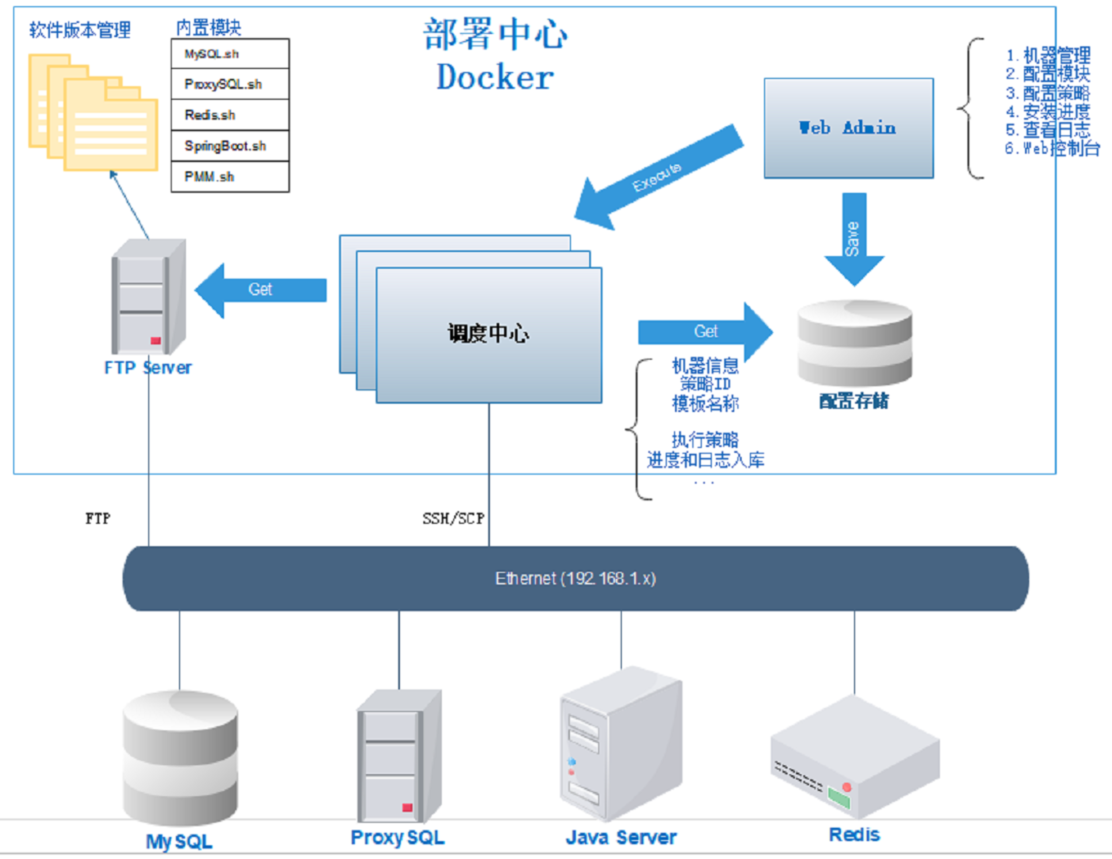
# 部署中心设计文档

# V1.1

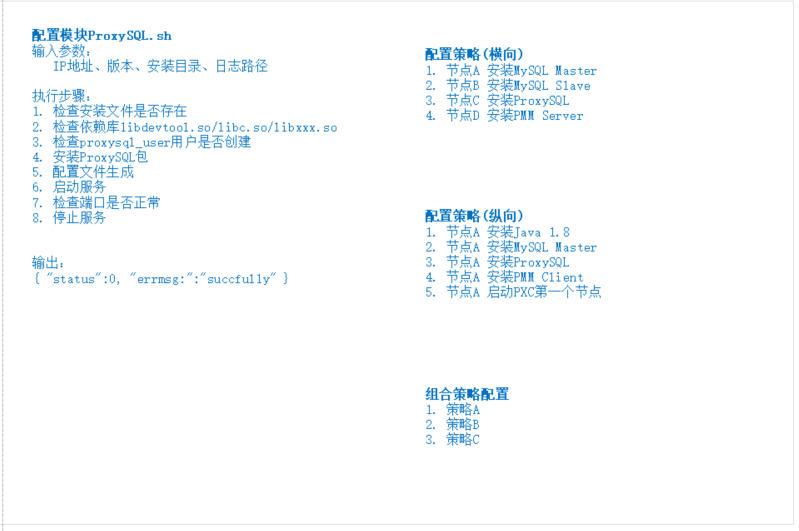
## 总体架构

部署中心以docker微服务的形式提供，分为三大模块:

