**UiBot Cloud 系统设计说明书**



**UiBot Cloud**

**系统设计说明书**

版本记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本号 | 日期 | 描述 | 修改 | 审核 |
| 0.1 | 2023年5月30日 | 创建文档 | 时爱家 |  |
|  |  |  |  |  |

1. **背景**

1.1 **目的**

本文档的架构设计部分从技术角度划分服务、层级和模块；

系统设计部分阐述各个业务子系统之间的交互方式和交互逻辑；

功能设计部分从业务角度阐述系统功能模块；

业务模型设计描述主要业务模型和模型字段；

接口设计描述具体接口。

本说明书的预期读者为涉及产品需求分析、架构、开发、测试与管理工作的所有人员，以及希望更深入了解UiBot 产品的售前、交付人员以及客户群体。

1.2 **范围**

本说明书主要介绍**UiBot Cloud**的Demo版本总体技术实现方案。

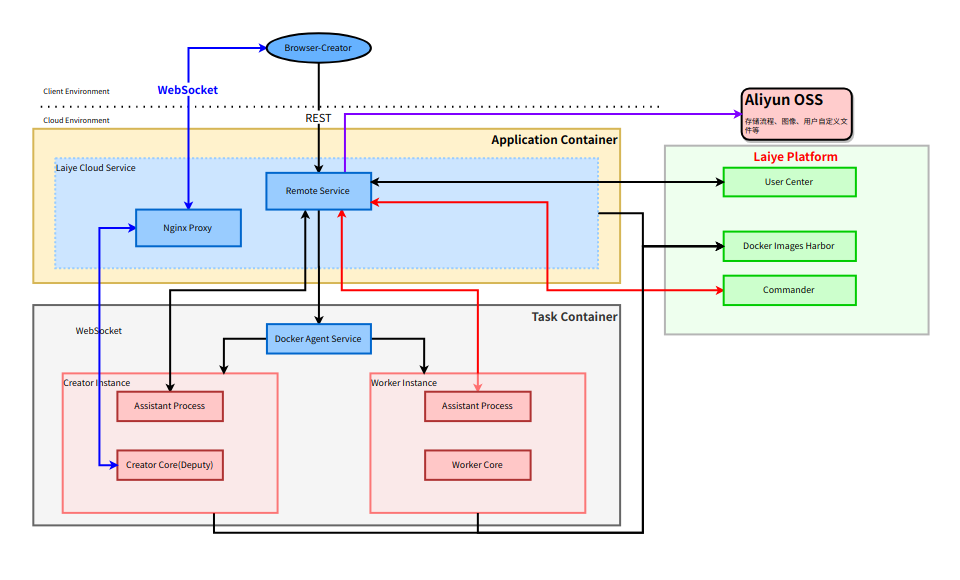
从架构设计，系统设计，功能设计，数据模型设计，接口设计等方面阐述。

**架构设计**

1.1 **服务架构**

描述各个服务之间的调用和依赖关系

1.1.1 **架构图**



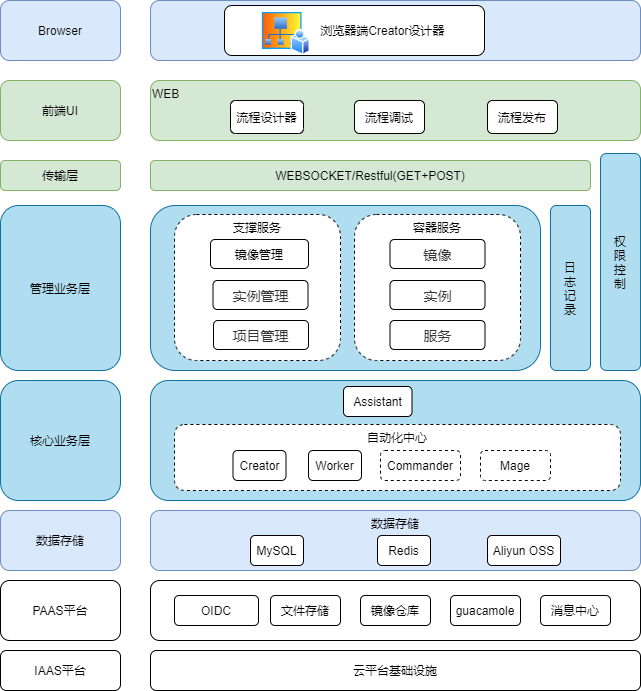
|  |  |  |  |
| --- | --- | --- | --- |
| 服务名称 | 服务类型 | 交互方式 | 描述 |
| Browser-Creator | 前端服务 | REST  WebSocket | 通过浏览器访问，部署在容器中 |
| Remote Service | 云服务 | REST | 权限、容器管理、项目管理等 |
| Docker Agent Service | 云服务 | REST | 调用docker swarm接口操作容器 |
| Nginx Proxy | 云服务 | WebSocket | 连接浏览器和Creator的WebSocket |
| Assistant Process | 代理进程 | REST | 接收支撑服务指令操作文件 |
| Creator Core | 核心服务 |  | 标准版UiBot Creator |
| Worker Core | 核心服务 |  | 标准版UiBot Worker |
| User Center | 三方服务 | REST | 用户中心 |
| Docker Images Harbor | 三方服务 |  | 容器镜像仓库 |
| Commander | 核心服务 |  | 云原生版UiBot Commander，待接入 |
| File System | 核心服务 |  | Laiye原生文件服务，待接入 |
| Aliyun OSS | 文件存储 | REST | 目前云原生服务使用的文件存储方式 |

