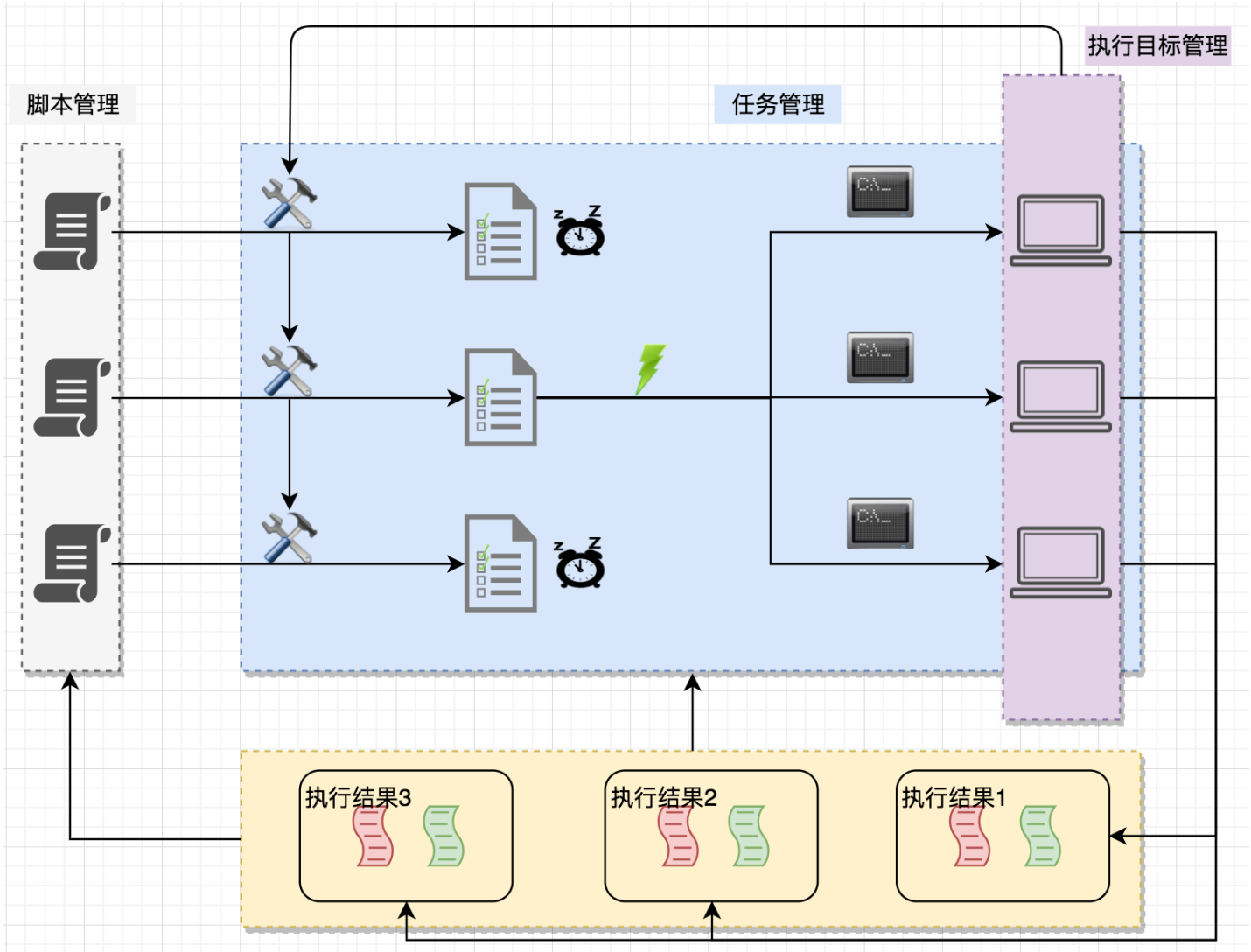


功能设计

三个功能：脚本管理、任务管理、执行目标管理

执行结果是执行目标执行完任务之后得到的输出，可用于脚本管理与任务管理的相关查询执行结果功能



脚本管理

管理权限范围内所有人员创建过的所有脚本，以列表的形式展现出来方便管理与执行

根据页面分类

- 列表页（批量查询、列表搜索、单个脚本的执行与删除）
- 创建脚本页（单个脚本的创建）
- 脚本详情页（单个脚本的删改查与执行、查询执行历史）

根据功能分类

- 脚本列表批量分页查询（名称、语言、创建人、创建时间）
- 脚本列表搜索（名称）
- 单个脚本的增删改查（名称、语言、创建人、创建时间、**执行历史**(通过执行历史查看功能查询，会根据单个脚本ID查询出该脚本所有执行过的任务日志))
- **脚本执行**(通过任务管理中的创建任务执行，会弹出执行脚本的弹窗创建可执行的任务)

权限

系统管理员、租户管理员

任务管理

脚本是静态的可执行文件，**任务**是脚本在系统中与执行目标绑定的动态呈现。

根据执行目标、执行方式、脚本来创建任务；管理权限范围内所有人员创建过的所有任务，以列表的形式展现出来方便管理与执行

根据页面分类

- 列表页（批量查询、单个任务的停止启动删除）
- 创建任务页（单个任务的创建）
- 任务详情页（单个任务的删改查与执行、查询任务日志执行的历史）
- 单个任务执行日志列表页（单个任务查询与该任务所有执行目标的查询）

根据功能分类

- 任务列表批量分页查询（名称、类型、执行状态、脚本信息、下次执行时间、创建人、创建时间）