* FTP软件仓库
* 调度中心
* Web管理



图一: 部署中心架构设计图



图二: 模块和策略例子

### FTP软件仓库

通过在Docker上启动FTP server，从外部挂载软件仓库的目录。软件仓库目录结构包括：

* 安装包管理 /repo/packages

Pakcages目录主要存放需要安装的软件，以及交易系统提供的微服务jar包。

* 应用管理 /repo/modules

Modules目录主要存放每个packages目录对应的软件需要的安装/更新脚本程序， 提供一系列的标准接口。

* 策略管理 /repo/scripts

Scripts目录主要存放策略脚本，每个策略包含了一批应用的安装部署执行过程。

### 调度中心

调度中心主要任务是接收到Web管理页面的请求，执行远程机器的管理和调度人物的操作。例如拷贝软件到远程、执行远程脚本、传输日志、监控机器服务等。

### Web管理

Web管理功能支持设备管理、应用管理、策略管理、绑定设备和策略，并提供Web控制台服务。

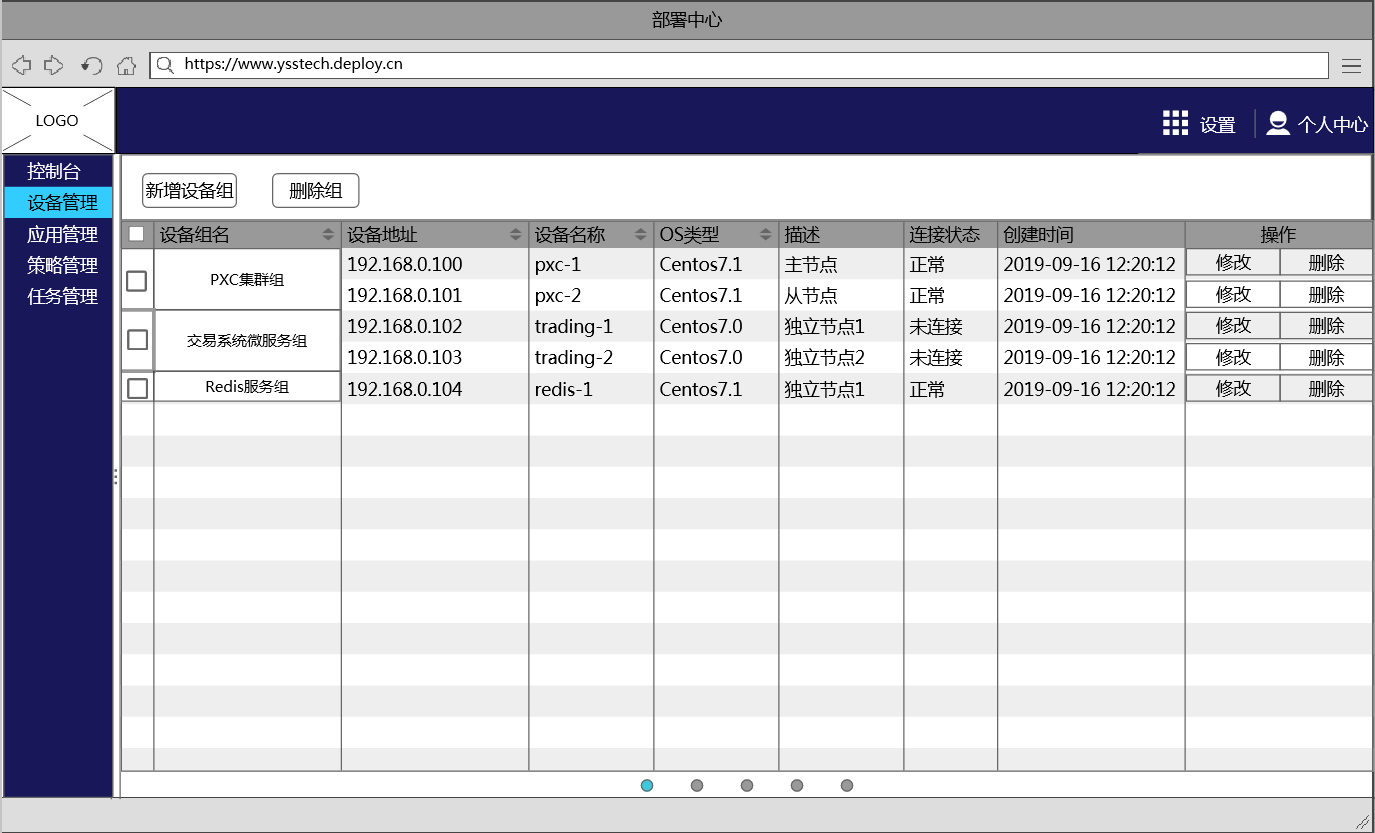
## 交互设计

#### 控制台

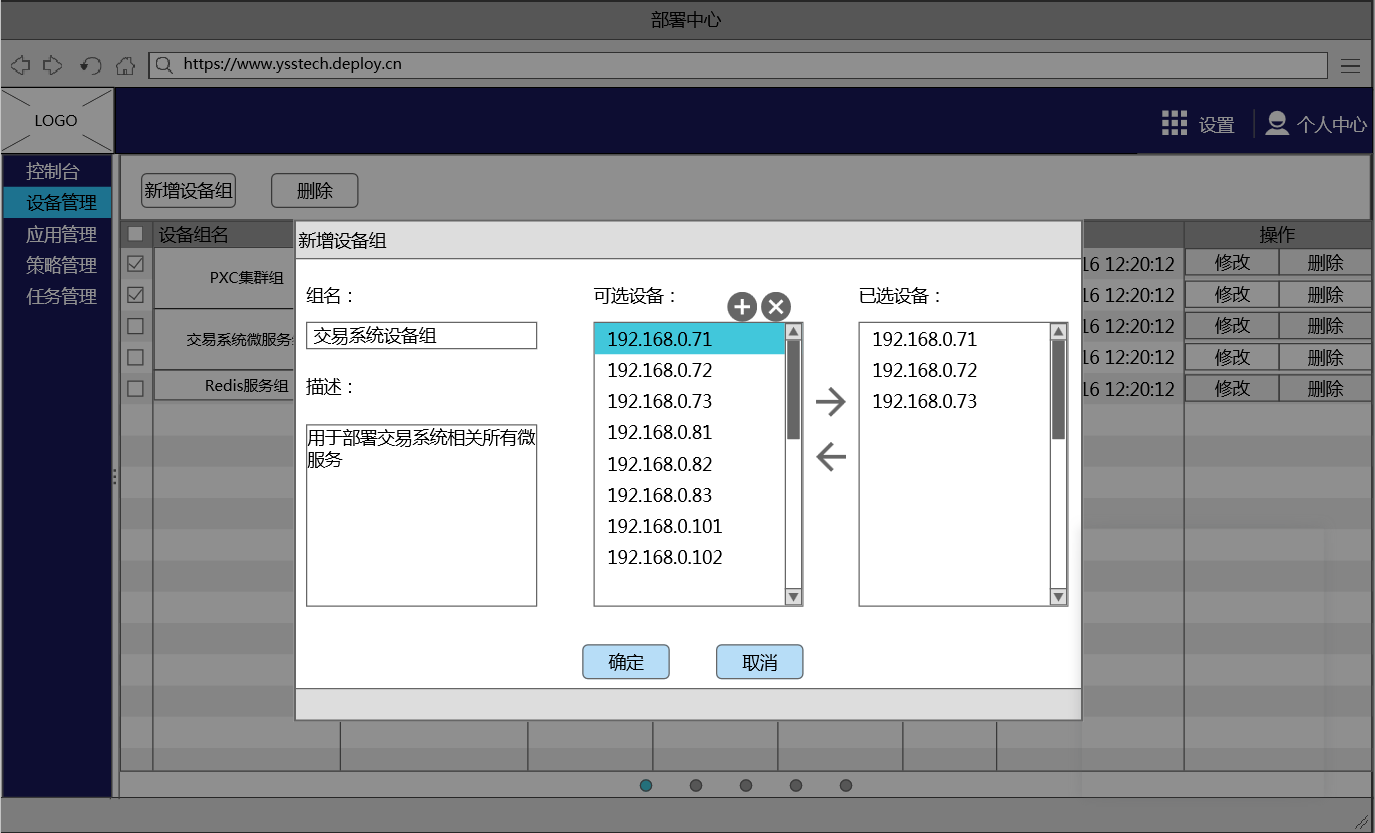


