**云运维平台V3.5(专属云)**



2018年1月

# 目录

[目录 2](#_Toc505087277)

[1 概述 3](#_Toc505087278)

[2 本版修订 4](#_Toc505087279)

[3 产品特性 6](#_Toc505087280)

[4 产品范围 7](#_Toc505087281)

[4.1 DevOps平台 7](#_Toc505087282)

[4.1.1 持续发布 7](#_Toc505087283)

[4.1.2 运维白屏化 9](#_Toc505087284)

[4.2 容器云平台 11](#_Toc505087285)

[5 产品主要功能 14](#_Toc505087286)

[5.1 DevOps平台 14](#_Toc505087287)

[5.1.1 持续交付 14](#_Toc505087288)

[5.1.2 运维白屏化 16](#_Toc505087289)

[5.2 容器云平台 18](#_Toc505087290)

[5.2.1 资源池管理 18](#_Toc505087291)

[5.2.2 容器服务 18](#_Toc505087292)

[5.2.3 容器调度 18](#_Toc505087293)

[5.2.4 统一接入 18](#_Toc505087294)

[5.2.5 服务网关 19](#_Toc505087295)

[5.2.6 弹性伸缩 19](#_Toc505087296)

[5.2.7 服务自愈 19](#_Toc505087297)

[5.2.8 服务健康检查 19](#_Toc505087298)

[5.2.9 环境变量管理 20](#_Toc505087299)

[5.2.10 持久存储管理 20](#_Toc505087300)

[5.2.11 容器网络管理 20](#_Toc505087301)

[5.2.12 中间件服务 20](#_Toc505087302)

# 概述

iuap云运维平台V3.5版本是用友云开发者中心的企业版，以下简称开发者中心。

用友云开发者中心旨在为开发者提供了资源管理、容器服务、持续集成、持续交付、镜像仓库等应用基础服务，同时为应用的微服务架构落地提供完备的支撑，结合DevOps的理念，通过提供自动化运维、日志管理、运营分析、中间件服务等功能，帮助开发及运维人员降低产品研发迭代过程中的负担。

用友云开发者中心包括：

1. DevOps平台：以可视化的方式实现应用的集成、测试、部署自动化，并提供在线控制台及日志，进行故障分析与排查；
2. 容器云平台：以应用为中心，简化上云过程，为应用提供扩容、缩容、升级、回滚等功能，支持服务发现、负载均衡；

用友云开发者中心架构如下：



# 本版修订

|  |  |
| --- | --- |
| **DevOps平台** | |
| **关键改进点1** | 支持对开发、测试、灰度和生产环境上部署各版本应用，支持适配各环境的配置文件。 |
| **关键改进点2** | 支持多种类型应用，包括PHP、Python、Go、Nodejs、静态网站（nginx）、Java、Java Web、Dubbo |
| **关键改进点3** | 持续集成时，支持将构建完毕的应用上传到开发者中心的镜像仓库并提取应用包中的配置文件到配置中心，可以更新应用包并按功能和应用权限显示当前用户的被赋权的应用以及当前用户部署的应用的清单 |
| **关键改进点4** | 持续部署：支持升级回滚，滚动升级，一键升级，快速回滚；支持分组管理已部署应用；支持持续部署和改进持续部署 |
| **关键改进点5** | 支持租户隔离的镜像仓库，支持私有镜像仓库的权限控制并可推送本地构建的代码到私有镜像仓库并分页加载展示 |
| **关键改进点6** | 支持收集用户操作过程的各种错误，异常，用于诊断问题 |
| **关键改进点7** | 支持跨主机跨资源池迁移应用 |
| **关键改进点8** | 支持在线查看应用的运维数据 |
| **关键改进点9** | 支持资源池对主机接入报警，被部署应用的服务失效或应用状态报警，同时增加了异常到正常状态下的报警提示 |
| **关键改进点10** | 支持泛域名解析 |
| **关键改进点11** | 支持应用运行日志收集，统一存储到ES中 |
| **容器云平台** | |
| **关键改进点1** | 支持管理员登录后自助创建资源池，管理资源池内自有主机和云主机 |
| **关键改进点2** | 资源池实现权限控制并提供了资源池回收和续期功能，支持支持Marathon资源池 |
| **关键改进点3** | 中间件服务支持权限控制且支持超级管理员的中间件管理功能 |
| **关键改进点4** | 支持根据RPS压力动态变化，扩缩应用实例，同时支持在应用异常，主机异常和资源池异常情况下恢复应用 |
| **关键改进点5** | 支持应用升级信息展示和应用变更操作记录（包括应用构建打包、升级，发布、重启，扩容、缩容） |

