs 模块分布示例。
- **install.config.new.sample**: 部署 bcs 时使用。
- **install\_minibk**: 最小化单机部署的封装脚本。
- **migrate/**: 蓝鲸套件迁移升级的脚本，目前没有用。
- **parse\_config**: 解析 install.config 并生成 Consul 主配置和服务定义的配置，`config.env` 等文件。
- **pip/**: 存放安装脚本所需的 pip 依赖包。
- **ports.env**: 定义所有组件需要设定的端口信息。

- **process\_watch + watch.rc:** 进程监控脚本。
- **register.rc:** 自动注册主机到 CMDB 的函数。
- **scripts/:** gse\_agent 和 gse\_proxy 安装所需要启停脚本，在安装 GSE 时会自动打包进去。
- **status.rc:** 检查进程状态相关的函数。
- **summary.rc:** 打印相关模块和所有模块的版本信息。
- **templates/:** 安装脚本初始依赖的配置模版，目前只是 Consul 的 Supervisor 配置。
- **templates\_render.rc:** 渲染配置模版，根据读取配置文件和动态生成的变量去替换模版里的占位符。
- **update.rc:** 一些更新配置或者证书的操作定义。
- **upgrade.rc:** 各模块的升级操作封装。
- **utils.fc:** 安装、初始化主要函数都在这里先找。
- **RELEASE.md:** 版本日志。
- **VERSION:** 版本号。

## 完成安装或日常升级后生成的文件

- **.agreed:** 同意安装协议后的标记文件。
- **.app.token:** 后台程序和 SaaS，访问 ESB 接口时需要使用的 app\_code 和 app\_token 的键值对。
- **.bk\_install.step:** 集成安装的步骤进度标记文件。
- **.controller\_ip:** 中控机的 IP 地址。
- **.migrate/:** 成功导入的 sql 文件的标记文件。使用了 chatr +i，具有不可删除属性。
- **.path:** 蓝鲸的安装路径。
- **.rcmdrc:** 使用 rcmd()函数远程执行命令时，会加载的 rc 文件。
- **/tmp/bkc.log:** bkcec 等脚本输出的日志。

## 软件下载

安装软件需要 **下载** 2 个文件：软件包、证书。

资源下载

完整包下载

分包下载

证书

完整包下载

蓝鲸社区版目前最新的稳定版本为： 5.1.26  
完整包中，包含了最新的部署脚本，产品软件，开源组件。  
**适用于新装环境及新用户使用**

5.1.26 <a href="#">稳定版</a>	2020-01-04	2.3G	MD5: 15e384c9f420b245f872fa2c27419ba3 <a href="#">↓</a>
5.0.4-beta	2019-02-08	1.6G	MD5: 905e69217b2d4e95840c83db7b1e6a03
4.1.16 <a href="#">稳定版</a>	2019-02-26	1.4G	MD5: fe13cd372df2cfb664977463b6638a0a
4.1.15	2019-01-15	1.5G	MD5: fffc99dddc4e5ba9fd4ac92a74afc27b
4.1.13	2019-01-15	1.5G	MD5: 60c2ba14c25255cbd7835a356b69511c
4.1.12	2019-01-15	1.5G	MD5: ad36dc8c3f9e96fbe1214cad9ec7a33e
4.0.15	2019-01-15	1.2G	MD5: bacb497d9865faad19fc247885b75fbd

[产品白皮书](#)[部署文档](#)

### • 软件包(完整包和分包)

为了方便用户获取社区版软件包，提供了 2 种下载模式：**完整包下载**、**分包下载**。

所以，社区版的“完整包”=“部署脚本”+“产品软件”+“开源组件”。用户可以根据自己的需要进行下载。

“

**部署脚本**: 用于自动化安装和维护蓝鲸软件产品，属于蓝鲸软件的通用“安装和维护工具”，可以用于安装维护不同版本的产品软件，和不同版本的开源组件。

**产品软件**: 是蓝鲸所有自研产品/服务的集合，包括平台级产品和 SaaS 级产品。用户可以根据需要选择更新整个产品，或者更新部分模块。

**开源组件**: 提供了蓝鲸软件在使用过程中依赖的所有开源组件和配置模板，可根据需要选择下载，若涉及到开源组件的版权信息，请遵守开源协议要求。

”

说明：首次安装用户，建议下载完整包。

### • 证书

软件安装过程中，需要提供 **证书文件**，才能确保软件的正常启动，最多输入 3 个 MAC 地址。

## 机器评估

对于蓝鲸部署所需的硬件配置选型，并无规定。

蓝鲸由众多开源组件和自研组件构成。

开源组件的硬件选型可以参考相应的官方文档。

蓝鲸产品本身的建议配置如下：

名称	配置
PaaS	2核 4G
CMDB	2核 2G
JOB	2核 4G
BKDATA	4核 12G
FTA	1核 2G

如果硬件资源富余，可以一开始拆分搭建部署。若硬件资源不足，一开始可以混合搭建，注意观测资源消耗情况，可以适时增加机器，迁移模块的方式来保证整体的可用性。

资源规划是一个复杂的、动态的过程，更像是一门艺术而不是科学。

这里给出的一个比较合理的初始配置，基于以下考虑：

1. 分布式模块达到高可用至少三个节点，所以至少需要三个 OS (物理机或虚拟机均可)。
2. BKDATA 是耗费资源最多的蓝鲸组件。请分配到 4 核 16G 以上的机器。
3. 若日志检索，蓝鲸监控是主要使用场景，请给 influxdb 和 elasticsearch 模块更多的内存，更好磁盘性能。如 SSD。
4. Nginx 模块所在的机器需要有对外提供服务，可访问的 IP。这是蓝鲸平台的总入口。
5. 如果有跨云管理需求，GSE 部署的机器需要有跨云的网络条件。

根据以上考虑，安装蓝鲸初始配置，请满足：

- 1台 4核 16G
- 2台 4核 8G

## 安装环境准备

开始安装蓝鲸社区版前，需按以下文档指南，做好准备工作。

**注意：**所有待安装蓝鲸的机器均需要按以下清单检查和操作。

### 获取安装包

蓝鲸社区版包含部署脚本、产品软件和开源组件。蓝鲸提供完整包与分包的下载通道，请自行到下载地址按需获取。新装环境及新用户建议下载完整包使用。

下载地址：<https://bk.tencent.com/download/>



下载完成后，请核对 MD5 码。

## 配置 YUM 源

在所有蓝鲸服务器上配置好 YUM 源，要求该 YUM 源包含 EPEL。

不能连外网 YUM 源的环境，可以配置一个内部的 YUM 源 或者本地 YUM 源。

## 本地 YUM 源

参考附录中的 [离线安装的配置方法](#)。

## 在线 YUM 源

推荐使用以下 YUM 源：

- [腾讯云 CentOS](#)
- [腾讯云 EPEL](#)

## CentOS 系统设置

准备好硬件，安装完原生 CentOS 系统后。我们需要对初始系统做一些配置，保证后续安装过程的顺畅和蓝鲸平台的运行。

**系统版本：** 要求 CentOS-7.0 以上版本，推荐 CentOS-7.5。

### 1. 关闭 SELinux

```
# 检查 SELinux 的状态，如果它已经禁用，可以跳过后面的命令
sestatus
```

可以使用以下命令禁用 SELinux，或者修改配置文件。

```
# 通过命令临时禁用 SELinux
setenforce 0

# 或者修改配置文件
sed -i 's/^SELINUX=enforcing/SELINUX=disabled/g' /etc/selinux/config
```

接着，重启机器：

```
reboot
```

## 2. 关闭默认防火墙(firewalld)

安装和运行蓝鲸时，模块之间互相访问的端口策略较多，建议对蓝鲸后台服务器之间关闭防火墙。

```
# 检查默认防火墙状态，如果返回 not running，可以跳过后面的命令
firewall-cmd --state
```

### 停止并禁用 firewalld

```
systemctl stop firewalld    # 停止 firewalld
systemctl disable firewalld # 禁用 firewall 开机启动
```

## 3. 安装 rsync 命令

安装脚本依赖 rsync 分发同步文件。

```
# 检查是否有 rsync 命令，如果有返回 rsync 路径，可以跳过后面的命令
which rsync

# 安装 rsync
yum -y install rsync
```

## 4. 停止并禁用 NetWorkManager

```
systemctl stop NetworkManager
systemctl disable NetworkManager
```

“

备注说明：该操作前提需确保主机为静态 IP，若为 DHCP 获取的 IP，则无法直接 disable NetworkManager，否则会出现主机重启后，或者主机运行一段时间 IP 租约地址到期后，网卡无法从网络重新正常获取 IP 地址的情况。

”

## 5. 调整最大文件打开数

```
# 检查当前 root 账号下的 max open files 值
ulimit -n
```

如果为默认的 1024，建议通过修改配置文件调整为 102400 或更大。

注意：limits.conf 初始文件的备份。

```
cat << EOF >> /etc/security/limits.conf
root soft nofile 102400
root hard nofile 102400
```

```
EOF
```

修改后，重新使用 root 登录检查是否生效。

## 6. 确认服务器时间同步

服务器后台时间不同步会对时间敏感的服务带来不可预见的后果。务必在安装和使用蓝鲸时保证时间同步。

```
# 检查每台机器当前时间和时区是否一致，若相互之间差别大于3s(考虑批量执行时的时差)，建议校时。
date -R

# 查看和ntp_server的时间差异(需要外网访问，如果内网有ntpd服务器，自行替换域名为该服务的地址)
ntpdate -d cn.pool.ntp.org
```