1.1.2 **服务描述**

|  |  |
| --- | --- |
| **前端** | 根据标准版UiBot Creator 6.0改造，从Electron剥离：   1. 重构去除对Electron的本地依赖； 2. 改造文件存储功能，从本地存储改为云端存储。 |
| **应用后端** | 包括应用支撑服务、容器代理服务和Nginx代理：   1. 支撑服务提供镜像管理、实例管理、项目管理、日期管理、权限管理等云原生的基本功能，同时负责连接前端和自动化后端，通过Docker环境暴露的接口管理容器环境，提供弹性负载能力； 2. 容器代理服务将容器管理功能进行二次封装； 3. Nginx代理提供前端和容器内Creator的WebSocket连接。 |
| **自动化后端** | 自动化后端部署在容器环境内，云原生平台在预置的操作系统镜像中安装相应的Creator或Worker，通过应用后端管理容器实例的生命周期，这些容器实例完成自动化相关的任务。 |
| **来也平台** | 云原生平台与来也平台提供的部分功能进行集成，包括：OIDC用户认证、统一的文件存储、Docker镜像仓库、Commander等。 |
| **三方平台** | 云原生平台还依赖第三方平台的功能，如：阿里云OSS文件存储、CDN等。 |

1.2 **分层架构**

1.2.1 **架构图**

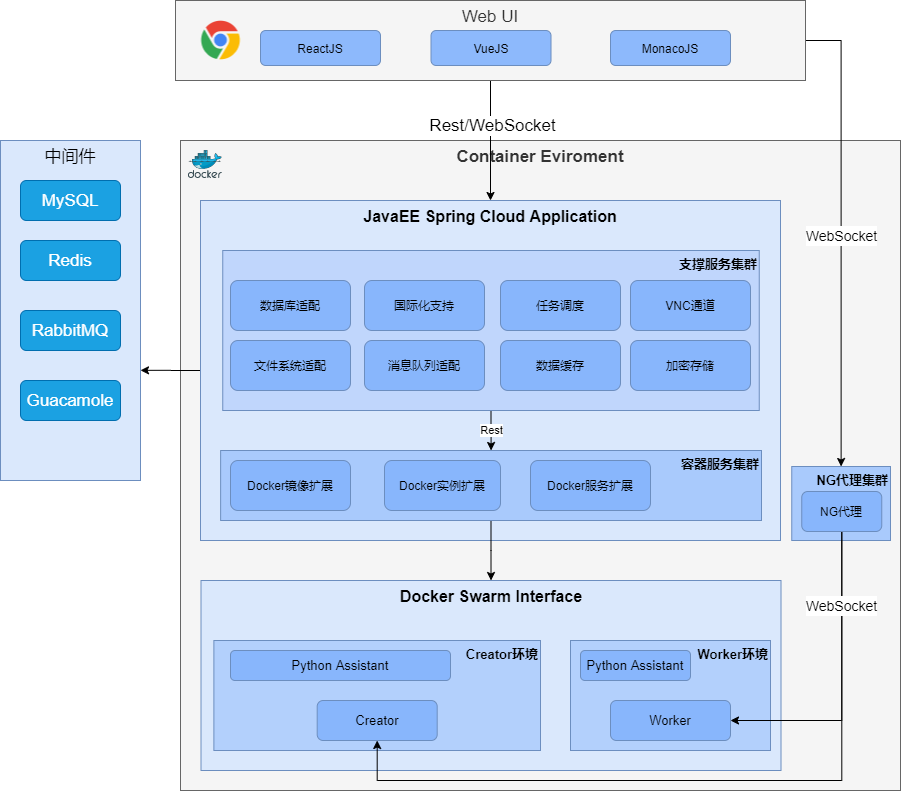


1.2.2 **描述**

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 描述 |
| Browser | 浏览器 | 目前仅支持Chrome浏览器 |
| 前端UI | 前端 |  |
| 传输层 | 云服务 |  |
| 管理业务层 | 云服务 |  |
| 核心业务层 | 核心服务 | 目前并未接入Commander和Mage |
| 数据存储 | 存储服务 |  |
| PAAS平台 | 中间件 |  |
| IAAS平台 | 基础设施 |  |

