

# Solidwork 上云服务

姚迪熙	李洪旭	彭乐天
F1803303	F1803303	F1803303
518021910367	518021910303	518021910017
计算机科学技术系	计算机科学技术系	计算机科学技术系
<i>Jimmyyao18@sjtu.edu.cn</i>	<i>lhx0414@sjtu.edu.cn</i>	<i>zxc-00@sjtu.edu.cn</i>

## 目录

<b>1</b>	<b>介绍</b>	<b>2</b>
<b>2</b>	<b>技术支持</b>	<b>3</b>
2.1	应用服务 solidworks . . . . .	3
2.2	云平台服务 openstack . . . . .	3
2.3	网络服务 Apache . . . . .	4
<b>3</b>	<b>项目制作</b>	<b>5</b>
3.1	制作镜像 . . . . .	5
3.2	openstack API 脚本调用 . . . . .	6
3.3	网站设计和逻辑/后端 . . . . .	8
<b>4</b>	<b>成果展示</b>	<b>9</b>
<b>5</b>	<b>难点及解决办法</b>	<b>10</b>
5.1	Apache 服务 . . . . .	10
5.2	Openstack API 调用 . . . . .	11
<b>6</b>	<b>拓展调查</b>	<b>12</b>
6.1	高并发服务 . . . . .	12
6.2	openstack 与个性化服务 . . . . .	13
<b>7</b>	<b>总结感想</b>	<b>13</b>

## 1 介绍

随着计算机技术的发展, 3D 打印, 机械制造走进了大家的生活。因而 3D 建模也随之产生, 令我们高兴的是随着研究人员和工程师的开发, 达索公司下属子公司所开发的 solidwork 软件也成为了 3D 建模的首选途径。因为其强大的功能广受各类从业人员, 教育人员和学生的青睐。因此随着工业的发展人们对 3D 建模和使用 solidwork 3D 建模软件的需求也不断增长。

但是 solidwork 软件存在着以下**痛点**需要解决。

1. 大规模的组件以及授权认证导致安装和卸载比较复杂。
2. 软件对硬件有一定要求, 许多人电脑无法满足配置。
  - (a) 运行软件需要至少 50HB 的磁盘空间。
  - (b) 软件对 CPU 和内存有一定要求。

我们进行了分析希望项目能够达到的**目标**是

1. 借助学校云平台与云服务提供 solidwork 软件云。
2. 通过前端和用户交互, 获得请求, 后端与云平台交互获得服务的双向结构。
3. 用户对我们的系统提出请求, 我们能够快速, 并通过简单的方法将服务返回给用户。
4. 用户通过请求获得有云中心提供的一台远程虚拟机的连接权限, 连入后, 如同在用一台远程的有 solidwork 软件的虚拟机。
5. 同学可以利用服务快速体验 solidwork。

在项目完成建设以后能够带来充分的**优势**

1. 从**效益**上, 可以满足大量需要使用 solidwork 软件但是没有足够物理环境人员的需求, 特别的针对我们学校的局部环境来讲, 所有的工科平台的学生, 设计学院的同学以及机械与动力工程学院的同学都需要使用到 solidwork。
2. 从**成本**上, 不光是 solidwork 软件, 随着计算机和互联网技术的发展, 我们在教学和日常工业中, 也可以摆脱物理机安装软件给同学们使用的情况。减少资金和资源的开销, 设想一下, 将现在学生创新中心里面装有应用程序的物理机转换为云服务会有很大的资源节省。

之后的篇幅分别从技术支持, 和具体实现细节展开。还有相应的项目结果展示, 过程中遇到的问题, 以及进一步的思考感谢。最后给出我们的总结。

## 2 技术支持

### 2.1 应用服务 solidworks

SolidWorks 是达索系统下的子公司，提供的软件和服务以 3D CAD 为主。功能强大，组建繁多，在工业和机械制造领域使用诸多。主要功能有 3D 建模，生成工程图等，我们的项目主要提供该软件的云应用服务。