如果输出的最后一行 offset 大于 1s 建议校时。

```
# 和 ntp 服务器同步时间
ntpdate cn.pool.ntp.org
```

更可靠的方式包括通过运行 ntpd 或者 chrony 等服务在后台保持时间同步。具体请参考官方文档 [使用 ntpd 配置 NTP](#) 或 [使用 chrony 配置 NTP](#)。

## 7. 检查是否存在全局 HTTP 代理

蓝鲸服务器之间会有 HTTP 请求，如果存在 HTTP 代理，且未能正确代理这些请求，会发生不可预见的错误。

```
# 检查 http_proxy https_proxy 变量是否设置，若为空可以跳过后面的操作。
echo "${http_proxy}" "${https_proxy}"
```

对于本机配置 http\_proxy 变量的方式，请依次查找文件 /etc/profile、/etc/bashrc、\$HOME/.bashrc 等是否有设置。或者咨询网络管理员/IT 部门协助处理。

在这些主机中，选择任意一台机器作为蓝鲸的运维中控机。之后的安装命令执行，如果没有特别说明，均在这台中控机上执行。

将下载的蓝鲸社区版完整包上传到中控机，并解压到 同级 目录下。以解压到 /data 目录为例：

```
tar xf bkce_src-5.1.26.tar.gz -C /data
```

解压之后，得到两个目录：src，install

- src：存放蓝鲸产品软件，以及依赖的开源组件

- install: 存放安装部署脚本、安装时的参数配置、日常运维脚本等

## 8. 检查resolv.conf是否有修改权限

检查 /etc/resolv.conf 是否被加密无法修改(即便是 root)，执行如下命令，检查是否有“i”加密字样：

```
lsattr /etc/resolv.conf
----i-----e-- /etc/resolv.conf
```

如果有则执行命令，临时解密处理，执行如下命令：

```
chattr -i /etc/resolv.conf
```

需要注意，在 resolv.conf 配置文件的首行，即第一个DNS地址需为 127.0.0.1，如下所示：

```
nameserver 127.0.0.1
nameserver 192.168.1.100
nameserver 192.168.2.100
```

“

备注说明：resolv 配置文件无需人工修改内容，后续安装脚本会自动为主机进行配置 127.0.0.1，因此只需检查是否允许修改即可。关于首行需要 127.0.0.1，这是由于后面蓝鲸内部组件的调用所需，域名通过 consul 解析，会探测服务运行状态，然后返回IP地址，例如访问 es，那么内部需要解析 es.service.consul 等，若首行不是 127.0.0.1，否则这些域名就通过外网去解析，无法返回正确的响应，导致服务运行异常，或者 SaaS 无法正常打开等情况。

”

## 配置文件

在 install 目录下，共有三个配置：

- install.config
- globals.env
- ports.env

### install.config

`install.config` 是模块和服务器对应关系的配置文件，描述在哪些机器上安装哪些模块。

每行两列，第一列是 IP 地址；第二列是以英文逗号分隔的模块名称。

详情参考 `install.config.3IP.sample` 文件(可将 `install.config.3IP.sample` 复制为 `install.config`)。

```
[bkce-basic]
10.0.0.1 nginx,rabbitmq,kafka(config),zk(config),es,appt,fta,consul,bkdata(databus)
10.0.0.2 mongodb,appo,kafka(config),zk(config),es,mysql,consul,bkdata(dataapi),beanstalk
10.0.0.3 paas,cmdb,job,gse,license,kafka(config),zk(config),es,redis,influxdb,consul,bkdata(monitor)
```

说明: - 该配置文件, ip 后面使用空格与服务名称隔开, 含有多个内网 ip 的机器, 默认使用 `/sbin/ifconfig` 输出中的第一个内网 ip, 在 ip 后面写上该机器要安装的服务列表即可, 部署过程中默认使用标准私有地址, 若企业环境使用非标准私有地址, 请参考 [本章后续内容-非标准私有地址处理方法](#) 的处理方法。 - zk 表示 ZooKeeper, es 表示 ElasticSearch。 - gse 与 redis 需要部署在同一台机器上。 - 增加机器数量时, 可以将以上配置中的服务挪到新的机器上, 分担负载。要保证: kafka, es, zk 的每个组件的总数量为 3。

## globals.env

该文件定义了各类组件的账号密码信息。功能开关控制选项等。可根据实际情况进行修改。配置项含义, 请查看文件中的注释。

- 该文件含密码信息, 请保证除了 root 用户外, 其他用户不可读。
- 各类账号密码建议修改, 注意设置的各类密码不能有 `/ $ ` < > &` 等特殊字符。
- 配置 HTTP 代理: 若公司不能访问外网, 但有自己的 proxy, 在该配置文件的 `BK_PROXY` 选项中指定代理地址。
- 若需要跨云管理功能(服务器在不同的 IDC, 内网不互通的情况):
- 需要将 gse 所在机器的外网 IP 填到该文件中 `GSE_WAN_IP` 配置项中的括号里 如: `export GSE_WAN_IP=(1.2.3.4)`, 若没有外网 IP 则留空。
- `HAS_DNS_SERVER` 配置默认为 0, 表示配置的蓝鲸域名需要通过 `/etc/hosts` 来解析, 此时部署脚本会自动修改每台机器的 `/etc/hosts` 添加相关域名。如果想走自己的 dns 配置, 改为非 0 即可。
- `HTTP_SCHEMA=http` 默认 `HTTP_SCHEMA` 设置为 HTTP 即蓝鲸软件全站为 HTTP, 若设置为 HTTPS 则蓝鲸软件全站为 HTTPS, 可支持 HTTP 和 HTTPS 的切换。
- 该配置文件中提供了访问蓝鲸三大平台的域名配置, 需要提前准备好。

```
export BK_DOMAIN="bk.com"           # 蓝鲸根域名(不含主机名)
export PAAS_FQDN="paas.$BK_DOMAIN"  # PAAS 完整域名
export CMDDB_FQDN="cmdb.$BK_DOMAIN"  # CMDDB 完整域名
export JOB_FQDN="job.$BK_DOMAIN"     # JOB 完整域名
```

说明: 1. `BK_DOMAIN` 的值不能为 "com" "net" 这种顶级域名, 至少二级域名开始。 2.

“ FQDN 的选择需要遵循 DNS 的命名规范，可选的字符集是 [A-Za-z0-9.] 以及 "-", 特别要注意，下划线 ( \_ ) 是不允许的。3. PAAS\_FQDN、CMDB\_FQDN、JOB\_FQDN 的值都必须在 BK\_DOMAIN 定义的根域名之下，保证登录鉴权的 cookie 文件有效。

”

## ports.env

端口定义。默认情况下，不用修改。特殊场景下，若有端口冲突，可以自行定义。

## 非标准私有地址处理方法

蓝鲸社区版部署脚本中(install 目录)下有以下文件中有获取 ip 的函数 get\_lan\_ip，非标准地址，均需要在安装部署前完成修改。

```
./appmgr/docker/saas/buildsaas
./appmgr/docker/build
./functions
./scripts/gse/server/gsectl
./scripts/gse/plugins/reload.sh
./scripts/gse/plugins/start.sh
./scripts/gse/plugins/stop.sh
./scripts/gse/agent/gsectl
./scripts/gse/proxy/gsectl
./scripts/gse/agentaix/gsectl.ksh
./agent_setup/download#agent_setup_pro.sh
./agent_setup/download#agent_setup_aix.ksh
./agent_setup/download#agent_setup.sh
```

这些文件列表，可能随版本迭代变动，也可以用以下命令查找出来包含这个函数的脚本文件有哪些：

```
grep -l 'get_lan_ip *()' -r /data/install
```

修改方法：

假设服务器的 ip 是：138.x.x.x，它不在标准的私有地址范围，那么你需要修改 get\_lan\_ip() 函数为：

```
get_lan_ip () {
...省略
    if ($3 ~ /^10\./) {
        print $3
    }
    if ($3 ~ /^138\./) {
        print $3
    }
}

return $?
}
```

## pip.conf

在线安装时，依赖 pip，需要配置可用的 pip 源。

```
vim /data/src/.pip/pip.conf

[global]
index-url = https://mirrors.cloud.tencent.com/pypi/simple
trusted-host = mirrors.cloud.tencent.com
```

- 设置为能连上的 pip 源，以腾讯云镜像源加速 pip 为例。默认的 pip 源配置通常无法使用，验证方式如下。
- 在每台机器上对 `pip.conf` 中配置的 url 进行操作：`curl http://xxxxxxx`，若能正常返回列表信息则为成功。

## 获取证书

1. 通过 `ifconfig` 或者 `ip addr` 命令获取 `install.config` 文件中，`license` 和 `gse` 模块所在服务器的第一个内网网卡的 MAC 地址。如果分别属于两台服务器，那么两个的 MAC 地址以英文;分隔。
2. 在官网 [证书生成页面](#) 根据输入框提示填入 MAC 地址，生成并下载证书。
3. 上传证书到中控机，并解压到 `src/cert` 目录下。

```
tar xf ssl_certificates.tar.gz -C /data/src/cert/
```