1.3 **技术架构**

1.3.1 **架构图**

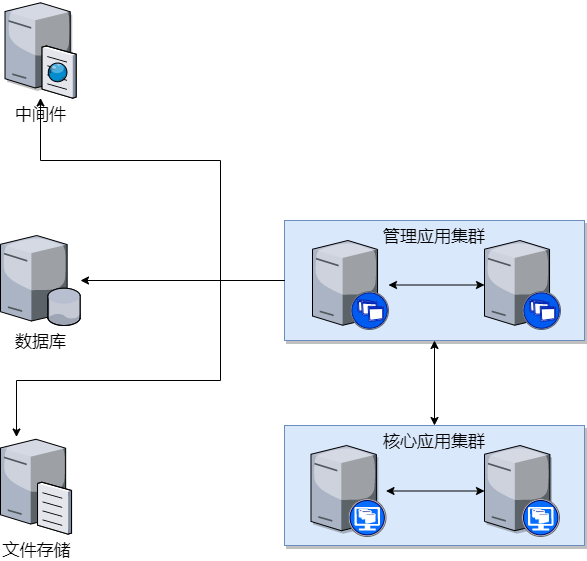


1.3.2 **描述**

|  |  |
| --- | --- |
| 名称 | 描述 |
| Web UI | ReactJS/VueJS/MonacoJS |
| 支撑服务 | 基于Java（1.8）EE Spring Cloud微服务框架，完成数据库适配/文件存储适配/消息队列适配/国际化支持等功能；  当前版本使用MySQL作为数据库，Redis作为缓存，RabbitMQ作为消息队列，Guacamole作为远程连接通道 |
| 容器服务 | 基于JavaEE Spring Boot SPI，封装Docker扩展能力，通过Rest方式调用Docker Swarm接口，主要提供容器镜像、实例管理，容器服务支持 |
| NG代理 | NG代理连接浏览器前端到容器内的Creator或Worker，实现元素抓取，运行，调试等核心功能 |
| 自动化助手 | 基于Python-3.8，负责支撑服务到自动化端Creator或Worker的连接，包括上传、下载、监控、心跳等功能 |