# 产品特性

|  |  |  |
| --- | --- | --- |
| **产品** | **模块** | **子模块** |
| 用友云开发者中心 | DevOps平台 | 具体参见【iuap云运维平台V3.5-三级条目.xlsx】 |
| 容器云平台 |

# 产品范围

## DevOps平台

### 持续发布

|  |  |  |
| --- | --- | --- |
| 模块 | 功能 | 描述 |
| 应用多环境 | 支持对开发、测试、灰度和生产环境灵活部署应用 | 支持对开发、测试、灰度和生产环境部署各版本应用，支持适配各环境的配置文件。 |
| 应用多类型 | 支持多种类型应用 | 应用类型包括PHP、Python、Go、Nodejs、静态网站（nginx）、Java、Java Web、Dubbo |
| 持续集成 | 上传应用 | 支持将构建完毕的应用上传到开发者中心的镜像仓库。 |
| 上传应用，应用包增加描述文件，提取配置文件到配置中心 | 支持将Java Web应用War包中的配置文件信息同步提取并按照环境类型和版本号将配置文件分类上传至配置中心 |
| 更新应用包 | 在已上传应用详情页上，更新应用包（重新上传），可指定新版本号；部署时可以选择具体的版本的war包或镜像进行部署；展示当前任务构造出来的镜像；点击某一镜像版本时，能够看到当时构建的日志 |
| 按功能和应用的权限控制，持续集成 | 显示当前用户的被赋权的应用以及当前用户部署的应用的清单 |
| 持续部署 | 升级回滚，滚动升级一键升级，快速回滚 | 已部署的应用支持上传新版本后，一键升级。多个版本的应用支持版本升级和回滚。 |
| 全面的应用管理 | 应用管理的主界面（资源池、CPU、磁盘、内存、实例数等），应用的管理详情界面（创建、扩缩、销毁、配置修改、运行日志、域名） |
| 完善应用事件管理，已部署应用详情完善 | 已部署的应用显示事件信息，包括人员操作和系统事件 |
| 支持分组管理，已部署应用通过分组管理 | 支持产品线，产品，应用和微服务个层级的应用管理。 |
| 持续构建部署 | 支持将应用部署，升级，回滚，扩缩，重启暂停，启动。 |
| 镜像仓库 | 镜像仓库 | 镜像仓库，实现从war包到镜像的完整过程，镜像能够统一管理，租户隔离，简单、易用。 用户上传的war包能够自动生成DockerFile 基于DockerFile自动生成镜像 镜像的版本号在仓库中的管理 war包有个界面单独管理 有应用部署的按钮，快捷操作 |
| 推送、发布本地构建的代码至私有镜像 | 应用推送、发布本地构建的代码 Docker镜像推送到私有仓库 查看docker push镜像 增加docker pull权限控制 OSS至共享存储 |
| 选择配置文件，基于版本部署应用 | 选择配置文件，选择不同的版本、不同的环境进行部署，部署失败提示 已部署应用卡片，有环境标识 |
| 按功能和应用的权限控制 私有仓库 | 显示当前用户的被赋权的应用以及当前用户部署应用的私有镜像记录 |
| 私有镜像仓库分页显示的 | 私有镜像记录越来越多，加载时影响性能，分页显示 |
| 控制台 | 用友云控制台（Dashboard） | 显示当前租户用户的资源使用状态,应用健康状态，快捷入口，常见问题，资源报警和并显示最近访问的个应用的PV，UV和RT值 |
| 按功能和应用的权限控制 控制台 | 显示当前用户的被赋权的应用以及当前用户部署应用的资源使用情况 |
| 用友云控制台（Portal）框架 | 实现开发者中心的基本功能，含菜单，注册开通，权限，租户，登录注销 |
| 应用管理 | 部署应用 | 将已创建的应用部署到资源池上，同时可选取配置文件，指定资源占用上限 |
| 在应用详情页，选择某一版本，回滚至该版本 | 支持已部署的应用回滚至指定版本，同时可选择适配相关版本的配置文件。版本区隔以部署时间为准，而不是版本号。 |
| 用户主机定时轮询拉取最新agent镜像并应用 | 用户自有主机上配置定时轮询功能，拉取最新agent镜像并自动部署，达到自动更新的目的 |
| 在已部署页签上的每个应用标签中显示资源池和所属主机，在已部署页签下再按照资源池和主机或者应用等二级页签对应用分类显示 | 在已部署页签上的每个应用标签中显示资源池和所属主机的下拉框，点选后可过滤出符合条件的应用。在已部署页签下再按照资源池和主机或者应用等二级页签对应用分类显示 |
| 发布管理模块增加内存限制 | 防止部署应用时内存超出资源池可用内存上限导致死机，对部署应用时内存加限制。 |
| 应用部署支持Dubbo | 应用部署支持Dubbo类型的应用 |
| 应用容器监控 | 监控容器，应用管理>监控tab页功能完善；  区分实时监控和小时监控。含CPU，内存信息； 图表统一为echarts，风格保持一致。 |
| 应用容器在线运维 | 容器控制台，支持用户对运行中的实例通过命令行操作进行维护。 |
| 错误中心 | 收集用户在操作过程的各种错误，异常。 用于我们自己的运维做诊断 | 收集用户在操作过程的各种错误，异常。 用于我们自己的运维做诊断 |