## 配置 SSH 免密登陆

登录到中控机，执行以下操作：

```
cd /data/install
bash configure_ssh_without_pass # 根据提示输入各主机的 root 密码完成免密登陆配置
```

## 安装前校验环境是否满足

按文档要求做完环境和部署的配置后，准备开始安装前，请运行以下脚本，来校验是否满足：

```
cd /data/install
bash precheck.sh
```

正常输出如下图所示：

```
start <<check_ssh_nopass>> ... [OK]
```

```
start <<check_password>> ... [OK]
start <<check_cert_mac>> ... [OK]
start <<check_get_lan_ip>> ... [OK]
start <<check_install_config>> ... [OK]
start <<check_selinux>> ... [OK]
start <<check_umask>> ... [OK]
start <<check_rabbitmq_version>> ... [OK]
start <<check_http_proxy>> ... [OK]
start <<check_open_files_limit>> ... [OK]
start <<check_domain>> ... [OK]
start <<check_rsync>> ... [OK]
start <<check_service_dir>> ... [OK]
start <<check_networkmanager>> ... [OK]
start <<check_firewalld>> ... [OK]
```

如果发现有 [FAIL] 的报错，按照错误提示和本文档修复。修复后，可继续跑 precheck.sh 脚本，直到不再出现 [FAIL]。

如果需要从头开始检查，请使用 `precheck.sh -r` 参数。

## 单机部署

单机部署方案是针对新老用户快速低成本的搭建一套精简版的蓝鲸社区版推出的。

从社区版 4.1.11 版本开始，支持完整版本蓝鲸的单机部署体验，前提要求是，单机 CPU 不低于 2 核，可用内存不低于 16GB。若机器性能没满足要求，只能部署精简版的蓝鲸平台。

精简版蓝鲸平台满足简单的运维场景需求，包含：PaaS 平台，配置平台，作业平台，以及安装 Agent 用的 节点管理 SaaS。

### 部署方式

- 腾讯云实验室
- 自行提供主机

### 自行提供主机

按照安装 [环境准备](#) 章节中，主机和系统环境的要求做好相应设置。

- 环境准备
- 准备一台 CentOS 7 以上操作系统的机器(物理机和虚拟机均可)。
- 按照安装 [环境准备](#) 章节中，主机和系统环境的要求做好相应设置。
- 配置好 YUM 源，包含 EPEL 仓库(可以通过 `yum info nginx` 测试下)。
- 从 [官网下载](#) 完整包，并解压到 /data/ 下。



```
bash $ tar xf bkce_src-5.0.3.tar.gz -C /data
```

- 获取机器的 MAC 地址后，下载 [证书文件](#)，解压到 src/cert 目录下

```
bash $ tar xf ssl_certificates.tar.gz -C /data/src/cert
```

- 配置参数
- install.config 这个文件安装脚本会自动生成，无需自行配置。
- globals.env 重点关注域名和 GSE 外网 IP 相关的配置，详情参考 [环境准备-配置文件](#) 一节。
- ports.env 一般不用修改。
- 执行安装

如果部署全部组件，请执行：

```
bash $ cd /data/install $ ./install_minibk -y
```

如果按需部署，假设只需要 PaaS，CMDB，JOB 平台，请执行：

```
bash $ cd /data/install $ ./install_minibk $ ./bk_install paas && ./bk_install  
cmdb && ./bk_install job
```

安装过程中遇到失败的情况，请先定位排查解决后，再重新运行失败时的安装指令。

## 访问蓝鲸

根据 [install/globals.env](#) 里配置的 PaaS 域名(PAAS\_FQDN)、账号 (PAAS\_ADMIN\_USER)、密码(PAAS\_ADMIN\_PASS)信息，登录访问(若域名没设置 DNS 解析，需配置本机 hosts)。

- 域名信息

```
bash export BK_DOMAIN="xxx.com" export PAAS_FQDN="paas.$BK_DOMAIN" export  
CMDB_FQDN="cmdb.$BK_DOMAIN" export JOB_FQDN="job.$BK_DOMAIN"
```

- 账号信息

```
bash export PAAS_ADMIN_USER=admin export PAAS_ADMIN_PASS="xxx"
```

日常维护和运维，单机部署和多机是一致的，请参考 [维护文档](#)。

## 使用蓝鲸

请参考蓝鲸各 [产品白皮书](#)。

# 环境验证

## 从后台验证环境是否安装正常

1. 登录到中控机，通过脚本自带的命令查询蓝鲸后台模块的运行状态

```
# 加载环境变量和蓝鲸安装维护的函数
source /data/install/utils.fc
```

```
# 查看运行状态
./bkcec status all
```

```
nginx: RUNNING
iam-nginx: RUNNING
server    iam-permsvr      RUNNING    pid 9068, uptime 10 days, 0:08:06
server    iam-subserver    RUNNING    pid 9074, uptime 10 days, 0:08:06
zk: RUNNING
rabbitmq: RUNNING
paas_agent()    paas_agent      RUNNING    pid 11149, uptime 10 days, 0:07:58
nginx: RUNNING
es: RUNNING
kafka: RUNNING
databus    databus_es      RUNNING    pid 6183, uptime 10 days, 0:08:39
databus    databus_etl      RUNNING    pid 6185, uptime 10 days, 0:08:39
databus    databus_jdbc     RUNNING    pid 6182, uptime 10 days, 0:08:39
databus    databus_redis    RUNNING    pid 6186, uptime 10 days, 0:08:39
databus    databus_tsdb     RUNNING    pid 6184, uptime 10 days, 0:08:39
fta        common:apisvr    RUNNING    pid 30939, uptime 0:00:13
fta        common:jobserver RUNNING    pid 30937, uptime 0:00:13
fta        common:logging   RUNNING    pid 30941, uptime 0:00:13
fta        common:polling0  RUNNING    pid 30938, uptime 0:00:13
fta        common:qos       RUNNING    pid 30931, uptime 0:00:13
fta        common:scheduler0 RUNNING    pid 30934, uptime 0:00:13
fta        common:webserver RUNNING    pid 30940, uptime 0:00:13
fta        fta:collect0     RUNNING    pid 30926, uptime 0:00:13
fta        fta:converge0    RUNNING    pid 30922, uptime 0:00:13
fta        fta:job          RUNNING    pid 30924, uptime 0:00:13
fta        fta:match_alarm0 RUNNING    pid 30929, uptime 0:00:13
fta        fta:poll_alarm   RUNNING    pid 30920, uptime 0:00:13
fta        fta:solution     RUNNING    pid 30923, uptime 0:00:13
consul-agent    RUNNING    pid 11479, uptime 27 days, 8:08:13
mysql: RUNNING
mongodb: RUNNING
zk: RUNNING
paas_agent()    paas_agent      RUNNING    pid 507, uptime 10 days, 0:08:00
nginx: RUNNING
es: RUNNING
kafka: RUNNING
beanstalk: RUNNING
dataapi    dataapi          RUNNING    pid 28940, uptime 10 days, 0:08:40
dataapi    dataapi-celery-1 RUNNING    pid 28939, uptime 10 days, 0:08:40
dataapi    update_cc_cache  RUNNING    pid 442, uptime 10 days, 0:08:02
consul-agent    RUNNING    pid 6549, uptime 27 days, 8:08:11
license: RUNNING
redis: RUNNING
open_paas    paas            RUNNING    pid 20560, uptime 10 days, 0:08:25
open_paas    appengine       RUNNING    pid 20563, uptime 10 days, 0:08:25
open_paas    esb             RUNNING    pid 20562, uptime 10 days, 0:08:25
open_paas    login           RUNNING    pid 20561, uptime 10 days, 0:08:25
open_paas    apigw           RUNNING    pid 20564, uptime 10 days, 0:08:25
gse_alarm: RUNNING
gse_ops: RUNNING
gse_opts: RUNNING
gse_api: RUNNING
gse_btsvr: RUNNING
gse_data: RUNNING
gse_dba: RUNNING
gse_task: RUNNING
gse_syncdata: RUNNING
gse_procmgr: RUNNING
```

```
gse task: RUNNING
gse_syncdata: RUNNING
gse_procmgr: RUNNING
gse_dataop: RUNNING
cmdb-nginx: RUNNING
server      cmdb_adminserver      RUNNING      pid 4941, uptime 27 days, 7:58:42
server      cmdb_apiserver         RUNNING      pid 4938, uptime 27 days, 7:58:42
server      cmdb_auditcontroller   RUNNING      pid 4937, uptime 27 days, 7:58:42
server      cmdb_datacollection    RUNNING      pid 4940, uptime 27 days, 7:58:42
server      cmdb_eventserver       RUNNING      pid 4939, uptime 27 days, 7:58:42
server      cmdb_hostcontroller    RUNNING      pid 4931, uptime 27 days, 7:58:42
server      cmdb_hostserver        RUNNING      pid 4932, uptime 27 days, 7:58:42
server      cmdb_objectcontroller  RUNNING      pid 4934, uptime 27 days, 7:58:42
server      cmdb_proccontroller    RUNNING      pid 4942, uptime 27 days, 7:58:42
server      cmdb_procserver        RUNNING      pid 4936, uptime 27 days, 7:58:42
server      cmdb_toposerver        RUNNING      pid 4933, uptime 27 days, 7:58:42
server      cmdb_webserver         RUNNING      pid 4935, uptime 27 days, 7:58:42
zk: RUNNING
job: RUNNING
es: RUNNING
kafka: RUNNING
influxdb: RUNNING
monitor      alarm:alarm0           RUNNING      pid 19324, uptime 10 days, 0:08:43
monitor      alarm:alarm1           RUNNING      pid 19323, uptime 10 days, 0:08:43
monitor      common:logging         RUNNING      pid 19352, uptime 10 days, 0:08:43
monitor      converge:converge0     RUNNING      pid 19345, uptime 10 days, 0:08:43
monitor      converge:converge1     RUNNING      pid 19346, uptime 10 days, 0:08:43
monitor      converge:converge2     RUNNING      pid 19347, uptime 10 days, 0:08:43
monitor      converge:converge3     RUNNING      pid 19348, uptime 10 days, 0:08:43
monitor      converge:converge4     RUNNING      pid 19344, uptime 10 days, 0:08:43
monitor      healthz:healthz0      RUNNING      pid 18776, uptime 0:03:34
monitor      kernel:match_alarm0    RUNNING      pid 19342, uptime 10 days, 0:08:43
monitor      kernel:match_alarm1    RUNNING      pid 19341, uptime 10 days, 0:08:43
monitor      kernel:match_alarm2    RUNNING      pid 19340, uptime 10 days, 0:08:43
monitor      kernel:match_alarm3    RUNNING      pid 19339, uptime 10 days, 0:08:43
monitor      kernel:qos             RUNNING      pid 19343, uptime 10 days, 0:08:43
monitor      kernel_api:kernel_api0 RUNNING      pid 19327, uptime 10 days, 0:08:43
monitor      nodata:nodata0         RUNNING      pid 19337, uptime 10 days, 0:08:43
monitor      nodata:nodata1         RUNNING      pid 19338, uptime 10 days, 0:08:43
monitor      recovery:recovery0     RUNNING      pid 19325, uptime 10 days, 0:08:43
monitor      recovery:recovery1     RUNNING      pid 19326, uptime 10 days, 0:08:43
monitor      run_cron               RUNNING      pid 19322, uptime 10 days, 0:08:43
monitor      run_data_access:run_data_access0 RUNNING      pid 11740, uptime 0:44:47
monitor      run_data_access:run_data_access1 RUNNING      pid 11538, uptime 0:44:52
monitor      run_data_access:run_data_access2 RUNNING      pid 7078, uptime 0:45:25
monitor      run_data_access:run_data_access3 RUNNING      pid 10843, uptime 0:44:58
monitor      run_data_detect:run_data_detect0 RUNNING      pid 12734, uptime 0:44:36
monitor      run_data_detect:run_data_detect1 RUNNING      pid 12561, uptime 0:44:39
monitor      run_data_detect:run_data_detect2 RUNNING      pid 12604, uptime 0:44:38
monitor      run_data_detect:run_data_detect3 RUNNING      pid 8246, uptime 0:45:14
monitor      run_poll_alarm:run_poll_alarm_bk_monitor RUNNING      pid 24000, uptime 0:02:45
monitor      run_poll_alarm:run_poll_alarm_gse_base_alarm RUNNING      pid 22625, uptime 0:08:05
monitor      run_poll_alarm:run_poll_alarm_gse_custom_str RUNNING      pid 10269, uptime 0:09:01
monitor      scheduler:scheduler0   RUNNING      pid 19356, uptime 10 days, 0:08:43
monitor      scheduler:scheduler1   RUNNING      pid 19355, uptime 10 days, 0:08:43
monitor      scheduler:scheduler2   RUNNING      pid 19354, uptime 10 days, 0:08:43
monitor      scheduler:scheduler3   RUNNING      pid 19353, uptime 10 days, 0:08:43
root@bkdata:~#
```