1.4 **部署架构**

1.4.1 **架构图**



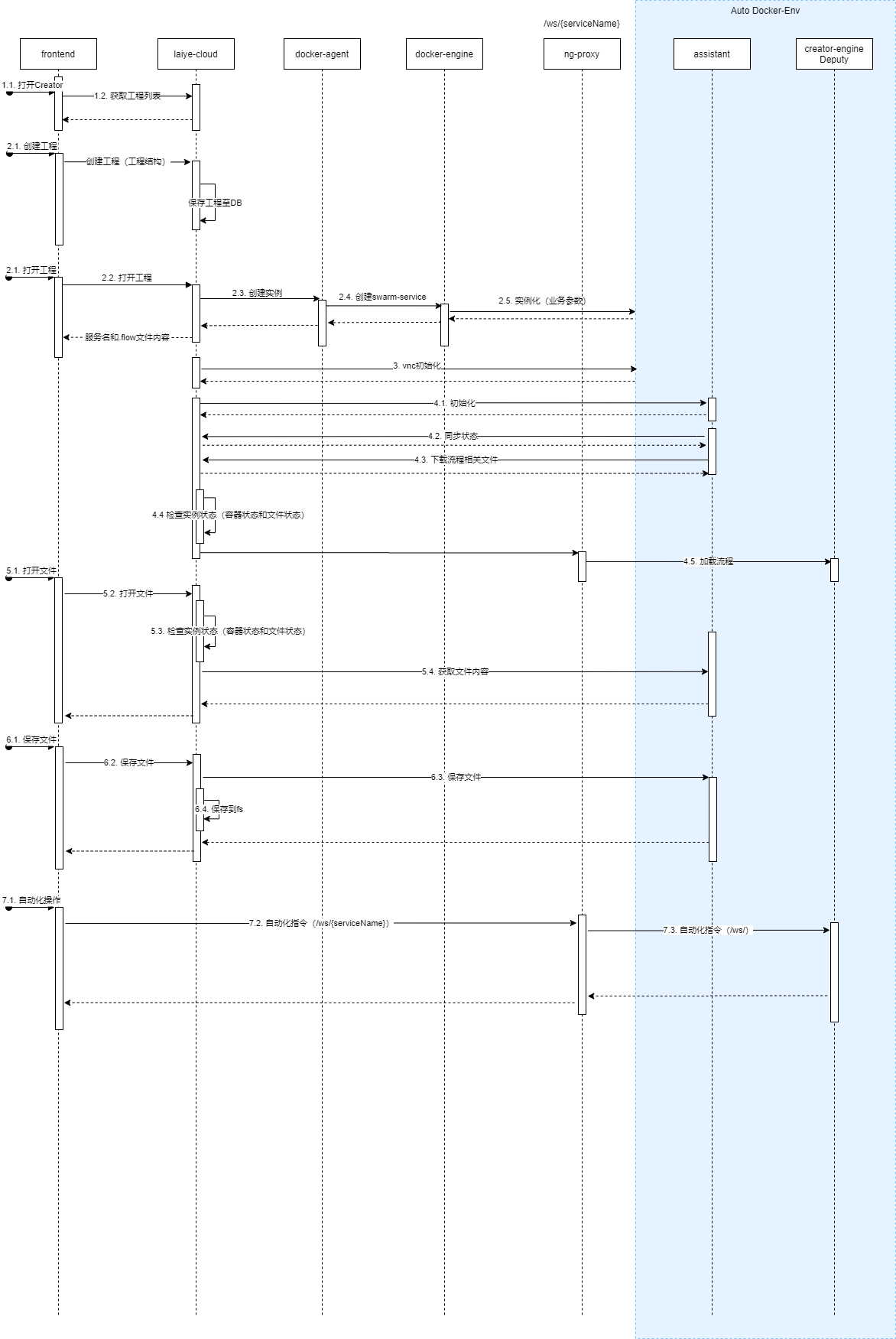
1.4.2 **描述**

|  |  |
| --- | --- |
| 名称 | 描述 |
| 管理应用集群 | 管理应用集群部署支撑服务、容器服务和NG代理 |
| 核心应用集群 | 核心应用集群部署Creator和Worker环境 |
| 数据库 | 持久化数据 |
| 文件存储 | 存储流程相关文件 |
| 中间件 | 提供缓存，消息队列等支持 |

**系统设计**

1.1 **Creator流程**

1.1.1 **流程图**



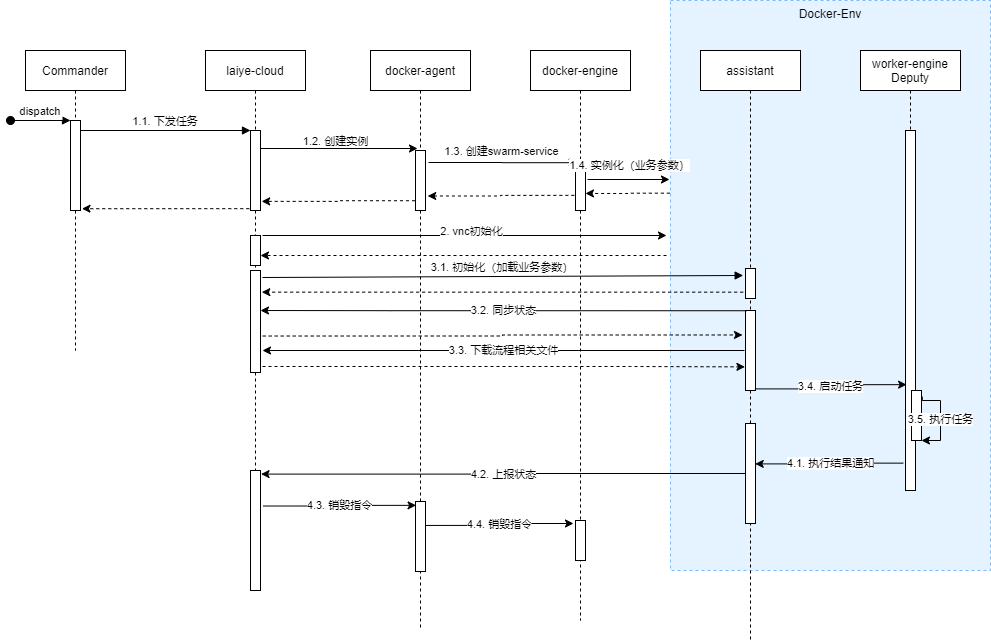
1.1.2 **描述**

|  |  |
| --- | --- |
| 名称 | 描述 |
| 前端 | 登录进入Creator，创建/打开流程，打开文件，保存文件，自动化操作 |
| 支撑服务 | 对容器服务的扩展实现弹性管理容器资源；  连接前端和核心管理层；  管理数据和文件存储。 |
| 容器服务 | 封装Docker Swarm调用 |
| 容器环境 | 基于Docker Swarm的Rest接口，提供容器镜像、实例、服务等功能扩展 |
| Creator助手 | 主要提供文件管理功能，为Creator将云端文件下载到本地和同步本地文件到云端 |
| Creator Core | Creator标准版 |

1.2 **Worker流程**

目前仅针对无人值守情况

1.2.1 **流程图**



1.2.2 **描述**

|  |  |
| --- | --- |
| 名称 | 描述 |
| Commander | 调度Worker的执行 |
| 支撑服务 | 对容器服务的扩展实现弹性管理容器资源；  连接前端和核心管理层；  管理数据和文件存储。 |
| 容器服务 | 封装Docker Swarm调用 |
| 容器环境 | 基于Docker Swarm的Rest接口，提供容器镜像、实例、服务等功能扩展 |
| Worker助手 | 主要提供文件管理功能，将任务所需文件下载到本地 |
| Worker Core | Worker标准版，执行任务 |

**功能设计**

|  |  |  |
| --- | --- | --- |
| 模块 | 功能点 | 描述 |
| 浏览器  Creator | 创建流程 |  |
| 流程设计 |  |
| 流程运行 |  |
| 流程调试 |  |
| 流程发布 |  |
| 支撑  服务 | Docker镜像管理 |  |
| Docker实例管理 |  |
| 项目管理 |  |
| 操作日志 |  |
| 角色权限管理 |  |
| 远程桌面连接 |  |
| 容器  服务 | Docker镜像扩展 |  |
| Docker实例扩展 |  |
| Docker服务扩展 |  |
| NG代理 | 代理WebSocket |  |
| 自动化  Creator  &  Worker  助手 | 下载文件 |  |
| 上传文件 |  |
| 监控文件变化 |  |
|  |  |
|  |  |