### 运维白屏化

|  |  |  |
| --- | --- | --- |
| 模块 | 功能 | 描述 |
| 动态迁移 | 支持跨主机跨资源池迁移应用。 | 支持在多个资源池之间，在旧资源池停止旧应用实例，在新资源池启用应用实例 |
| 监控 | 在线查看运维数据 | 用户查看应用的监控信息包括：页面访问量，用户访问量，响应时间，网站流量和全国地区访问量图；查看用户行为分析结果：浏览器性能分析结果；应用业务分析结果 |
| 报警中心 | 资源池对主机接入报警 | 资源池主机掉线或者性能负载超过上限时，通过邮件短信向指定人员报警 |
| 服务报警 | 被部署应用的服务失效，通过邮件短信向指定人员报警 |
| 报警中心增加异常到正常状态下的报警提示,短信和邮件 | 当各种报警情况由异常转为正常时，通过邮件短信向指定人员提示 |
| 应用状态报警 | 应用掉线，状态异常后通过邮件短信向指定人员报警 |
| 事件中心 | 应用变更操作记录 | 应用构建打包、升级，发布、重启操作事件记录应用扩容、缩容事件记录 |
| 变更大盘 | 应用升级信息展示 | 查看应用升级信息 查看应用升级操作记录 |
| 配置中心 | 修改应用的配置，管理配置中心 | 修改应用配置，从应用名称、环境类型和版本号三个维度管理应用的配置文件 |
| 中间件服务超级管理员监控 | 超级管理员可以查看管理所有中间件服务的状态，可以销毁和续期。 |
| 按功能和应用的权限控制 配置中心 | 显示当前用户的被赋权的应用以及当前用户部署的应用的配置中心 |
| 前端 - 配置中心修改配置文件 | 支持在配置中心修改配置文件，支持在应用管理的配置中心修改线上配置文件。 |
| 应用部署按照不同环境区分，展示上标注 | 在应用管理的配置文件页签上显示配置文件时，按照不同环境类型和版本号区分，展示上标注 |
| 配置中心适配微服务框架 | 支持微服务的配置文件管理 |
| 日志收集  （ES） | 应用日志统一处理 | 应用运行的日志，统一存储到ES中，用自己的逻辑展示，可按照关键字搜索 处理不同volume的情况 |
| 操作日志，在开发者中心页面上埋点 | 在开发者中心上的所有操作记录日志，全部保存到数据库，用于审计，统计用户前端的操作热度 |
| 消息中心 | 短信服务，后台管理细化 | 短信服务支持授权，邀请码，资源池，报警中心等短信通知 |
| 支持记录和报告关键操作 | 构建完成，部署完成，出错，扩容、缩容完成 。重启完成，右上角有个消息。 |