如上图所示，所有模块状态都是 **RUNNING** 则说明蓝鲸社区版后台模块部署正常。

## 2. 检查 BKDATA 清洗数据的任务是否正常启动

- 社区版 V5.0 及更早版本，在部署 BKDATA 的主机上执行：

```
bash # 检查是否有包含 databus_etl 的任务 crontab -l
```

- 社区版 V5.1 及之后版本，在部署完 BKDATA 后，检查进程 **run\_cron** 是否存活：

## 3. 检查 BKDATA 运行状态

``bash # 从中控机中跳转到 BKDATA Svr，执行检查脚本检查接口，无报错即正常

```
ssh $BKDATA_IP /data/bkce/bkdata/dataapi/bin/check_databus_status.sh ``
```

```

root@btmodel bin# ./check_databus_stat.sh
--ES--
% Total      % Received % Xferd  Average Speed   Time    Time     Time  Current
   0      0    0     0      0      0      0      0      0      0
100    100    100    15.5M    0      0      0      0      0      16

--TSDB--
% Total      % Received % Xferd  Average Speed   Time    Time     Time  Current
   0      0    0     0      0      0      0      0      0      0
100    100    100    89.1M    0      0      0      0      0    89777

tsdb_2_apache_net
tsdb_2_apache_net,connector":{"state":"RUNNING","worker_id":10},tasks":{"state":"RUNNING","id":0,"worker_id":10.0,"3:10054"}}
tsdb_2_tomcat_jsp
tsdb_2_tomcat_jsp,connector":{"state":"RUNNING","worker_id":10},tasks":{"state":"RUNNING","id":0,"worker_id":10.0,"4"}}
tsdb_2_mysql_innodb
tsdb_2_mysql_innodb,connector":{"state":"RUNNING","worker_id":1},tasks":{"state":"RUNNING","id":0,"worker_id":1,"3:10054"}}
tsdb_2_mysql_net
tsdb_2_mysql_net,connector":{"state":"RUNNING","worker_id":10},tasks":{"state":"RUNNING","id":0,"worker_id":10.0,"4"}}
tsdb_2_mysql_rep
tsdb_2_mysql_rep,connector":{"state":"RUNNING","worker_id":10},tasks":{"state":"RUNNING","id":0,"worker_id":10.0,"4"}}
tsdb_2_uptimecheck_udp
tsdb_2_uptimecheck_udp,connector":{"state":"RUNNING","worker_id":1},tasks":{"state":"RUNNING","id":0,"worker_id":10.0,"3:10054"}}
tsdb_2_tomcat_servlet
tsdb_2_tomcat_servlet,connector":{"state":"RUNNING","worker_id":1},tasks":{"state":"RUNNING","id":0,"worker_id":10.0,"4"}}
tsdb_2_nginx_net
tsdb_2_nginx_net,connector":{"state":"RUNNING","worker_id":1},tasks":{"state":"RUNNING","id":0,"worker_id":10.0,"4"}}
tsdb_2_redis_stat
tsdb_2_redis_stat,connector":{"state":"RUNNING","worker_id":1},tasks":{"state":"RUNNING","id":0,"worker_id":10.0,"4"}}
tsdb_2_system_mem
tsdb_2_system_mem,connector":{"state":"RUNNING","worker_id":1},tasks":{"state":"RUNNING","id":0,"worker_id":10.0,"4"}}
tsdb_2_system_cpu_detail
tsdb_2_system_cpu_detail,connector":{"state":"RUNNING","worker_id":1},tasks":{"state":"RUNNING","id":0,"worker_id":10.0,"3:10054"}}
tsdb_2_system_load
tsdb_2_system_load,connector":{"state":"RUNNING","worker_id":1},tasks":{"state":"RUNNING","id":0,"worker_id":10.0,"4"}}
tsdb_2_apache_performance
tsdb_2_apache_performance,connector":{"state":"RUNNING","worker_id":1},tasks":{"state":"RUNNING","id":0,"worker_id":10.0,"3:10054"}}
tsdb_2_system_net
tsdb_2_system_net,connector":{"state":"RUNNING","worker_id":1},tasks":{"state":"RUNNING","id":0,"worker_id":10.0,"3:10054"}}
tsdb_2_redis_aof
tsdb_2_redis_aof,connector":{"state":"RUNNING","worker_id":1},tasks":{"state":"RUNNING","id":0,"worker_id":10.0,"4"}}
tsdb_2_tomcat_cache
tsdb_2_tomcat_cache,connector":{"state":"RUNNING","worker_id":1},tasks":{"state":"RUNNING","id":0,"worker_id":10.0,"4"}}
tsdb_2_system_proc
tsdb_2_system_proc,connector":{"state":"RUNNING","worker_id":1},tasks":{"state":"RUNNING","id":0,"worker_id":10.0,"4"}}
tsdb_2_redis_rdb
tsdb_2_redis_rdb,connector":{"state":"RUNNING","worker_id":1},tasks":{"state":"RUNNING","id":0,"worker_id":10.0,"4"}}
tsdb_2_system_swap
tsdb_2_system_swap,connector":{"state":"RUNNING","worker_id":1},tasks":{"state":"RUNNING","id":0,"worker_id":10.0,"4"}}
tsdb_2_tomcat_net
tsdb_2_tomcat_net,connector":{"state":"RUNNING","worker_id":1},tasks":{"state":"RUNNING","id":0,"worker_id":10.0,"4"}}
tsdb_2_uptimecheck_http
tsdb_2_uptimecheck_http,connector":{"state":"RUNNING","worker_id":1},tasks":{"state":"RUNNING","id":0,"worker_id":10.0,"3:10054"}}
tsdb_2_system_inode
tsdb_2_system_inode,connector":{"state":"RUNNING","worker_id":1},tasks":{"state":"RUNNING","id":0,"worker_id":10.0,"3:10054"}}
tsdb_2_uptimecheck_tcp
tsdb_2_uptimecheck_tcp,connector":{"state":"RUNNING","worker_id":1},tasks":{"state":"RUNNING","id":0,"worker_id":10.0,"3:10054"}}