硬件和系统：根据 solidworks 官网 [1] 的一些信息，对软件的硬件需求有一个基本了解。软件基本的硬件要求不是很高。对系统要求是 windows10 或者 7 即可。

进阶配置：接下来研究了下进阶的信息，Solidworks 软件不设有集群功能，故本项目

Hardware and Hypervisors	
Hardware	
Processor	3.3 GHz or higher
RAM	16 GB or more PDM Contributor/Viewer or Electrical Schematic: 8 GB or more
Graphics	Certified cards and drivers
Drives	SSD drives recommended for optimal performance

图 1: solidworks 基本要求

暂不提供集群云服务。Solidworks 软件主要界面在于 3D 建模，软件主要通过多线程来处理指令，在一些特殊的时候例如渲染，导出模型等会使用多核计算。但本身大部分操作如建模不需要，因此不用特别高的核心数目。

激活和授权：对于 *SOLIDWORKS 2019 SP05*（之后简称 SW19pro）版本，学校提供校园网授权版本。只要用户接入校园网，即可直接使用 solidworks 无须其他激活和认证操作。

### 2.2 云平台服务 openstack

OpenStack [3] 是一个云操作系统，通过数据中心可控制大型的计算、存储、网络等资源池。所有的管理通过前端界面管理员就可以完成，同样也可以通过 web 接口让最终用户部署资源。

OpenStack 开源项目由社区维护，包括 OpenStack 计算（代号为 Nova），OpenStack 对象存储（代号为 Swift），并 OpenStack 镜像服务（代号 Glance）的集合。openstack 的详细架构见图 2

在 OpenStack 云中，使用 OpenStack API 来启动服务器云主机，创建镜像、给实例和

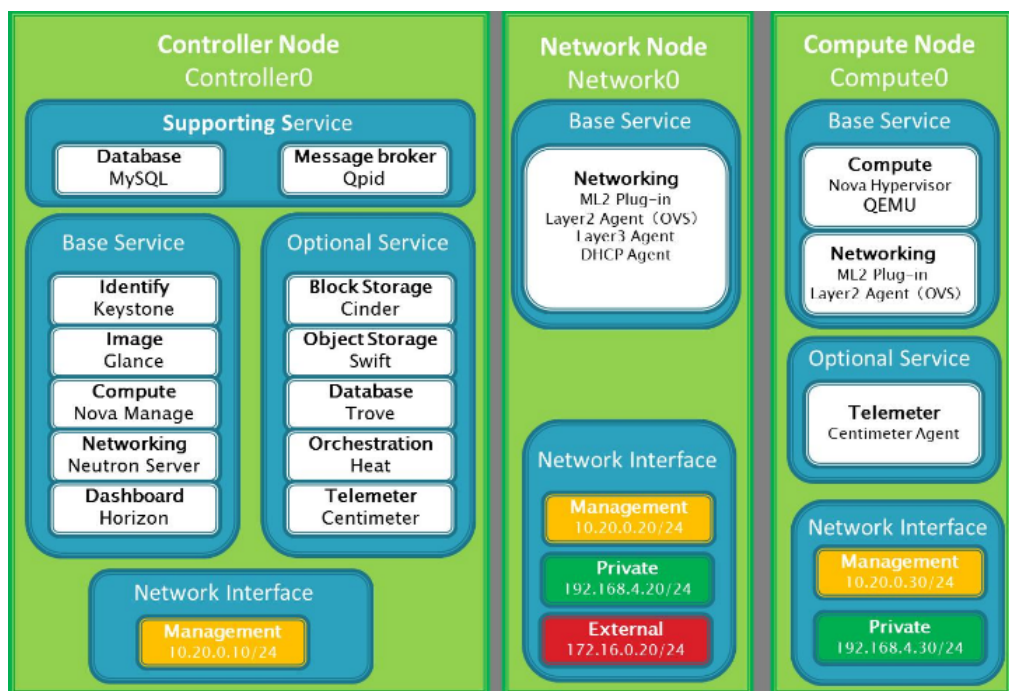


图 2: openstack 详细架构图

镜像分配元数据、创建存储容器和对象以及完成其他相关的动作。根据 openstack API 官方文档 [4]，有四种方法发送 openstack API 请求

1. cURL
2. OpenStack 命令行客户端
3. REST 客户端
4. OpenStack Python 软件开发工具包 (SDK)