## 容器云平台

|  |  |  |
| --- | --- | --- |
| 模块 | 功能 | 描述 |
| 混合云服务 | 资源池管理（自助添加模式） | 支持管理员登录后自助创建资源池，管理资源池内自有主机和云主机。 |
| 资源池内主机管理 | 资源池管理界面，把服务器加到资源池里，创建新的资源池，添加服务器，资源池展示距离多长时间被回收，以及资源池内资源使用状态----CPU，内存，磁盘，容器数等。一个资源池可以包含多台主机，主机可以在多个资源池之间迁移。 |
| 创建资源池 | 支持管理员登录后申请资源池，系统管理员审核后，管理员可以获得有时限的资源池。管理员可以创建多个资源池。 |
| 按功能和应用的权限控制 资源池 | 显示当前用户的被赋权的资源池以及当前用户创建的资源池 |
| 将主机添加到资源池过程中的状态展示 | 向资源池添加自有主机或者云主机时需要显示反映添加过程进度的日志信息，添加成功或者失败要有提示。 |
| 资源池续期功能 | 使用阿里云主机的体验资源池只有24小时试用期，试用期满后自动收回，但是系统管理员可以为即将到期的资源池续期，续期时段最小15天，最长365天。 |
| 资源池回收 | 当资源池到期或者资源池主机被迁出后，支持删除资源池，资源池所占资源被释放，同步更新管理信息。 |
| 服务自愈 | 支持在应用异常，主机异常和资源池异常情况下恢复应用 | 支持在应用异常，主机异常和资源池异常情况下恢复应用的策略 |
| 中间件服务 | 缓存服务Redis | 输入服务名称、描述、用户名称、密码等，并选择合适的配额，创建服务。默认每个租户允许创建个服务实例创建成功后在中间件上启动Redis缓存服务的实例 |
| 数据库服务MySQL | 输入服务名称、描述、用户名称、密码等，并选择合适的配额，创建服务。默认每个租户允许创建个服务实例。创建成功后在中间件上启动Mysql的服务实例.打开外网管理地址链接，进入phpMyAdmin登录界面 |
| 分布式应用程序协调服务Zookeeper | 输入服务名称、描述、用户名称、密码等，并选择合适的配额，创建服务。默认每个租户允许创建个服务实例创建成功后在中间件上启动Zookeeper的服务实例 |
| 消息服务RabbitMQ | 输入服务名称、描述、用户名称、密码等，并选择合适的配额，创建服务。默认每个租户允许创建个服务实例.创建成功后在中间件上启动RabbitMQ的服务实例 |
| 超级管理员的中间件管理功能 | 支持超级管理员登录后管理已存在的中间件服务，支持服务实例的销毁和续期。 |
| 构建服务Jenkins | 定制Jenkins实例，输入服务名称、描述、用户名称、密码等，并选择合适的配额，创建服务。默认每个租户允许创建个服务实例.创建成功后在中间件上启动Jenkins的构建服务实例 |
| 按功能和应用的权限控制 中间件服务 | 显示当前用户的被赋权的中间件服务以及当前用户部署的中间件服务 |
| 负载均衡服务Nginx | 使用Nginx实现负载均衡。输入服务名称、描述、用户名称、密码等，并选择合适的配额，创建服务。默认每个租户允许创建个服务实例.创建成功后在中间件上启动Nginx的负载均衡服务实例.负载均衡实例支持权限,销毁,续期,域名管理和转发策略 |