```

#### 4. 检查 Agent 状态是否正常:

- 在中控机执行

```
``bash ./bkcec status gse_agent
```

# 返回 "Running" 说明 Agent 状态正常 # 如果返回 "Exit" 则需要手动安装或者启动 # 手动安装

```
./bkcec install gse_agent # 启动 ./bkcec start gse_agent ``
```

## 5. 检查健康状况 API:

- 蓝鲸产品后台提供了健康检查的接口，用 HTTP GET 请求访问，接口地址和端口用变量表达：

```
``bash cd /data/install && source utils.fc
```

```
# PaaS 注意 URL 末尾带上/ curl http://$PAAS_FQDN:$PAAS_HTTP_PORT/healthz/
```

```
# CMDB curl http://$CMDB_IP:$CMDB_API_PORT/healthz
```

```
# JOB curl http://$JOB_FQDN:$PAAS_HTTP_PORT/healthz ``
```

- 蓝鲸监控 SaaS 的监控检查接口，可以用浏览器直接访问：

```
http://$PAAS_FQDN:$PAAS_HTTP_PORT/o/bk_monitor/healthz/
```

## 从页面验证功能是否运行正常

请先 **配置 host 或者 DNS** 解析后，确认访问社区版域名(部署完成后提示的域名)是否正常。

- 配置 host / DNS 解析：配置 DNS 服务或者在本地配置 host 解析请参照 蓝鲸配置文件 global.env 或者中控机下 `/etc/hosts`。

从蓝鲸工作台-开发者中心-服务器中检查 **正式服务器** 是否激活。

个人信息

应用中心

应用创建

我的应用

S-mart应用

服务器信息

第三方服务

服务器信息

+ 添加服务器信息

服务器ID	Token	IP	Agent端口	App服务端口	类别	启用	操作
			4245	8010	测试服务器	是	<div><div></div><div></div></div>
			4245	8010	正式服务器	是	<div><div></div><div></div></div>

从蓝鲸工作台-开发者中心-第三方服务中检查 **RabbitMQ 服务** 是否激活。

个人信息

应用中心

我的应用

S-mart应用

外链应用

服务器信息

第三方服务

服务介绍

新手指南

资源下载

第三方服务信息

+ 添加第三方服务

地址(IP或域名)	端口	用户名	密码	类别	启用	操作
rabbitmq.service.consul			*****	RabbitMQ服务	是	<div><div></div><div></div></div>

使用说明

RabbitMQ服务是用来做应用后台任务（ celery 任务）的消息队列

RabbitMQ服务可部署在任意一台服务器上，但需保证 15672 和 5672 端口未被占用

如果搭建RabbitMQ服务集群，需确保集群所有节点配置的用户名/密码/端口一致

端口默认 15672（一般不需要更改）

只能激活一个RabbitMQ服务集群

登录蓝鲸，从各 SaaS 验证环境功能是否运行正常。

“

产品功能介绍&使用手册请参考各 [产品白皮书](#)

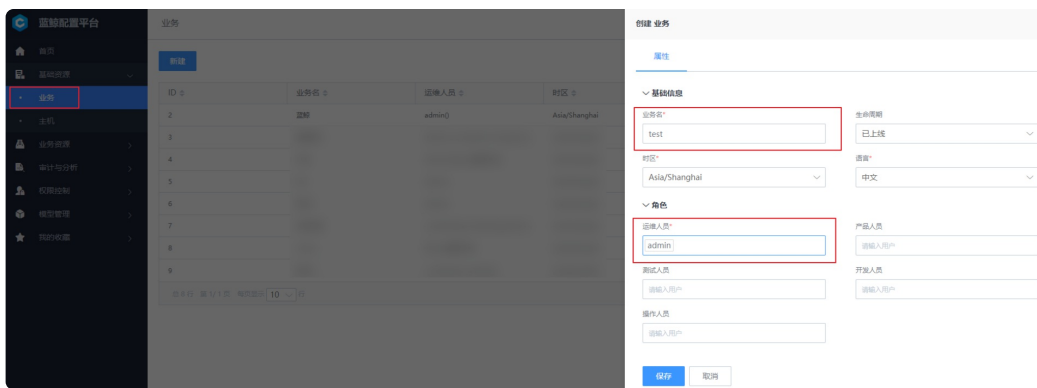
”

## 快速使用

为了验证刚部署安装的环境是否正常，除了从后台验证，也需要从页面上进行功能性的验证。

## 创建业务

- 首先在配置平台(CMDB)上新建一个业务，将 admin 添加至运维人员。



- 在业务下新建集群以及模块。(蓝鲸业务不支持集群及模块操作)