#### 设备管理

1.列表页



2.新增设备组

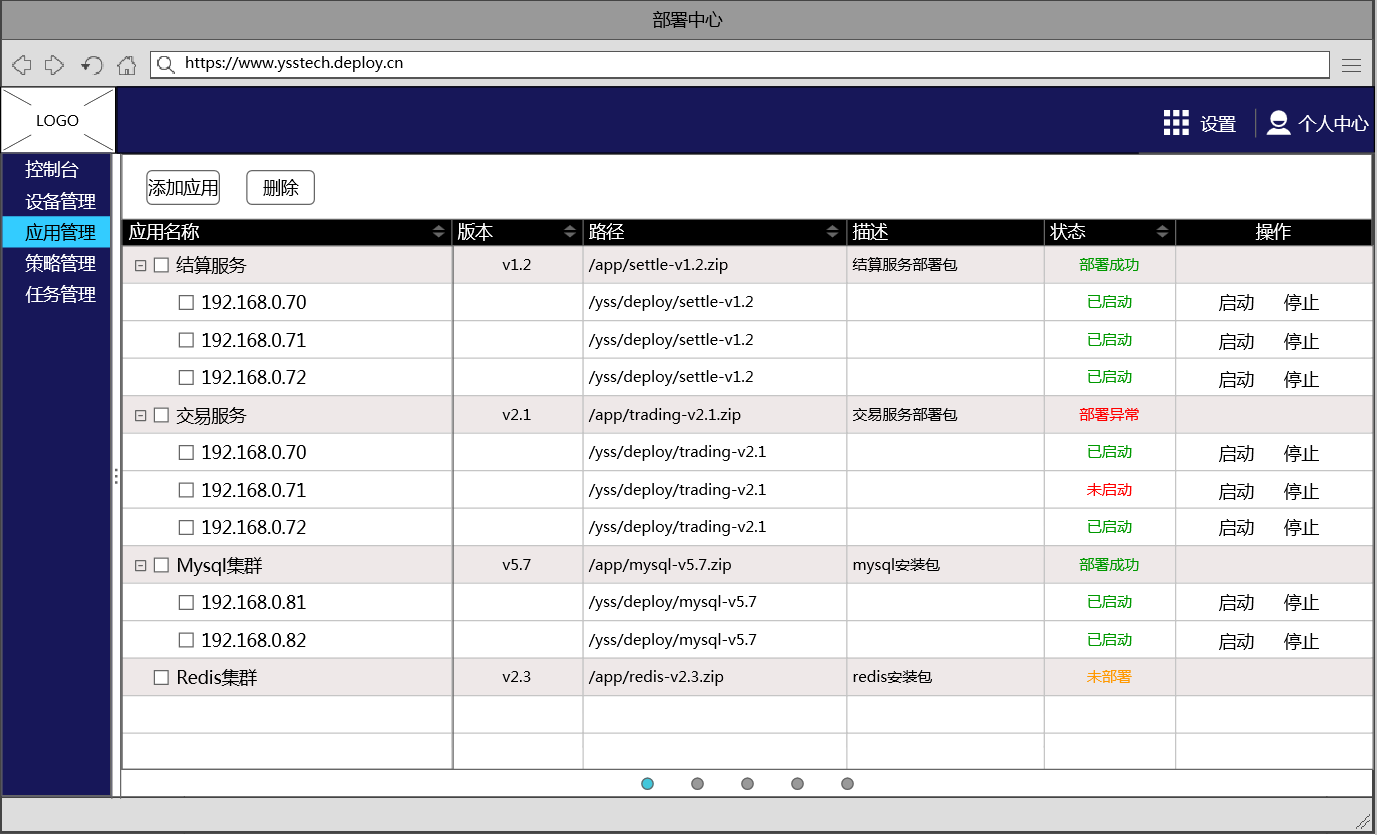


1. 新增设备

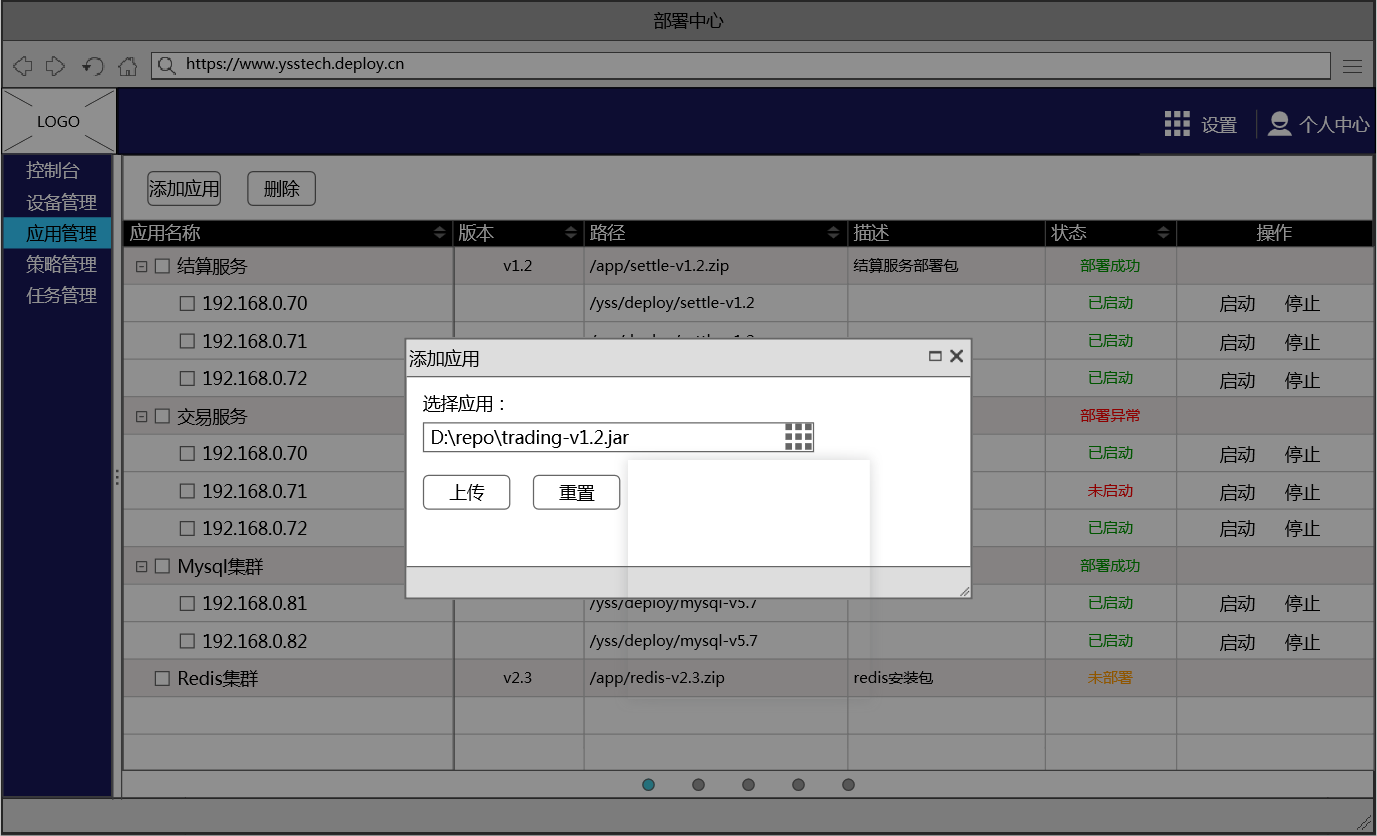


#### 应用管理

1. 列表页



1. 添加应用

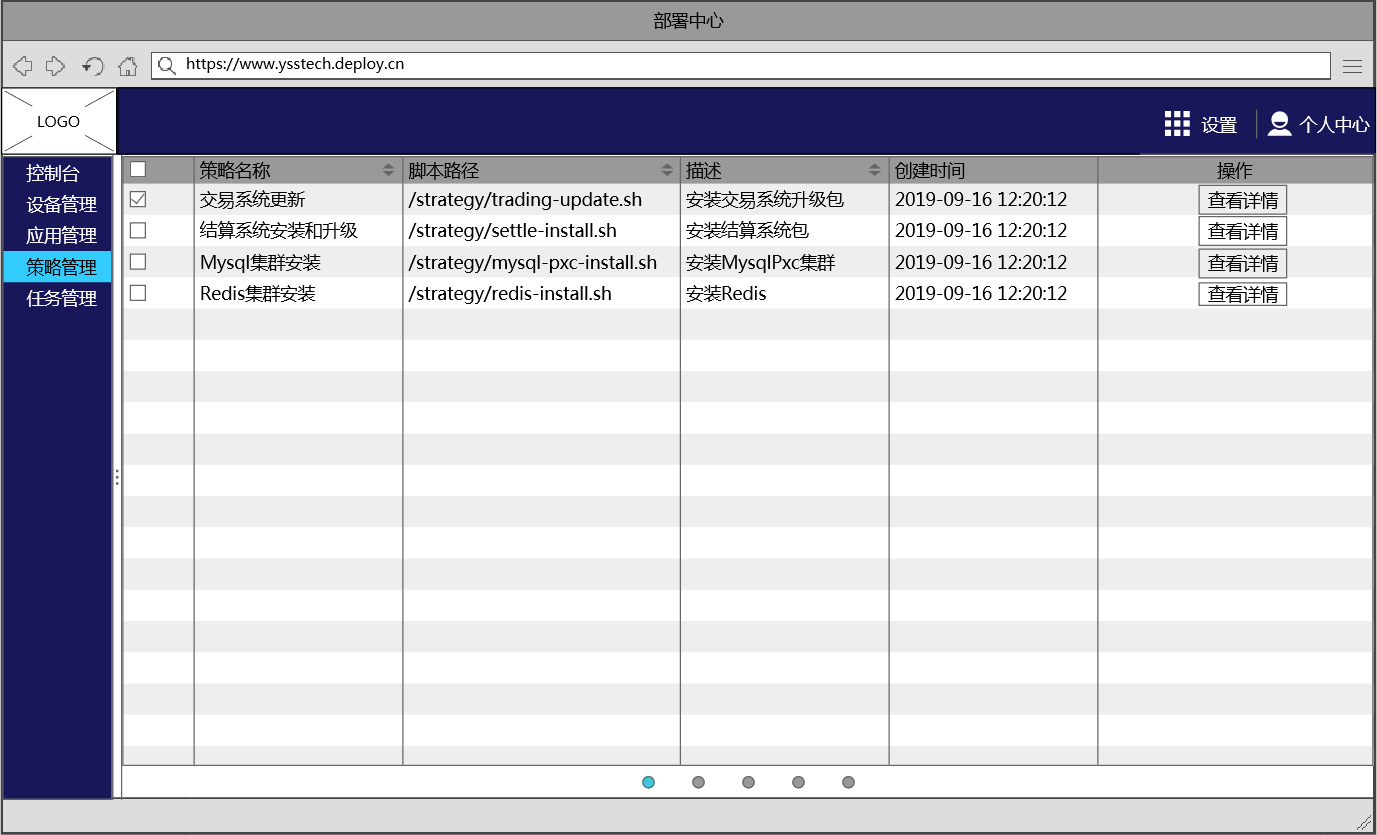


#### 策略管理

#### 策略列表

#### 部署任务

1. 列表页

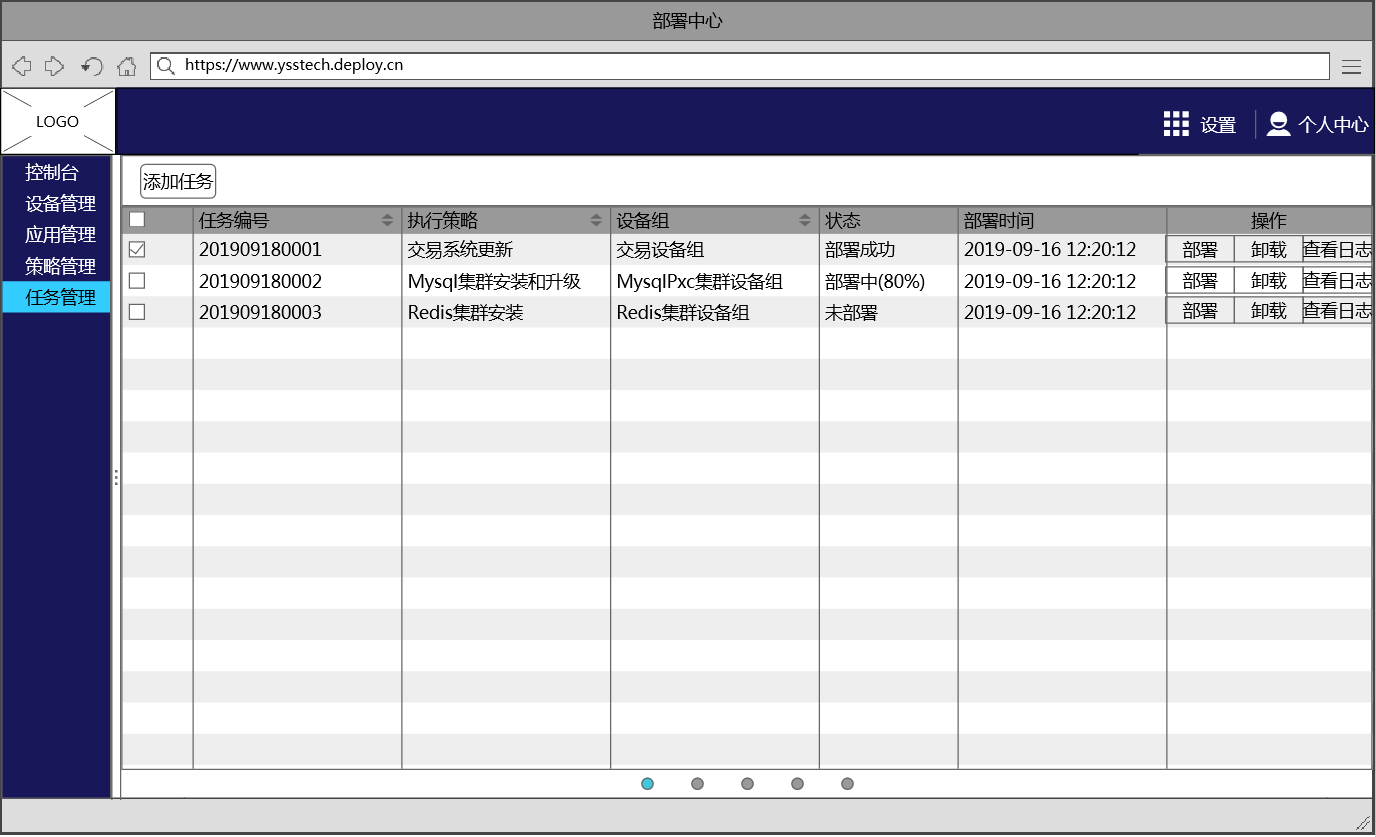


1. 策略详情

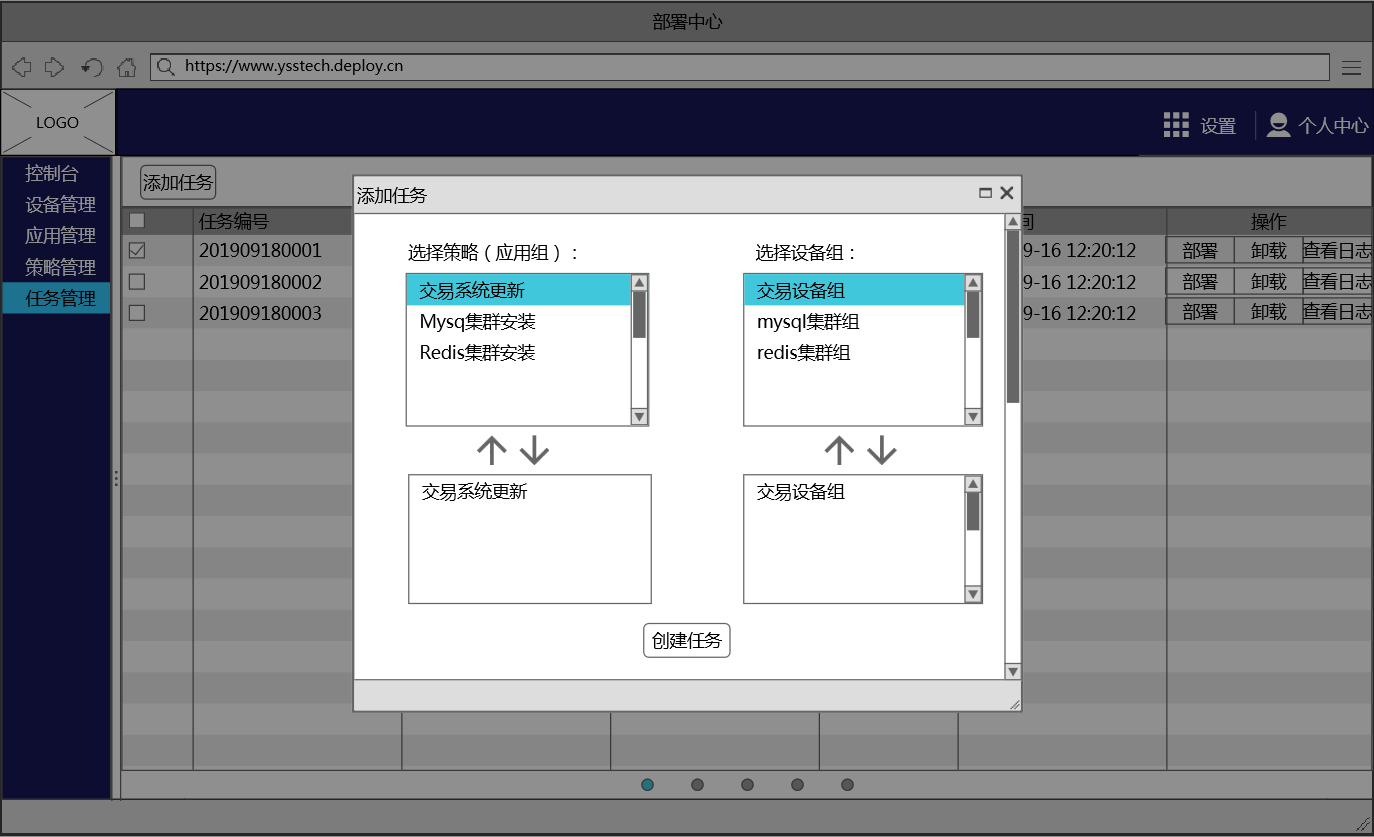


#### 任务管理

1. 列表页



1. 添加任务



1. 详情日志



## 概要设计

### 设备管理

#### 概要

1. 设备管理包括组管理和机器管理，分别在组和机器两张数据库表中存储。见下面数据库设计(组信息表、机器信息表)
2. 每个组对应多台机器，组只有‘删除’功能，删除组时只从组信息表中删除组ID关联的记录。
3. 机器有‘删除’、‘查看服务’、‘Web控制台’的功能。当物理机器IP修改或移除后，需要在设备管理界面删除该机器信息，需要更新表(组信息表、机器信息表)。