- 任务列表搜索（名称）
- 单个任务的增删改查、启动停止
 - （
 - 名称、类型、执行周期、选择脚本、
 - **选择主机**（调用执行目标管理功能中的查询执行目标列表）
 - **执行目标**（查询执行目标列表，需要支持名称搜索）与**执行历史**（查询该任务执行过的日志列表，需要支持名称搜索）
 - ）
- 任务执行日志列表查询（任务日志ID、任务类型、脚本名称、执行状态、开始时间、结束时间、耗时、执行人）
- 单个任务执行日志详情查询（主机ID名称、云资源池、区域、执行状态、标准输出、错误输出）

脚本执行功能

基础执行功能：使用jsch实现，能分别获得stderr和stdout的输出

定时任务功能：使用quartz实现，数据库方面需要创建一些quartz的特定表，同时维护自己的任务管理表

脚本执行失败通知

调用Manager中的消息中心 `createNotice` 与 `SendNotices` API来发送失败通知

执行目标管理

配置CMP后台与执行目标的连接并管理连接状态，提供目标给创建任务功能

根据页面分类

- 任务执行目标主机列表页（批量查询、根据名称查询、单个执行目标的删除）
- 创建执行目标页（单个执行目标的创建）
- 执行目标页详情（单个执行目标的删改查）

根据功能分类

- 执行目标主机列表查询（主机ID、名称、状态、SSH连接信息）
- 执行目标列表搜索（名称）
- 执行目标的增删改查（SSH连接信息）
- 代理节点列表查询（ID）
- 将执行目标与代理节点连接起来，此时不需要将公钥放到执行目标中，由用户去做