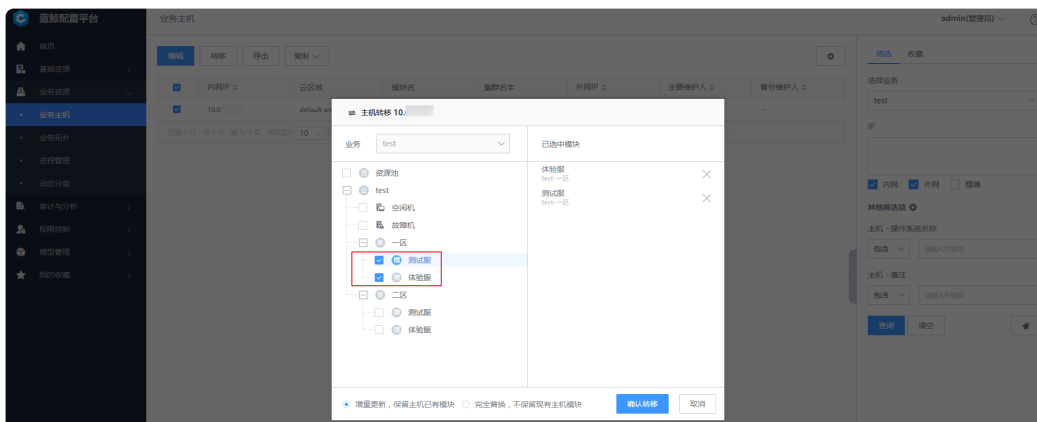


下述步骤需在节点管理完成后操作。

- 成功安装完 Agent 后，在 配置平台(CMDB) 中的对应业务下，则会出现安装好 Agent 的主机。

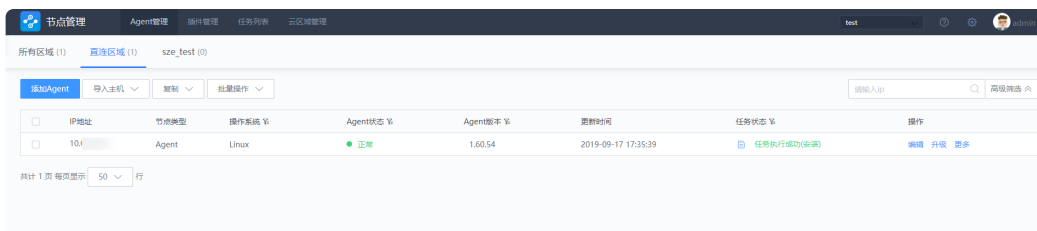


- 将资源池下的主机移动到模块下。

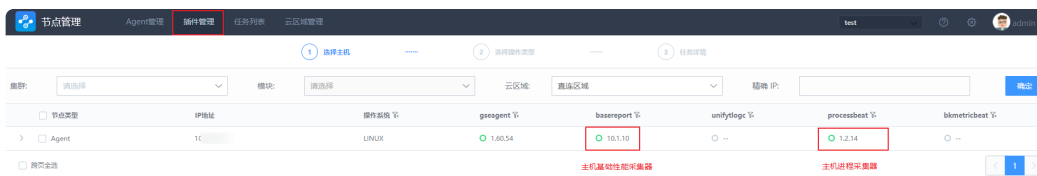


## 安装 Agent

- 在对应业务下安装主机 Agent。

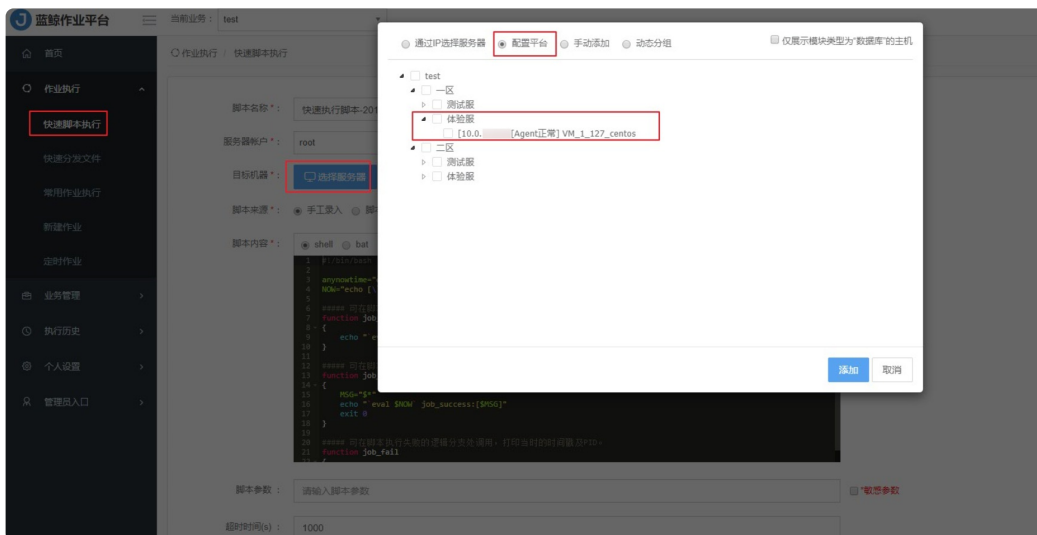


- Agent 安装成功后查看下采集器是否正常。

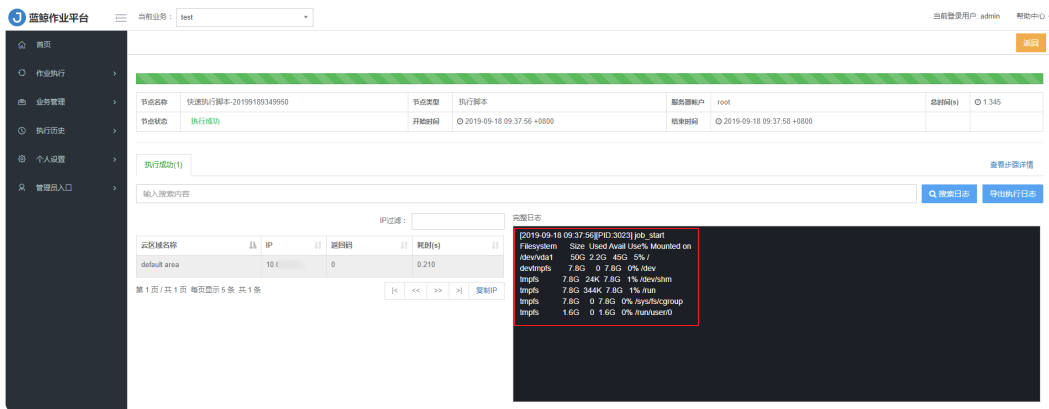


## 执行作业

- 测试快速执行脚本可用性。

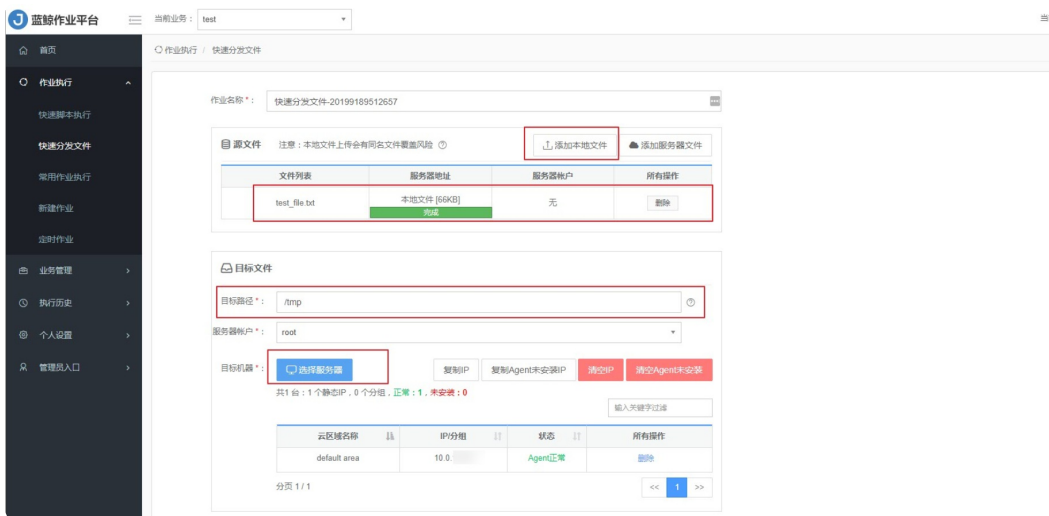


- 查看作业执行情况。

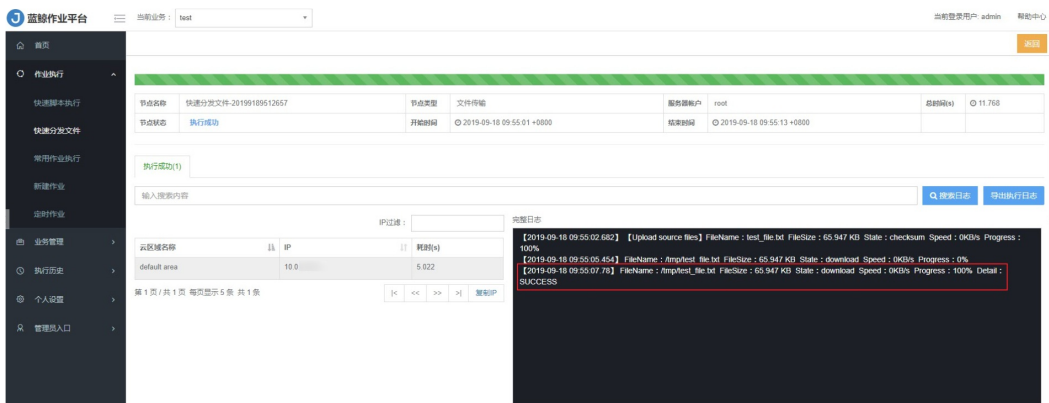


- 测试快速分发文件可用性。

上传测试文件，分发至目标主机。

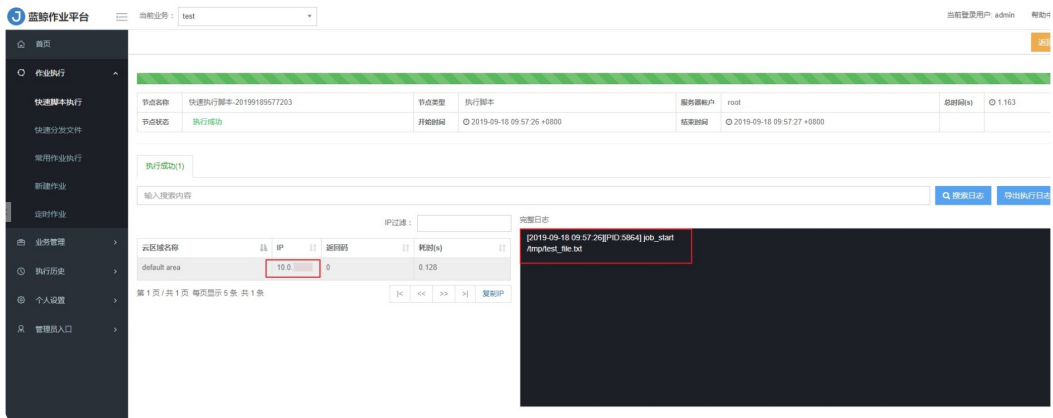


查看文件分发情况。



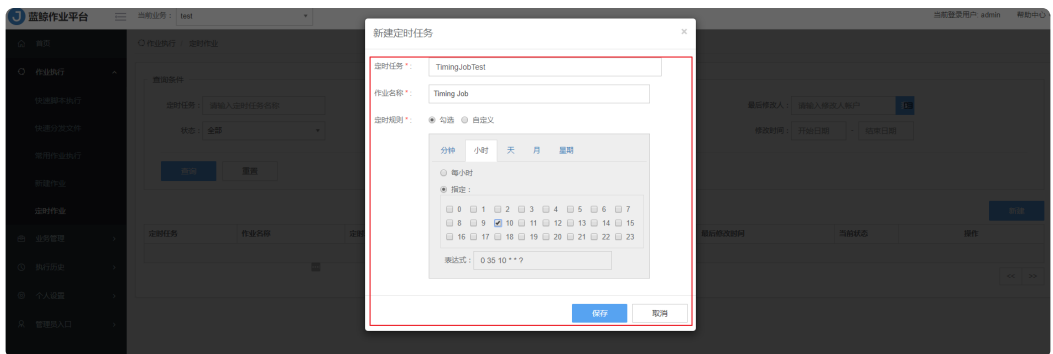
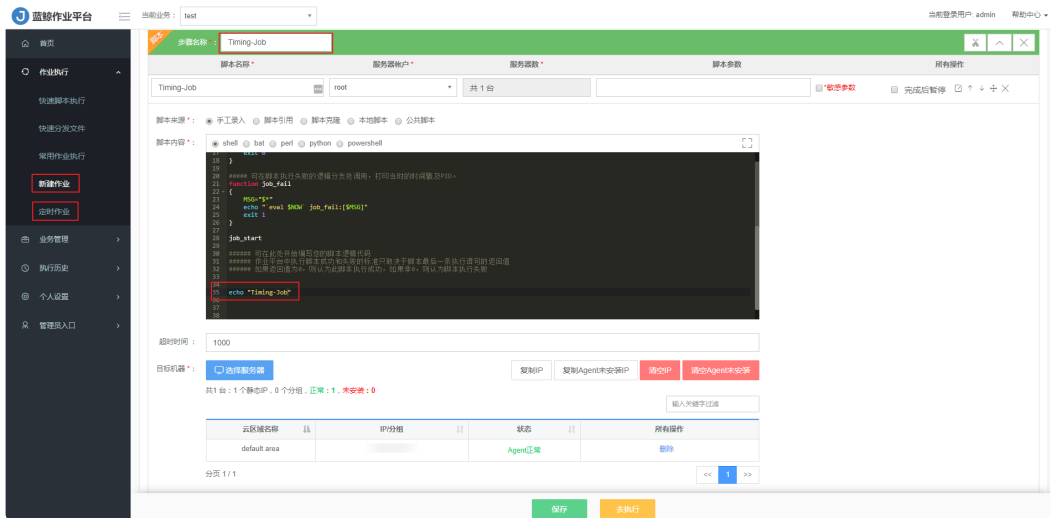
查看目标主机路径下是否存在该文件。