4. 添加机器时，设置好用户和端口后，尝试通过ssh免密测试远程机器是否能正常访问，并设置机器的连接状态。

#### 创建组

1. 管理设备的唯一入口时从创建组进入，创建组时可以选择‘机器列表’，从数据库中列出所有被管理的机器，如果不存在，则通过‘添加机器’来增加一台新的设备。
2. 添加机器时，填好机器逻辑名、IP地址、用户和端口后，点击‘连接’按钮尝试通过ssh免密访问远程机器，若正常连接可以返回远程机器的设备名。添加机器成功则在机器表中添加一条记录。
3. 创建组时选择好机器列表后，系统在组数据库表中为每一条机器信息和组信息添加一条记录到组信息表中，参见组信息表。

#### 机器服务列表

1. 机器与服务的绑定关系保存在节点服务列表中，参见节点服务信息表。
2. 服务提供‘启动’和‘停止’功能。点击功能按钮提交到调度中心执行远程脚本。

### 应用管理

#### 描述

应用管理的主要任务是管理软件仓库中的安装包(微服务安装包、系统软件安装包)，存放在软件仓库目录/repo/packages目录中，每个软件安装包都对应一个安装模块(脚本),存放在/repo/modules目录下。

1. 部署中心每次启动都扫描/repo/modules目录下的脚本，根据ModuleName、Path和版本与数据库模块信息表中的记录对比，存在不同记录就更新，否则新建一条记录。
2. 该模块脚本不提供修改、删除功能。每次升级后，把上一个版本的脚本文件按时间后缀备份到/repo/backup/modules目录。
3. 每个模块脚本提供必须的基本功能：

|  |  |
| --- | --- |
| 模块提供的功能 | 功能描述 |
| ModuleName | 获取模块名称 |
| Path | 获取相对repo目录下的相对路径 |
| Abspath | 获取Linux系统下绝对路径 |
| Help | 获取帮助 |
| Version | 获取按照包的版本 |
| Install | 执行安装包的全新安装功能 |
| Update | 执行安装包的升级功能 |
| Check | 执行安装的服务检测功能 |

#### 内置参数

每台机器都内置一些环境变量，模块脚本和策略脚本可以直接使用。

**## 远程配置**

# FTP服务器IP地址

CC\_FTP\_SERVER=192.168.0.70

# FTP服务的端口

CC\_FTP\_PORT=21

# FTP仓库相对根路径

CC\_FTP\_ROOT=/pub/repo

# FTP 客户端登陆的账号密码

CC\_FTP\_USER=ftp

CC\_FTP\_PSWD=ftp

**## 本地配置**

# 本地仓库路径主目录

CC\_HOME\_DIR=/root/repo

# 本地安装包存放目录

CC\_PACKAGES\_DIR=/root/repo/packages

# 本地模块脚本存放目录

CC\_MODULES\_DIR=/root/repo/modules

# 本地策略脚本存放目录

CC\_SCRIPTS\_DIR=/root/repo/scripts

# 本地日志存放目录

CC\_LOGS\_DIR=/root/repo/logs

# 本地应用安装部署根目录

CC\_DEPOLY\_DIR=/root/deploy

#### 系统升级

1. 管理员可以通过文件选择器上传固定格式的ZIP包，部署中心接收后把他解压到FTP服务器的repo目录下，执行覆盖之前备份一下modules目录下的脚本到repo/backup/modules，并增加后缀YYYYMMddHHmmss。解压时要判断ZIP包中的modules目录下的文件名，先备份后解压。
2. 升级包解压之后，部署中心重新扫描/repo/modules下的ModuleName、Path和Version，与数据库模块表的记录对应，做插入或更新操作。