# 产品主要功能

## DevOps平台

### 持续交付

#### 代码构建

提供简单配置即可使用的构建平台，实现代码编译构建云端化，可视化配置的构建服务。支持从云端Git仓库中拉取代码进行构建，一键构建应用新版本。同时支持使用http协议进行用户名密码验证登录和使用ssh协议进行私钥验证登录。构建成功的应用包自动挂载到应用持续集成的新版本号中，形成自动化构建交付流程。

#### 持续集成

支撑企业实现持续交付，缩短交付周期，提升交付效率。以可视化的方式实现了应用的集成自动化，并提供在线控制台及日志，以进行故障分析与排查。统一开发、测试、灰度和生产环境的部署和管理，通过持续集成平台，结合容器技术，进行应用构建的统一管理。同时，开发者中心构建生成的应用镜像将统一发布到镜像仓库中实现了应用的统一管理。

#### 镜像仓库

共享容器生态，尽情无限探索，镜像仓库中的应用开箱即用，随需部署各类应用。镜像仓库分为公有和私有两部分，公有镜像仓库对开发者常用的镜像进行了集成，如nginx,tomcat,Alpine Linux等。开发者可以便捷的从开发者中心的公有镜像仓库中进行pull操作或直接部署公有镜像至资源池中。私有镜像仓库保存的是用户所属租户所上传的镜像，是开发者进行自有应用docker镜像管理的节点，在私有镜像仓库中开发者可以进行镜像的浏览、删除，并且可以从镜像仓库进行应用镜像的部署。

#### 持续发布

提供基于持续集成构建的Docker镜像为应用交付载体的应用持续发布，通过丰富的可视化自主配置项实现用户自定义、自组装完成高自由度的应用一键发布任务。持续发布的镜像存储于镜像仓库中，满足应用一次构建，到处运行的需求。同时支持一键发布开发、测试、灰度、生产环境的部署任务，并支持应用部署日志查看、搜索以及任务事件的追溯。持续发布还提供了简单高可用的域名管理，用户可以自由选择使用平台提供的自动生成的三级域名或自由绑定自有域名，实现互联网应用的域名配置。

#### 发布大盘

通过租户下的容器相关信息收集，在发布大盘中直观的展示租户所有容器的运行状况和资源利用情况。同时提供了快速创建持续集成实例的入口和其它相关平台的快捷入口和资源报警功能。此外，包含容器健康状况比例图的展示，同时还提供了每个租户下访问量TOP3的应用信息展示，如UV、PV、RT等信息，便于进行查看大盘实时情况。开发者中心的常见问题解答也集成在发布大盘中，便于用户了解掌握。

#### 版本管理

支持关联镜像版本的应用版本管理， 便于用户进行可视化的版本管理和切换。应用版本对应了不同的应用包或使用的不同基础镜像，通过应用关联查询的不同版本，统一了应用版本管理的入口，实现了应用管理的统一化、标准化，简化对应用进行版本管理的过程，提升应用迭代效率。

#### 升级回滚

提供基于应用版本管理的应用静默升级和回滚，通过镜像仓库的镜像支持，可以允许用户选择指定版本的应用进行升级或回滚。同时应用的升级和回滚默认使用蓝绿发布模式进行，可以完成零宕机版本切换，使用户完成平滑的线上版本变更，提升了应用的用户体验和服务可用性。

#### 蓝绿发布

基于蓝绿部署的理念，开发者中心提供了应用的蓝绿发布模式。该模式下应用的版本切换将使用老版本继续运行，部署新版本应用然后进行测试，确认新版本可用后将流量切换到新版本应用并将老版本应用关停的操作，在节省版本切换上线时间的同时保障应用可用。