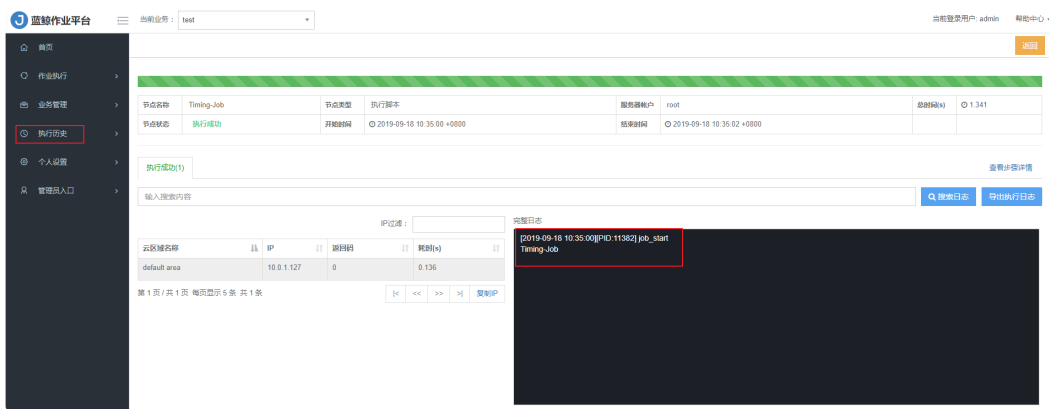


- 测试定时作业可用性。

左侧导航栏选择【新建作业】，作业新建完成后，选择【定时作业】进行添加定时作业任务。然后启动任务。



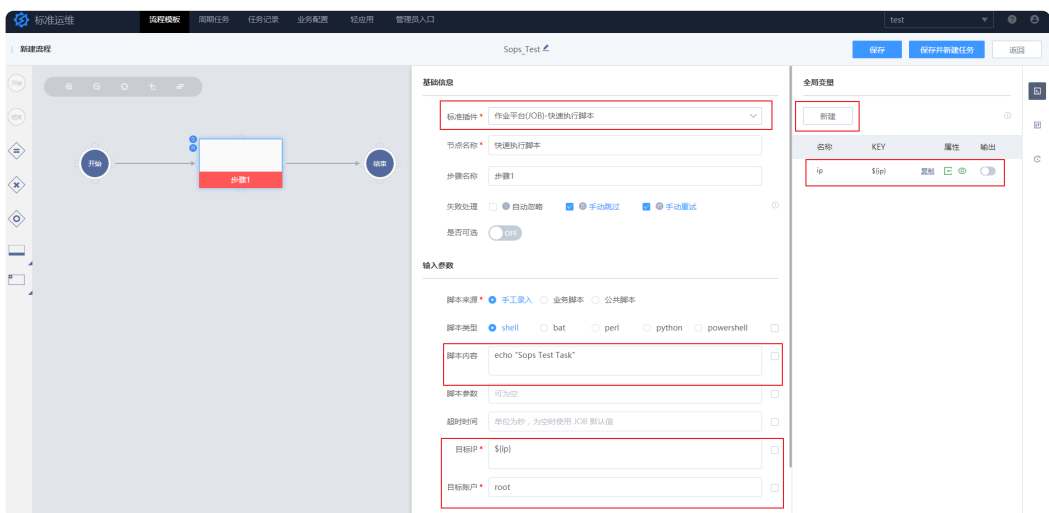
左侧导航栏选择【执行历史】查看定时作业执行情况。



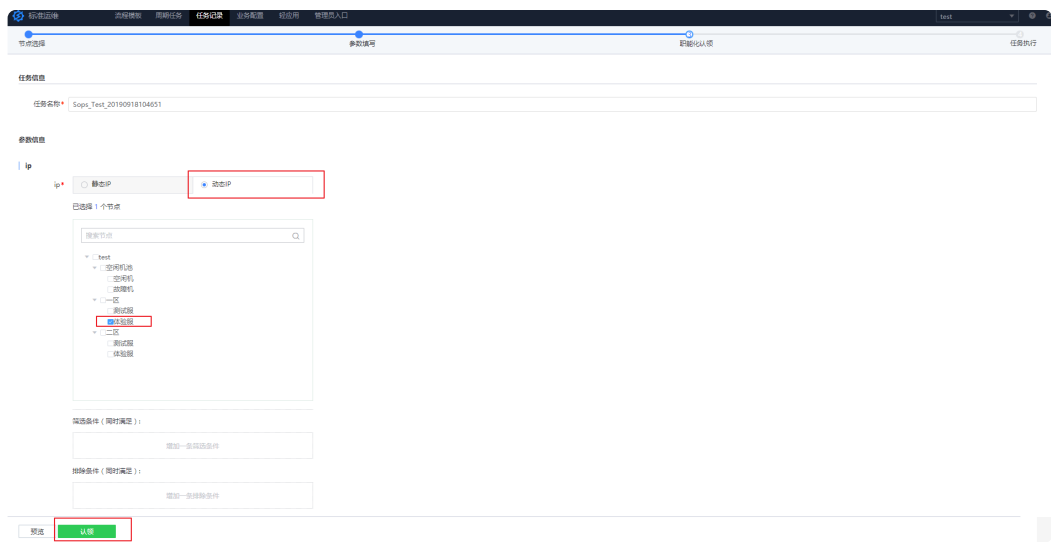
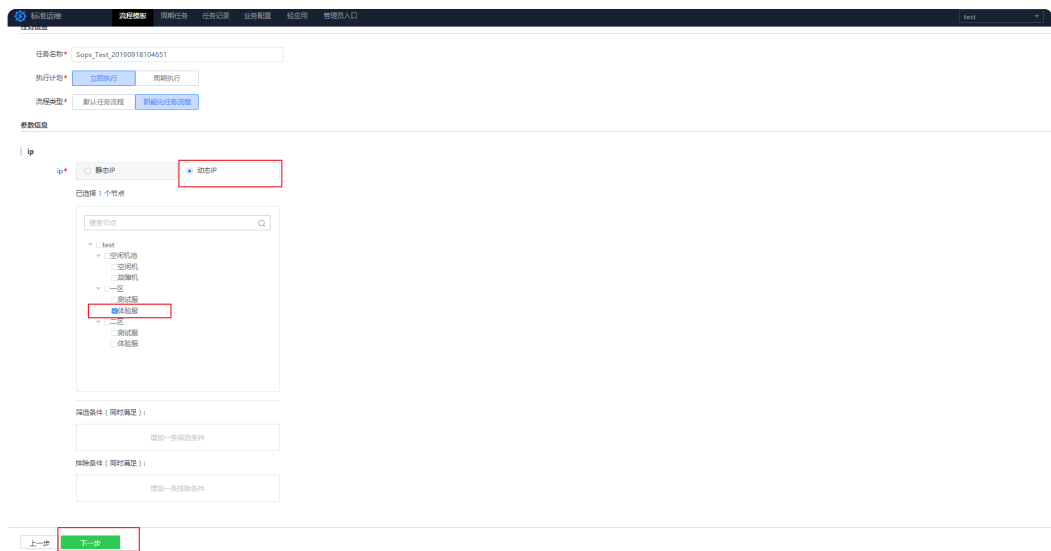
## 创建流程

- 新建一个业务流程模板。

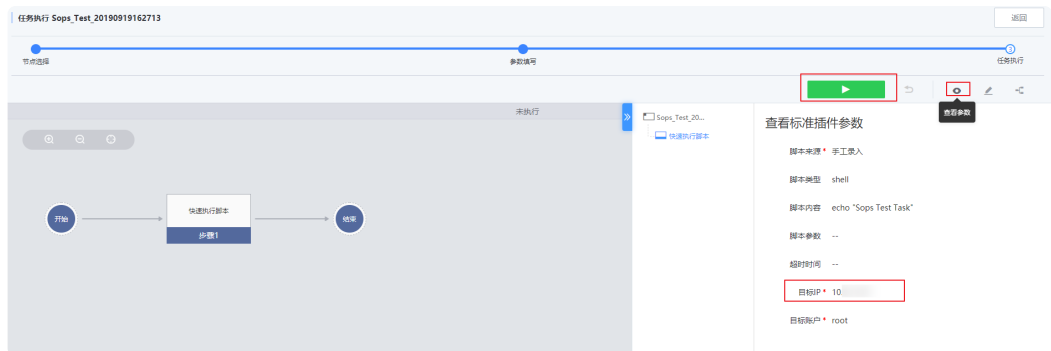
选择【快速执行脚本】，输入脚本内容，目标 IP 通过变量进行传入，类型选择【IP 选择器】，IP 选择器(简单版)跟 IP 选择器的区别。