代理节点管理

同上

连接到各种主机的方式

一、能通过公网IP或使用具有公网IP的VPC转发访问执行目标

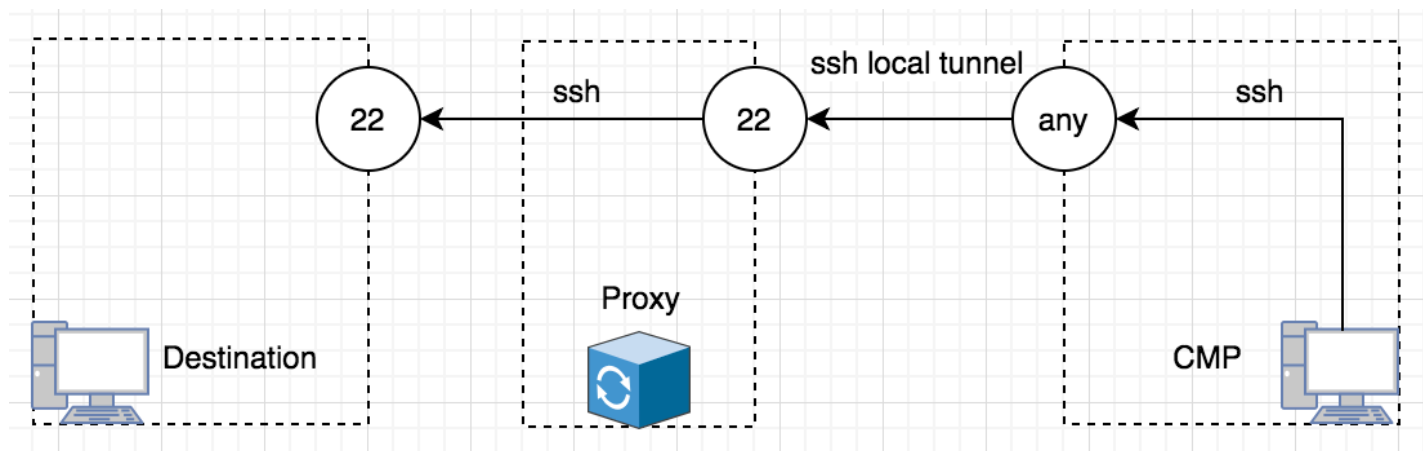
用户必须自己解决的问题：将执行目标绑定公网IP或对VPC做端口转发

需要填到创建执行节点页面的字段为：执行目标/VPC(username,ip,port)

二、能通过代理节点访问执行目标

用户必须自己解决的问题：

1. 部署能访问CMP后台与内网执行目标的代理节点
2. 在代理节点上通过免身份验证的方式能SSH直接登录到执行目标



整体流程：

1. CMP后台在本地创建与代理节点的SSH隧道，这条隧道会立即在本地使用随机的一个端口与代理节点的SSH端口建立SSH隧道
2. CMP后台使用SSH连接该本地端口，请求将转发给代理节点，这样就能以代理节点的身份去执行命令，访问到执行目标

CMP会在本地执行的命令如下：

1. 建立SSH隧道的命令：

```
ssh -fNL 本地端口:执行目标IP:执行目标的SSH端口 -p 代理节点端口 代理节点用户@代理节点IP
```

2. 执行命令：

```
ssh -p CMP本地端口 代理节点用户@127.0.0.1 "脚本内容"
```

需要填到创建代理节点页面的字段为：代理节点(username,ip,port)，执行目标(username,ip,port)

三、两种方式都需要的最后一步：将公钥放到执行目标里

在用户填写完需要的字段之后，还需要让用户填写SSH密码来把CMP的public key放到执行目标上，或者告知用户在执行目标上执行如下命令：

```
(省略具体命令) 【PUBLIC KEY】 >> $auth_keyfile;
```