**业务模型设计**

1.1 **Docker镜像**

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 描述 |
| 编号 | 数字 |  |
| 名称 | 字符串 | 如：  dockercloud.uibot123.com:9190/laiye-cloud/chromevnc |
| 标签 | 字符串 | 如：latest |
| 用户 | 字符串 | 如：800166 |
| 创建时间 | 时间 |  |
| 备注 | 字符串 |  |

1.2 **Docker实例**

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 描述 |
| 编号 | 数字 |  |
| 镜像编号 | 数字 |  |
| 服务名称 | 字符串 | Docker服务的名称，全局唯一 |
| 服务类型 | 枚举 | 0: Creator; 1: Worker |
| 用户 | 字符串 |  |
| 登录密码 | 字符串 | 登录Docker实例的VNC密码 |
| 创建时间 | 时间 |  |
| 备注 | 字符串 |  |

1.3 **工程项目**

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 描述 |
| 编号 | 数字 |  |
| 名称 | 字符串 | 相同用户唯一 |
| 路径 | 字符串 |  |
| 类型 | 枚举 | 0: 流程; 1: 功能库 |
| 用户 | 字符串 |  |
| 置顶标记 | 布尔 | 用于排序 |
| 最后打开时间 | 时间 | 用于排序 |
| 创建时间 | 时间 |  |
| 备注 | 字符串 |  |

1.4 **角色权限**

1.4.1 **角色**

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 描述 |
| 编号 | 数字 |  |
| 名称 | 字符串 | 如：管理员 |
| 编码 | 字符串 | 如：Manager |
| 创建时间 | 时间 |  |
| 备注 | 字符串 |  |

1.4.2 **用户角色关系表**

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 描述 |
| 用户 | 字符串 | 来源于OIDC，如：800166 |
| 角色编号 | 数字 | 如：1 |