学校的云平台<sup>1</sup>就是一个基于 openstack 服务开发的平台，因此我们的目标也就是能够与 Openstack（云平台）进行交互，获取资源。

在我们的项目中，我们主要使用了 openstacksdk 与 openstack 的 python API 交互，在网站后端实现交大云平台（jcloud）上云资源的管理。

## 2.3 网络服务 Apache

Apache 是世界使用排名第一的 Web 服务器软件，这是一个开源 Web 服务器，并且拥有良好的社区和跨平台能力。Apache 处理请求是阻塞型的，是同步多进程型，并且一个链

<sup>1</sup>home.jcloud.sjtu.edu.cn

接请求对应一个进程，比较稳定 bug 更少，并且处理动态文件的能力更强。我们选择其提供我们的网络服务，我们开启了一台云主机（4 核 32G 内存），使用的系统为 Ubuntu1804-64 位，在上面部署了阿帕奇服务框架 (httpd-2.4.46)，监听端口默认 80 端口，并且配置了 php 8.0 服务以保证 Apache 可以处理静态页面（前端 html）和动态请求（后端 Php）。

通过 Apache，用户可以使用浏览器获得我们提供的云服务，在后端 Apache 能够处理请求，和本地的脚本进行交互，完成与 openstack 的交互，形成服务链。

### 3 项目制作

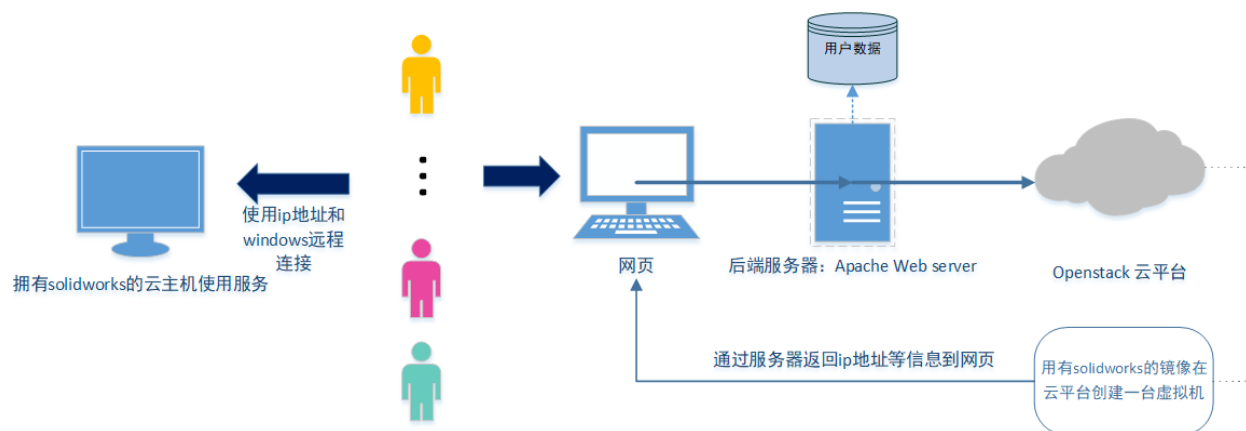


图 3: 项目流程图

我们项目的流程图如 3 所示，对用户来说用户只需要在网页上创建开启虚拟机获得 ip 地址，然后使用 windows 远程登陆就可以使用云服务。而我们的前端在网页上呈现，html 将请求发送到后端，后端 php 处理请求以后调用相应的脚本，这些都在服务器上完成。服务器同时拥有一个数据库存储用户信息。服务器调用 python 脚本后，通过 API 和云平台进行交互开辟虚拟机再将虚拟机的内容发送到网页前端。

#### 3.1 制作镜像

为了将 solidworks 上云，网页服务器后端通过脚本中 openstack API 调用学校云平台 (openstack) 服务的时候创建一个虚拟机，使用已经压制好的镜像。该镜像需要安装好 solidworks 并且完成相应的授权。