### 策略管理

#### 描述

策略管理的主要功能是管理策略脚本，每个策略脚本包含了对多台设备，多个模块的一组操作。策略脚本存放在/repo/scripts目录下，每次升级后主动扫描该目录，更新策略信息表。

#### 内置参数

除与模块内置参数保持一致外，另外通过命令行传递基本参数：

GID： 组IP地址列表串，如：192.168.0.70,192.168.0.71,192.168.0.72

INSTALL\_DIR： 基于部署根目录下的子目录做为安装目录

BACKUP\_DIR: 基于备份根目录下的子目录做备份目录

LOG\_DIR：基于日志根目录下的相对路径做备份日志

例如：

./java\_ms\_trade\_install.sh install \

-GID 192.1680.72,192.168.0.71,192.168.0.72 \

-INSTALL\_DIR java\_ms/tradesys \

-BACKUP\_DIR java\_ms/tradesys \

-LOG\_DIR java\_ms/tradesys

### 计划任务

#### 描述

创建任务主要是绑定组与策略的关系，策略与组是1对N的关系，绑定关系保存在计划任务信息表中。任务有‘部署’和‘卸载’功能。

部署：启动部署时，根据组内IP地址执行对应的策略的Install功能，执行完成后检查返回结果，0为成功，其他为失败。

卸载：点击卸载按钮，根据组内IP地址执行对应的策略的Uninstall功能，执行完后检查返回结果，0为成功，其他为失败。

#### 添加服务

当创建计划任务后，当对一项任务启动‘部署’功能时，部署中心扫描策略脚本里所包含的模块脚本名，同时把模块对应的服务添加到节点服务信息表中，避免管理员需要手动添加要监控服务。

当启动‘卸载’功能时，同样扫描策略脚本中所包含的模块脚本名，同时把对应的监控服务从节点服务信息表中删除。

## 接口定义

### 组管理接口

#### 外部接口

/depolyCenter/group

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口 | 请求参数 | 响应结构 | 格式 | 接口描述 |
| /group/list | gid - 组ID | {  Code:0,  Errmsg:””,  Grouplist: [{  gid:”10”,  group\_name:”pxc集群组”,  devlist::[{  host:”192.168.0.70”,  devname::”dev-70”,  os\_type:”centos7.0”,  desc:”pxc主节点”,  status:1},  {  host::”192.168.0.71”,  devname::”dev-71”,  os\_type:”centos7.0”,  desc:”pxc从节点”,  status:2  }]  }] | JSON | 返回所有组和设备列表信息,gid为空表示返回全部。  Status：  0-未连接   1. 正常连接 2. 连接失败 |
| /group/del | gid - 组ID | {  “code”:0,  “errmsg”:”success”  } |  | 删除组 |
| /group/create | name - 组名  desc - 描述  devlist-机器列表 |  |  | 创建新组 |

### 设备管理接口

#### 外部接口

/depolyCenter/dev

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口 | 请求参数 | 响应结构 | 格式 | 接口描述 |
| /list | 无 |  |  | 列出所有机器列表 |
| /del | id - 机器id |  |  | 删除单个机器节点 |
| /create | Name - 机器名  Ip - IP地址  User - 用户名  Port - 端口 | Code:0,  Errmsg:”连接正常” |  | 添加机器 |
| /service/deviceList | Id - 机器id | Code:0,  Errmsg:”成功” |  | 列出机器对应的服务列表 |
| /service/listStatus | 模块（应用）id  设备列表-dlist |  |  | 按应用取设备上服务启动状态 |

#### 内部接口

/depolyCenter/dev/internal

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口 | 请求参数 | 响应结构 | 格式 | 接口描述 |
| /connect | Ip - IP地址  User - 用户名  Port - ssh端口 | Code:0,  Errmsg:”连接正常”  dev\_name: 设备名 |  | 连接机器,返回成功或失败 |
| /service/action | Id - 机器id  Op - 操作 | Code:0,  Errmsg:”操作成功” |  | 操作单个机器，返回成功或失败  Op：   1. 启动 2. 停止 |

### 应用管理接口

#### 外部接口

/depolyCenter/modules

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口 | 请求参数 | 响应结构 | 格式 | 接口描述 |
| /list |  | {  Code:0,  Errmsg:””,  ModuleList: [{  id:”1”,  name:”java微服务-清算系统”,  Path:”/modules/settlement.sh”,  Version:”v1.2”  },{  id:”2”,  name:”java微服务-结算系统”,  Path:”/modules/trade.sh”,  Version:”v5.0.2”  },  ]  } | JSON | 返回所有应用列表 |
| /upgrade | FilePath | {  “code”:0,  “errmsg”:”success”  } |  | 上传升级包 |
| /del | moduleId |  |  | 删除应用包 |

#### 内部接口

/depolyCenter/internal

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口 | 请求参数 | 响应结构 | 格式 | 接口描述 |
| /scan |  | {  “code”:0,  “errmsg”:”success”  } | JSON | 扫描应用，应用存放在/repo/modules下的所有模块脚本，指定ModuleName、Path和Version，与数据库模块表的记录对应，做插入或更新操作。  扫描策略，策略存放在/repo/scripts下所有脚本，指定StrategyName、Path、Version与数据库策略信息表的记录对应，做插入和更新操作。 |
|  |  |  |  |  |