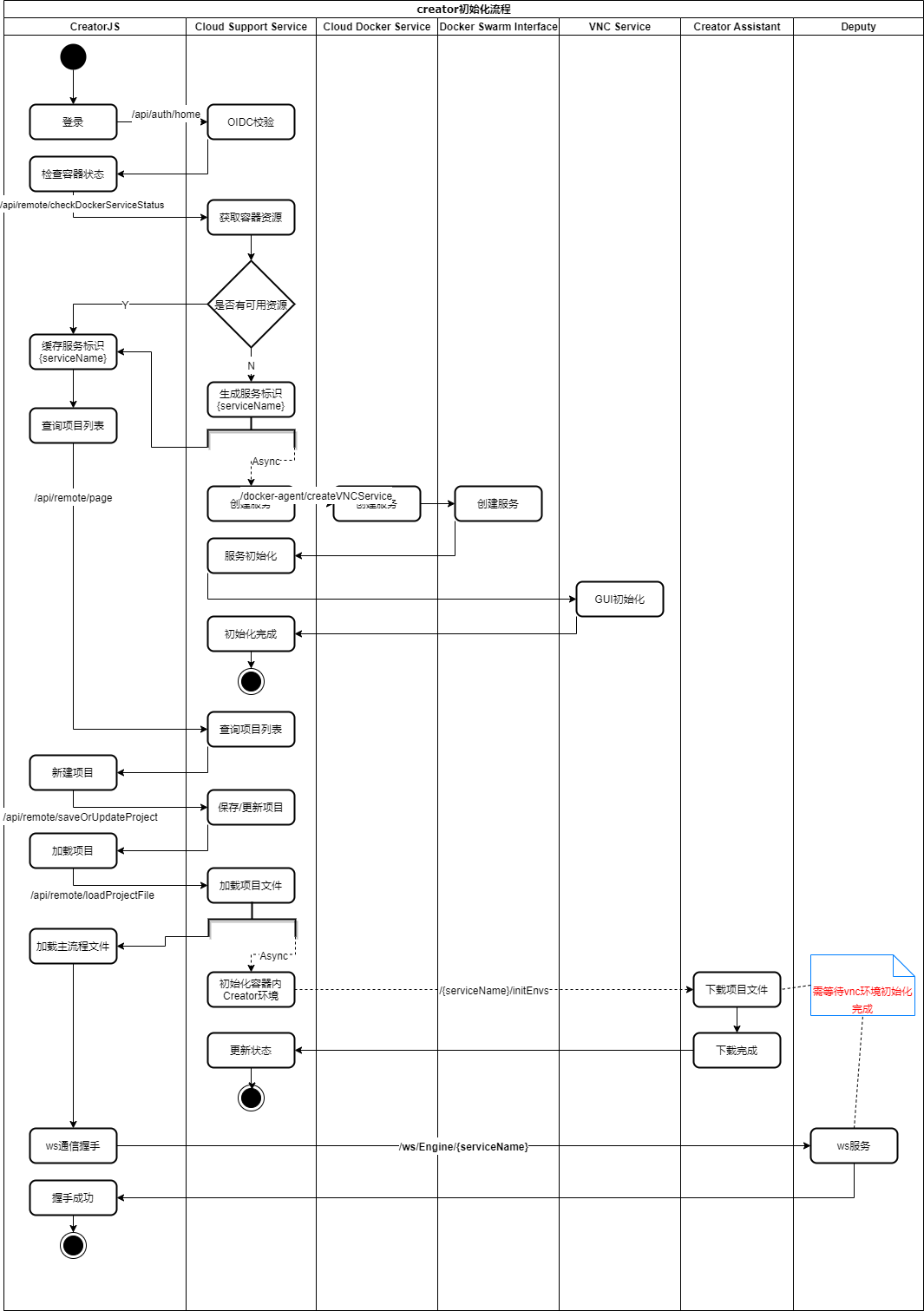
1.5 **操作日志**

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 描述 |
| 编号 | 数字 |  |
| 用户 | 字符串 | 如：800166 |
| 操作类型 | 枚举 | 如：创建镜像|修改实例 |
| 操作时间 | 时间 |  |
| 日志 | 字符串 |  |