首先在云平台账户下面创建一台 windows10 的虚拟机，配置选择增强款-通用型 4 核/16 GB。云硬盘选择 100G，软件安装完成后加上系统约占用 60G 左右的空间。打开云主机以后登陆上海交通大学软件授权中心 [2] 下载 SW19pro。接下来绑定浮动 IP 保证主机可以联通校园网以后，安装网站指导进行解压缩和安装就可以。

在安装之后制作相应的镜像，找到云主机，在选项中选择“创建快照”。然后进入存储

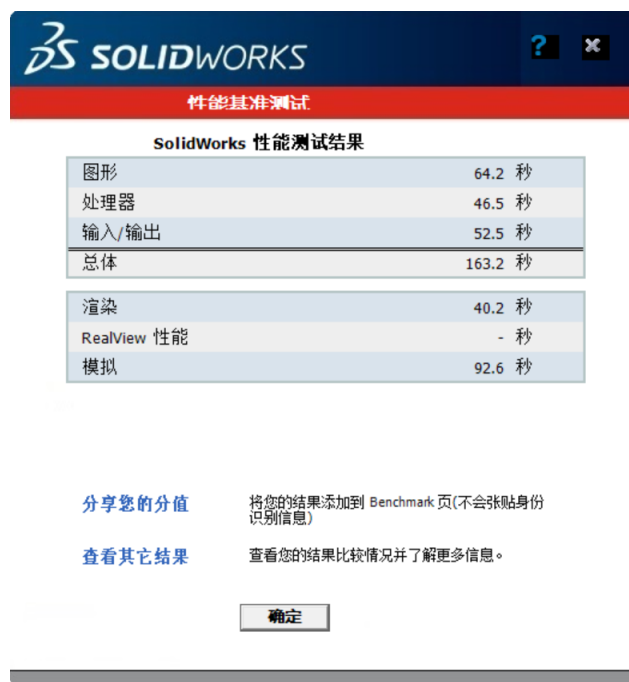


图 4: solidworks 性能测试结果

→ 云主机快照 → 创建镜像即可。至此完成镜像的创建，由于权限原因该镜像目前还是私有镜像。

最后测试该镜像，此时使用云平台新建云主机，在镜像中选择应用版 windows 镜像。可以找到已经创建好的镜像 *solidworks19pro*，选择该镜像并且设定一样配置以后创建，绑定浮动 ip。然后登陆云主机以后测试是否成功。使用 solidworks 软件自带的测试基准工具，测试是否交大授权的功能都可正常使用，以及性能表现。测试结果如图 4所示。

所有的功能都是可以使用的，测试结果达标，虽然没有达到非常高的标准，但是作为云服务平台，资源性能有限，如果想要更好的服务一方面需要更大的配置，用户收费会更高。另一方面，基于 KVM 的虚拟机和裸金属服务器，物理服务器性能也有相对差异。目前做到基本可以满足大家的日常学习使用需求。

### 3.2 openstack API 脚本调用

本项目用过 openstacksdk，实现了与 openstack API 的 python 绑定，通过执行 python 脚本来管理云平台资源。实现的步骤大致分为如下几步：

#### 1. 建立与 jcloud 云平台的连接

openstacksdk 提供了多种与云平台连接的方式，如通过密码参数直接连接、通过已认证的会话连接、通过配置文件连接等。本项目通过使用用户名密码参数实现会话的建立，通过会话实现连接，具体见图 5。

```

9  auth = v3.Password(auth_url='https://home.jcloud.sjtu.edu.cn:5000/v3',
10                      username='Jason',
11                      password='200414Lhx',
12                      project_name='jimyyao18_project',
13                      user_domain_name='Default',
14                      project_domain_name='Default'
15                      )
16
17  sess = session.Session(auth=auth)
18
19
20  conn = connection.Connection(
21      session = sess,
22      region_name='RegionOne',
23      identity_interface='public',
24  )