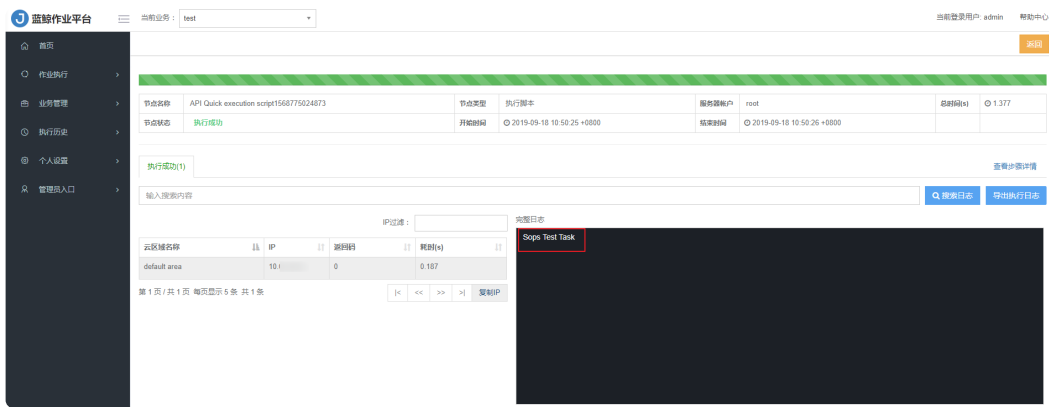
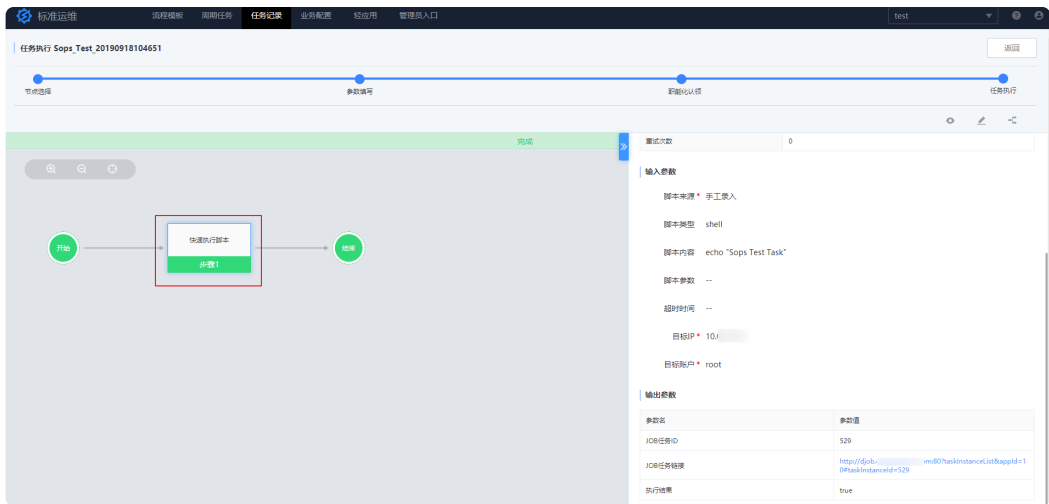
- 新建并认领流程任务。



- 查看参数无误并执行任务。



- 任务执行完成后，点击步骤可以查看执行的详细信息。点击对应链接可以跳转至作业平台查看作业执行情况及信息。



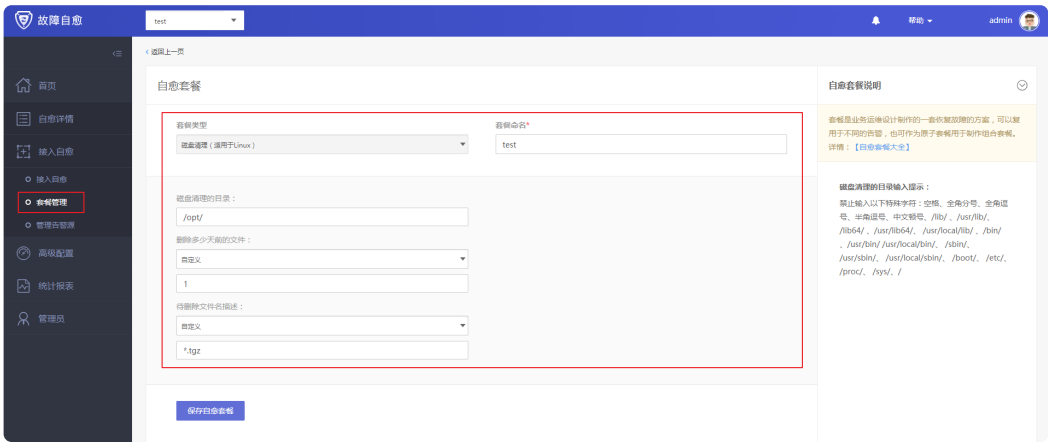
## 主机监控

- 查看蓝鲸获取主机信息，主机状态，主机监控，服务状态图情况。

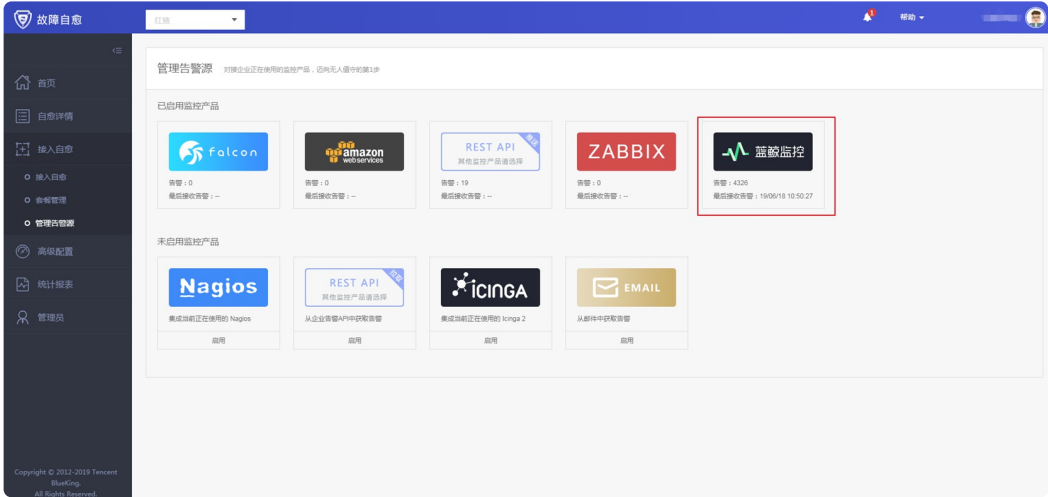
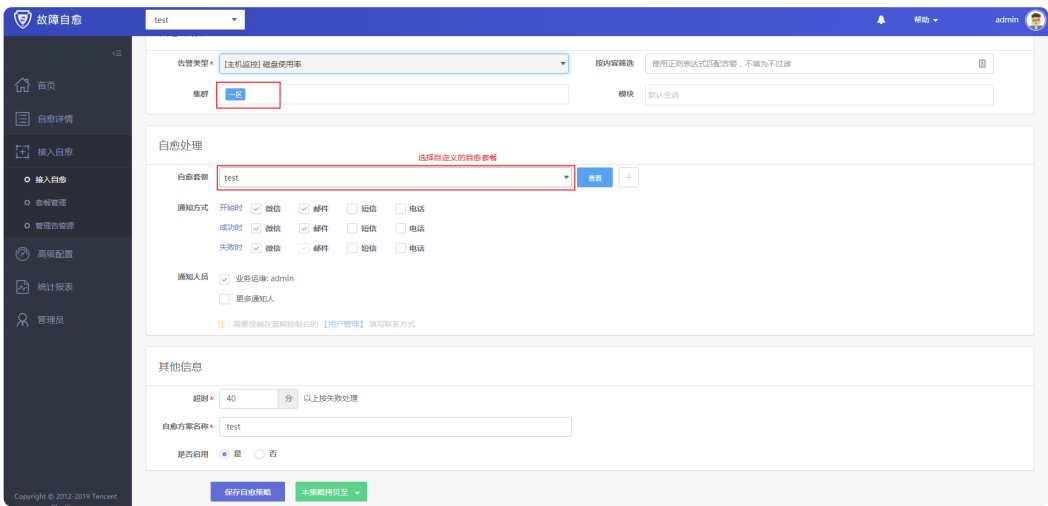


## 故障自愈

- 创建自愈套餐(如果不创建也可以自行使用默认套餐)。



- 接入自愈，蓝鲸监控模拟告警，看故障自愈是否能够获取到告警并实现自愈。



更多信息

说明：“快速入门”是蓝鲸软件的一个基础操作，并不能涵盖所有的内容，通过下面文章介绍，可以获取更多信息。

- 部署维护：蓝鲸支持多机部署，详见[部署维护](#)的内容。
- 场景案例：为了方便用户获取蓝鲸软件的最佳实践，提供了覆盖CI、CD、CO领域的[实践案例](#)。
- 开发指南：软件支持私有化部署，提供的是通用的服务，若需要进行定制化服务，提供了[二次开发](#)的教程。
- 常见问题：提供了丰富的[问题库](#)。