### 运维白屏化

#### 自动化运维

通过全面的监控报警、日志收集、健康检查、服务自愈、泛域名解析及应用链路管理等，减轻运维负担。

##### 跨资源池动态迁移

支持将应用在多个资源池之间迁移。应用无需再次修改配置并手动部署，支持直接在新资源池中自动部署应用，当新实例部署成功后，再将旧资源池中的旧应用实例停止，保证应用可持续提供服务。

##### 实例管控

支持查看任何应用启动的实例，并对其进行管控，包括查看和调整应用实例数量，查看实例的ID、运行主机、健康状况、当前运行状态等信息。

##### 域名定义

支持定义应用的访问地址及域名。访问地址支持定义二级域名，可供其他应用间相互调用。支持用户将自有域名定义到应用，绑定的自有域名除要求已备案外，无其他硬性要求。

##### 在线终端

提供使用在线终端输入命令的功能。通过该功能可直接在浏览器中提供的Shell运行环境界面输入指令，查看并控制当前运行的容器。此在线终端不仅支持静态命令的输入与静态文本的输出，同时还支持类似top、vim等动态的命令，最大程度拟合终端版SSH连接工具，减少运维人员直接连接后台虚拟机带来的繁琐工作以及可能发生的隐形风险，同时也是开发和测试人员对应用进行测试时的有利工具。

#### 配置中心

提供对应用配置文件、配置项的可视化管理，提供利用SDK获取全量信息和配置变化的通知等功能。

同时，提供支持租户隔离的配置信息管控，管理员可以进行应用管理、配置管理、文件上传、在线修改、查看客户端连线状态、配置通知邮箱等操作。

#### 日志管理

支持自动提取和可视化呈现日志，并根据收集的海量异常日志快速搜索，以报表的方式展现业务的线上运营状况。提供简单、易用的日志分析及管理功能。其可对日志进行集中管理，以及准实时搜索、分析等简单易用的功能。同时，可进行线上业务的准实时监控、及时定位业务异常原因、排除故障，跟踪分析Bug等功能。

#### 业务监控

##### 应用监控

支持在线查看应用的运维数据，为查看运行应用的监控数据提供快速入口。包括查看实时过过去24小时内，应用占用的CPU、内存、网络流入、网络流出的曲线。通过监控曲线，可快速的了解应用在过去一段时间的资源占用情况。

#### 变更大盘

提供记录用户操作功能。用户对应用的每一个操作，如扩缩等均生成一条变更单，进行详细的记录。变更单包含了所有应用的变更。可根据变更单记录追溯用户操作历史，查看历史应用的变动，同时可以根据变更回滚至相应的版本。

#### 消息中心

在应用构建或部署完成，应用实例出错，以及对应用进行扩容、缩容操作完成后，系统实时收集各种类型的事件信息。

##### 报警中心

提供系统报警功能，用户只要设置好报警规则，根据相应的分组邮箱或配置的短信地址，可以做到对系统停机、CPU负载过高、系统剩余资源过少、服务异常中断或者中间件异常等场景的报警。

##### 事件中心

支持记录应用构建打包、升级，发布、重启操作事件，记录应用扩容、缩容事件记录，同时可展最后一次修改信息或任务失败原因等信息。

## 容器云平台

### 资源池管理

资源池是一组用来部署应用的主机的集合，它能够更加有效的利用主机或虚拟机的CPU、内存等资源。资源池是一个逻辑概念，可以根据不同需要，按相应维度创建自己的资源池，并向资源池中添加属于自己的主机。以资源池的方式进行计算节点管理，自由添加自有主机，进行智能化的资源调度与分配。能够同时管理数千个物理机或虚拟机节点。