### 策略管理接口

/depolyCenter/strategy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口 | 请求参数 | 响应结构 | 格式 | 接口描述 |
| /list |  | {  Code:0,  Errmsg:””,  StrategyList: [{  id:”1”,  name:”java微服务-清算系统-安装”,  Path:”/scripts/settlement\_install.sh”,  Version:”v1.2”  },{  id:”2”,  name:”java微服务-结算系统-安装”,  Path:”/scripts/trade\_install.sh”,  Version:”v5.0.2”  },  ]  } | JSON | 返回所有应用列表 |

### 部署任务接口

#### 外部接口

/depolyCenter/schdule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口 | 请求参数 | 响应结构 | 格式 | 接口描述 |
| /list |  | {  Code:0,  Errmsg:””,  SchduleList: [{  id:”1”,  StrategyID:”1”,  StrategyName:”交易系统安装”,  Status:1,  Gid:”10”,  groupName:”交易系统组”,  devlist::[{  host:”192.168.0.70”,  devname:”dev-70”,  os\_type:”centos7.0”,  desc:”pxc主节点”,  status:1 },{  host:”192.168.0.71”,  devname::”dev-71”,  os\_type:”centos7.0”,  desc:”pxc从节点”,  status:2  }] | JSON | 返回所有部署任务列表  Status:  1 - 已完成  2 - 执行中  3 - 未执行 |
| /create | GroupIds - 组id数组  StrategyId - 策略id | {  “code”:0,  “errmsg”:”success”  } |  | 创建部署任务，需要组+策略 |

#### 内部接口

/depolyCenter/schdule/internal

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口 | 请求参数 | 响应结构 | 格式 | 接口描述 |
| /install | Id - 任务id | {  “code”:0,  “errmsg”:”success”  } |  | 执行任务部署 |
| /uninstall | Id - 任务id | {  “code”:0,  “errmsg”:”success”  } |  | 执行卸载部署 |
| /viewlog | Id - 任务id | {  “code”:0,  “errmsg”:”success”  } |  | 查看部署任务日志 |
| /scanModule | ID - 入伍id | {  “code”:0,  “errmsg”:”success”  } |  | 扫描任务对应的策略，把策略中的服务添加到组对应的机器服务信息表中。 |

## 数据库结构

### 机器信息表

表T\_device

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 类型 | 是否NULL | 主键/索引 | 备注 |
| ID | INT(11) | NOT NULL | 主键 | 自增ID |
| NAME | VARCHAR(32) | NOT NULL |  |  |
| DEV\_NAME | VARCHAR(32) |  |  |  |
| IP\_ADDR | VARCHAR(32) | NOT NULL |  | IPV4 ADDRESS |
| USER | VARCHAR(32) | NOT NULL |  | SSH/SCP LOGIN USER |
| PORT | INT(4) | NOT NULL |  | SSH/SCP PORT |
| OS\_TYPE | VARCHAR(32) |  |  | 操作系统类型 |
| STATUS | INT(4) |  |  | 0-未连接   1. 连接成功 2. 连接失败 |
|  |  |  |  |  |

### 组信息表(包含机器ID)

表T\_group

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 类型 | 是否NULL | 主键/索引 | 备注 |
| ID | INT(11) | NOT NULL | 主键 | 自增ID |
| GID | INT(11) | NOT NULL | 允许重复 | 组ID，GID需要程序自增 |
| NAME | VARCHAR(32) | NOT NULL |  | 组名，多设备有冗余 |
| DEVICE\_ID | INT(11) | NOT NULL |  | 组包含的设备ID |
|  |  |  |  |  |

### 节点服务信息表

表T\_device\_services

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 类型 | 是否NULL | 主键/索引 | 备注 |
| ID | INT(11) | NOT NULL | 主键 | 自增ID |
| DEVICE\_ID | INT(11) | NOT NULL | 索引 | 设备ID |
| MODULE\_ID | INT(11) | NOT NULL | 索引 | 模块ID |
|  |  |  |  |  |
|  |  |  |  |  |

### 模块/策略信息表

表T\_module\_strategy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 类型 | 是否NULL | 主键/索引 | 备注 |
| ID | INT(11) | NOT NULL | 主键 | 自增ID |
| TYPE | INT(1) | NOT NULL | 索引 | 模块/策略类型区分   1. 模块 2. 策略 |
| NAME | VARCHAR(32) | NOT NULL | 索引 | 模块逻辑名 |
| PATH | VARCHAR(128) | NOT NULL |  | 模块相对路径 |
| VERSION | VARCHAR(32) | NOT NULL |  | 模块版本 |
| HELP | VARCHAR(255) | NULL |  | 模块帮助 |
| DESCRIPTION | VARCHAR(255) | NULL |  | 模块描述 |

### 计划任务信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 类型 | 是否NULL | 主键/索引 | 备注 |
| ID | INT(11) | NOT NULL | 主键 | 自增ID |
| STRATEGY\_ID | INT(11) | NOT NULL | 索引 | 策略ID |
| GROUP\_ID | INT(11) | NOT NULL | 索引 | 组ID |
| STATUS | INT(4) | NOT NULL |  | 任务状态   1. 未开始 2. 执行中 3. 已完成 |
| LAST\_TIME | CHAR(14) | NOT NULL |  | 最近执行时间 |
| CREATE\_TIME | CHAR(14) | NOT NULL |  | 任务创建时间 |
|  |  |  |  |  |

### 管理员操作日志表

表T\_admin\_logs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 类型 | 是否NULL | 主键/索引 | 备注 |
| ID | INT(11) | NOT NULL | 主键 | 自增ID |
| USER | VARCHAR(32) | NOT NULL |  | 管理员 |
| ACTION | INT(4) | NOT NULL |  | 操作动作 |
| CREATE\_TIME | CHAR(14) | NOT NULL |  | 创建时间 |
| MARK | VARCHAR(255) | NULL |  | 备注 |

## 更新记录

|  |  |  |  |
| --- | --- | --- | --- |
| 作者 | 版本 | 内容 | 时间 |
| 陈吉 | V1.0 | 初始 | 2019/09/18 |
| 贺汕森 | V1.1 | 新增交互原型设计 | 2019/09/18 |