**接口设计**

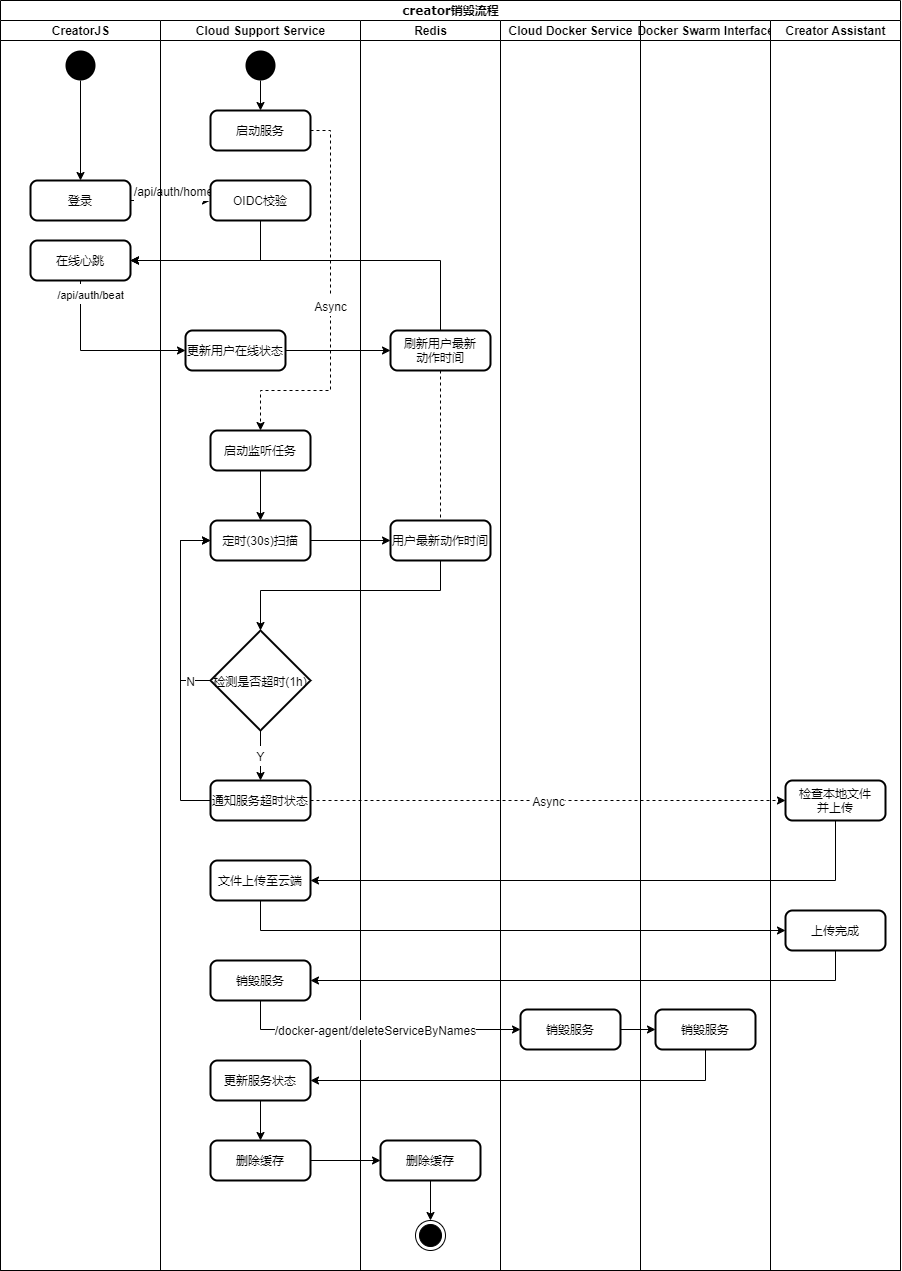
1.1 **核心流程**

1.1.1 **初始化**

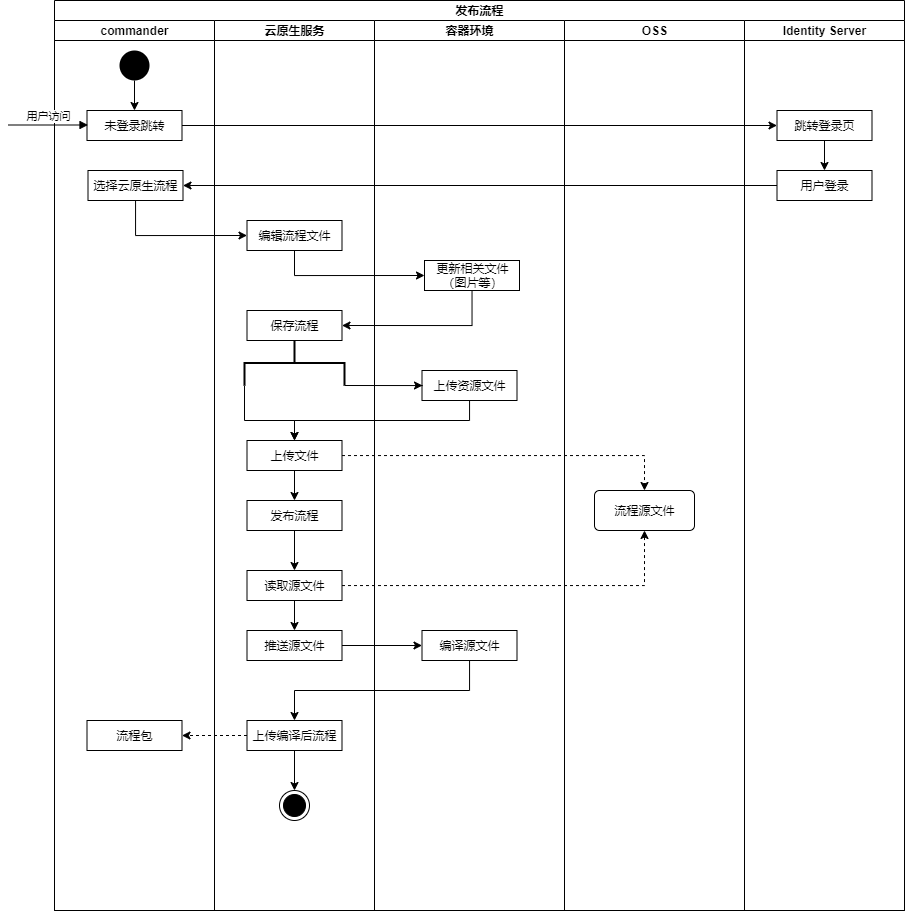


当前版本考虑容器资源的弹性负载，使用前分配资源，使用完成回收资源。

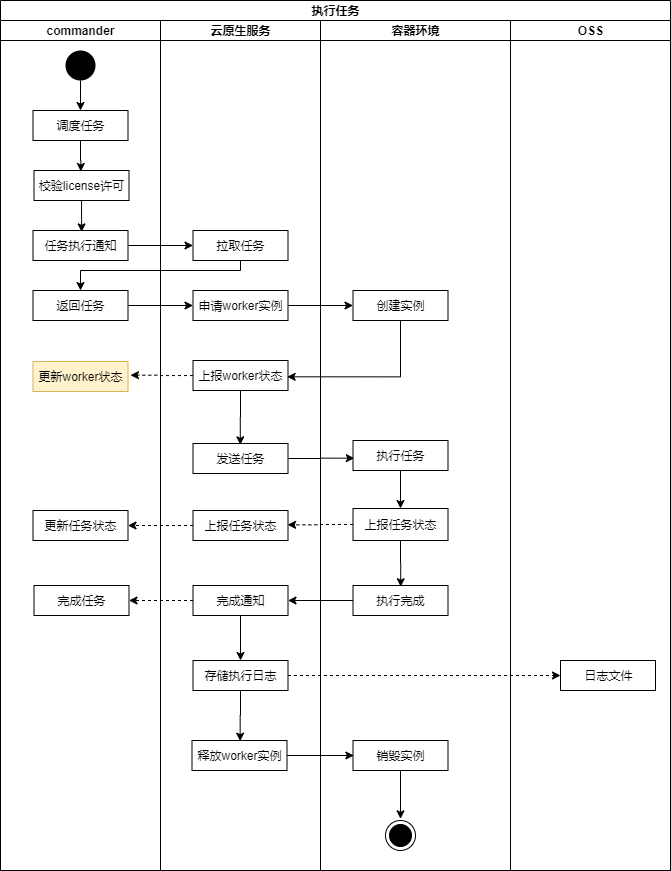
1.1.2 **销毁**



1.1.3 **发布流程**



1.1.4 **执行任务**



1.2 **支撑服务**

1.2.1 **登录校验**

|  |  |  |
| --- | --- | --- |
| 接口 | 参数 | 说明 |
| /api/auth/home |  | 登录校验，未登录则跳转到OIDC登录 |
| /api/auth/user |  | 获取登录用户信息 |
| /api/auth/beat |  | 前端心跳监测 |
| /api/auth/logout |  | 注销当前用户 |