### 容器服务

容器服务是容器云平台的核心能力，提供了Docker镜像部署后容器实例的管理功能，支持大规模容器集群的管理，为应用的微服务架构落地提供完备支撑。

总体以应用为中心，简化上云过程，为应用提供了容器调度、统一接入、扩容、缩容、升级、回滚等功能，具备自动服务网关、健康检查、服务自愈等特性。

### 容器调度

平台支持数万的容器调度，能够根据CPU、内存等资源，进行容器向主机的分发，实现资源利用率的最大化。

目前平台支持 Mesos和Kubernetes两种不同的容器调度器。

### 统一接入

为了方便应用的调试与访问，以容器方式部署的应用，需要能够以独立域名的方式暴露出来。本系统的统一接入模块，通过泛域名解析的方式，将请求转发到相应的容器。

通过实现无限二级域名功能，可以自动的创建出灵活的免费url转发，在IDC部门实现自动分配免费网址，实现企业内容器形式应用的快速接入。

### 服务网关

容器部署到环境中后，会产生大量的容器实例，为提高请求打到内部的调用效率，需要通过服务网关将请求分发到后端的各个容器。

平台能够将部署后的容器IP及端口信息，自动的保存到转发配置中。从而实现服务网关的自动化管理。

### 弹性伸缩

以容器运行的应用，可以将微服务拆分为很细的粒度，这样在面对不同的访问量时，可以按需增加服务实例，以提高系统的响应处理能力。本平台可以根据需要实时产生容器服务节点，手动实现快速的扩容缩容。也可以通过监控各容器的资源消耗情况（CPU、负载、内存等），根据相应的策略，可自动进行应用实例的伸缩。

适合业务量不断波动的应用服务，在业务高峰时，增加服务实例，在业务低谷时，减少服务实例，从而提高资源的综合利用率。

### 服务自愈

服务自愈是针对容器发生故障自动处理的解决方案，能够提升企业服务的可用性，并降低故障处理带来的人力投入。

系统能够不断的监控每个Docker实例的健康状态，当检测到异常的实例时，自动的杀掉异常的实例，并启动新的实例补充进来，恢复到业务集群的正常状态。

### 服务健康检查

平台通过健康检查的机制，来判断容器实例的可用性。提供HTTP、TCP两种检测方式。

针对七层（HTTP）监听，健康检查通过HTTP HEAD方式进行检测，通过返回的状态码，判断健康情况。这种检查方式，可以在检查的URL逻辑中，编写代码自行实现检查的细节。

针对四层（TCP）监听，健康检查通过IP+端口方式进行检测，通过TCP探测，获取健康情况。这种检查方式，主要进行粗粒度的服务检测。

为避免频繁的健康检查造成的服务压力影响，需要能够控制检测的规则，在多次检测成功或失败后，才切换健康状态。健康检查可以设置开始检测时间、轮询检测的时间间隔、响应超时时间、最大失败次数等信息。

### 环境变量管理

在编写Dockerfile时， 可以通过ENV来为容器设置相关的环境变量，设置的环境变量在容器运行时可以通过环境变量来进行引用。

本平台提供这种能力，可以在应用部署时为容器设置环境变量，这样应用程序（尤其基于Spring开发的java应用），可以从环境变量中获取应用的配置信息。

### 持久存储管理

通常一个服务的升级流程是杀掉旧版本的容器实例，然后拉取并启动新版本容器。默认情况下，旧版容器包含的所有数据都会被消除。

有状态的容器都有数据持久化需求，将容器中的数据输出到宿主机的存储中。本平台能够通过主机的磁盘映射（Volumes）的方式，实现容器的磁盘挂载。这种数据卷模式下，容器的文件保存在宿主机中。

### 容器网络管理

在平台部署的容器，支持host、bridge、overlay等网络模式，能够支持一容器一IP的特性，从而防止应用程序会被分配至非标准端口而造成端口冲突的状况。

### 中间件服务

平台提供主流的缓存、数据库、消息等中间件比如Redis服务，MySQL服务，RabbitMQ服务，ZooKeeper服务等，开发者可按需选用支撑服务，方便快速搭建开发测试环境。

本版本中间件服务已经受功能和应用权限控制，用户登录后，显示当前用户的被赋权的中间件服务以及当前用户部署的中间件服务。

支持超级管理员登录后管理已存在的中间件服务，支持服务实例的销毁和续期。