注意：public key与private key需要提前生成好并且放到部署CMP的主机的~/ssh文件夹里与前后端的代码里

其他问题

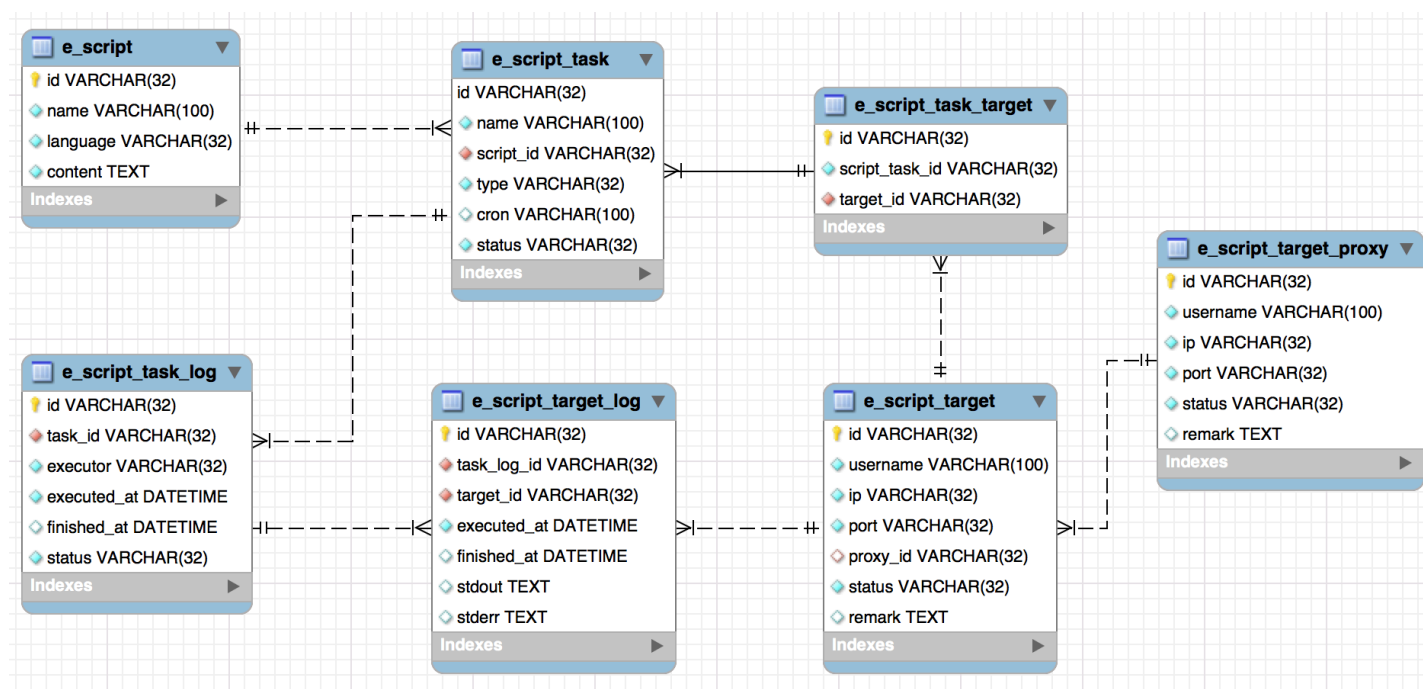
SSH连接超时问题：

```
sessionProxy.setConfig("ServerAliveInterval", "60");
```

支持脚本传参：定义特殊环境变量，并将参数写到这里，脚本在执行的时候会将改变量放到脚本后面执行，类似这种

```
bash -c '#!/bin/bash \necho $0' yourparameter
```

数据库设计



对于定时任务的表是采用quartz官方提供的建表SQL，是quartz自己维护任务状态的结构，对于CMP的脚本定时任务来说，则需要将自己的定时任务放到e_script_task中

e_script

脚本信息表，存储脚本的各种信息

外键：无

e_script_task

脚本任务表，存储根据脚本、执行目标、执行方式生成的任务

外键：

1. 与e_script_info的关系，多对一

e_script_task_target

任务-执行目标表，存储任务与执行目标的一对多关系，一个任务可对应多个执行实例，通过任务ID拿到所有连接的信息，从而拿到实例的信息；如果没有绑定实例，则拿到业务上需要的某些字段，现阶段不考虑。

外键：

1. 与e_script_task的关系，多对一，一个任务会有多个连接，不同的任务可能会公用相同的连接
2. 与e_script_target的关系，多对一

e_script_task_log

脚本执行日志表，记录每个脚本在每个执行目标中的执行结果，每条记录代表某个任务执行一次

外键：

1. 与e_script_task的关系，多对一

e_script_target_log

每个目标上执行的记录，多个记录对应同一个执行日志

外键：

1. 与e_script_task_log的关系，多对一
2. 与e_script_target的关系，一对一

e_script_target

脚本执行目标主机表，记录已经建立过SSH连接的主机连接信息

外键：

1. 与e_script_target_proxy的关系，多对一，多个主机可能会使用同一个代理

e_script_target_proxy

脚本执行目标代理主机表，记录已经建立过SSH连接的代理主机连接信息