1.2.2 **角色权限**

|  |  |  |
| --- | --- | --- |
| 接口 | 参数 | 说明 |
| /api/auth/role/page | page: 当前页  size: 每页数量 | 分页查询角色信息 |
| /api/auth/userRole/getByUserId | userId: 用户id | 根据用户id获取角色权限 |

1.2.3 **工程管理**

|  |  |  |
| --- | --- | --- |
| 接口 | 参数 | 说明 |
| /api/remote/project/saveOrUpdateProject | json | 保存或更新项目信息 |
| /api/remote/project/pathExists | json | 判断path是否存在 |
| /api/remote/project/checkDockerServiceStatus | json | 监测Docker服务状态 |
| /api/remote/project/loadProjectFile | json | 获取项目文件内容 |
| /api/remote/project/loadProjectDir | json | 获取项目目录文件列表 |
| /api/remote/project/page | json | 分页查询项目列表 |

1.2.4 **容器实例管理**

|  |  |  |
| --- | --- | --- |
| 接口 | 参数 | 说明 |
| /api/remote/dockerInstances/createVNCService | port: 端口  imageid: 镜像id  width: 宽  height: 高 | 根据镜像创建实例（服务） |

1.2.5 **开放接口**

开放接口访问无需登录校验

|  |  |  |
| --- | --- | --- |
| 接口 | 参数 | 说明 |
| /api/open/guacd |  | 打开guacdmole远程桌面 |
| /api/open/updateStatus | userId: 用户id  serviceName: 服务名称 | 提供给creator assistant回调，更新状态 |
| /api/open/uploadFlow | File: 二进制文件流  filePath: 文件路径 | 上传文件 |
| /api/open/downloadFlow | filePath: 文件路径 | 下载文件 |
| /api/open/listFlow | userId: 用户id  flowName: 流程名称 | 根据用户和流程获取文件列表 |

1.3 **容器服务**

该服务主要是对于docker api的封装，只提供有限的功能来保证安全。在部署上该服务只部署在docker swarm的master节点，只提供docker 内部network访问，不对外网暴露来保证安全。该服务主要提供了如下几个接口：

|  |  |  |
| --- | --- | --- |
| 接口 | 参数 | 说明 |
| /docker-agent/createVNCService | 镜像名，vnc密码，分辨率，环境变量等 | 创建并初始化vnc的swarm service。 |
| /docker-agent/deleteService | swarm service id | 删除swarm service |
| /docker-agent/images |  | 列出所有的docker镜像 |
| /docker-agent/service |  | 列出所有的swarm service |
| /docker-agent/inspectService | swarm service id | 返回swarm service 详情 |
| /docker-agent/getServiceByName | swarm service name | 根据name来查询service |

1.4 **Creator助手**

py-assistant主要负责和cloud做通信，来辅助deputy完成流程文件的上传下载。以及在work模式下负责和deputy通信，来调用deputy完成流程运行。详细介绍如下：

1.4.1 **Restful接口**

使用python的flask编写，主要提供了如下几个接口：

|  |  |  |
| --- | --- | --- |
| 接口 | 参数 | 说明 |
| /initEnvs | service\_name，userid，flow\_name等。 | 初始化系统环境变量，同时根据userid和flow\_name来下载流程文件并保存在容器内，下载完成后会通知cloud下载完成状态。 |
| /downloadFlow | 文件及文件路径 | 当用户在creator修改流程文件后，cloud会调用此接口来将用户最新文件上传到容器内。 |
| /uploadFlow | 项目目录 | 当用户在creator点击保存，或者在触发容器销毁之前，由cloud来调用此接口，会将用户工程目录中所有文件上传到cloud中，由cloud保存到fs上。 |

1.4.2 **fs-watchdog**

该功能会监听用户的流程目录，当该目录发生文件变动后，会将变动文件上传到cloud中，来保证用户在流程设计阶段所产生的中间文件等能准确的保存下来。

1.4.3 **ws-worker**

该功能是一个ws的client，负责和deputy通信，在work启动后，会以环境变量的方式传入userid，flowname等关键信息，由该客户端完成流程文件下载和保存，完成后通过ws通信的方式调用deputy来完成任务执行。

1.5 **NG代理**

该服务通过nginx来代理后端的deputy的ws，实现前端creator可以直接访问后端的deputy来实现流程的运行调试等。

nginx的核心配置如下：

|  |
| --- |
| server {  listen 80;  server\_name testcloud.uibot123.com;  location ~ /ws/(.\*)/(.\*) {  resolver 127.0.0.11;  proxy\_pass <http://$2:42222/$1>;  proxy\_read\_timeout 300s;  proxy\_send\_timeout 300s;  proxy\_set\_header Host $host;  proxy\_set\_header X-Real-IP $remote\_addr;  proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  proxy\_http\_version 1.1;  proxy\_set\_header Upgrade $http\_upgrade;  proxy\_set\_header Connection $connection\_upgrade;  }  } |

前端creator会通过类似于

|  |
| --- |
| **ws://testcloud.uibot123.com/ws/Engine/800074-HGXtY** |

这样的url来访问，其中 Engine 对应后端deputy的ws endpoint，800074-HGXtY对应docker实例的服务名称。