```

图 5: 连接

2. 调用 openstack API 实现需要的云主机的创建，并返回云主机的浮动 ip 地址  
云主机的创建需要几个必要的元素

- Name，即云主机的名字
- Image，即镜像文件
- Flavor，即云主机的 cpu 与内存配置
- Network，即云主机的网络配置

通过对 openstacksdk 文档 [3] [4] 的研究，我们可以调用 openstack 的 compute 类中的对应函数实现 image 列表与 flavor 列表的搜索，调用 openstack 的 network 类中的对应函数实现 network 列表的搜索，从中找到我们创建云主机需要的配置。在我们的项目中我们使用的配置见下表

Name	用户的账号名
Image	solidworks_19pro
Flavor	research.4C.16G
Network	ydx

表 1: 云主机配置

其中云主机被命名为用户注册时使用的用户名，镜像文件”solidworks\_19pro” 就是我们之前在 Section3.1中创建的镜像文件，“research.4C.16G” 是云平台上已有的 cpu 与

内存配置，“ydx”是我们创建的私有网络，地址为 192.168.1.0/24。

在创建好我们需要的云主机之后，为了让用户能够方便地通过 ip 地址远程连接云主机，我们需要给云主机设置一个浮动 ip 地址。因为 solidworks 在校园网中才可以验证，该浮动 ip 需要依赖于校园网创建。创建好浮动 ip 后，将浮动 ip 赋给之前创建好的云主机，这样就完成了基本的云主机的创建。

### 3. 实现云主机需要的附加功能在实现了原始功能的基础上，我们实现了部分附加的云主机功能：

- (a) 云主机的开机关机功能
- (b) 云主机的重启功能，包括硬重启与软重启
- (c) 云主机的删除操作
- (d) 云主机密码修改

因为云主机创建完成之后，云平台默认会设置一个随机的密码，所以我们需要调用密码修改操作来让用户设置其各自的密码，来实现云主机的登陆。同时，在后续的使用中也可以根据需要修改云主机的密码。

基本上官方文档中提供了我们这个项目所需要的所有的函数的参照，但是我們也需要通过调试选择最合适的方式，我们在 Python 脚本中写入这些调用服务的函数。具体实现的脚本代码附加在附录中。

## 3.3 网站设计和逻辑/后端

在完成了对远程虚拟机创建和控制的程序设计和函数封装工作后，为了使得我们设计的服务能够更加便捷地满足同学们的需求，我们为 solidworks 设计了一个专门的网站。通过访问网站同学们就能够随时随地请求我们的服务获取装有 solidworks 的虚拟机。

在本项目中我们实现的是一个轻量级的网站，包含了用户注册和登录界面以及控制虚拟机的共三个页面。我们为从零开始使用服务用户设想的使用流程如下：

首先用户在注册界面使用个人信息（在交大比如 Jaccount）注册本网站的服务账号，在完成注册后从登录界面登录并进入虚拟机控制界面，通过创建虚拟机按钮在网站的服务器端调用 php 脚本新建带有 solidworks 的虚拟机，网站在完成虚拟机创建后便将虚拟机的地址返回给用户供用户使用。用户使用完虚拟机后，可以选择关闭虚拟机以供以后使用，或者如果长期不再使用 solidworks 服务的话，也可以选择删除虚拟机。

网站在后端维护一个用户的数据库，将用户名作为键，存储用户的网站账号密码信息用于登录时核对，存储用户的虚拟机和密码用于作为执行脚本命令的参数，负责指定用户控制的虚拟机。当出现键冲突时，说明有新用户尝试注册与数据库已存的用户名相同的账号，数据库检查到错误并在前端用户交互处阻止该账号的创建。



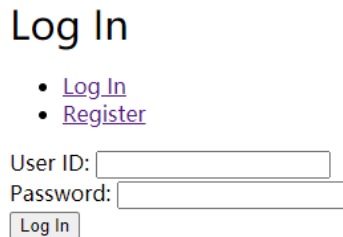
在网站的设计中，我们还提出了以下优化方案，用于保障网站的稳定性和用户的使用体验。

- 定期清理网站中长期未使用服务的账号占用的虚拟机，避免云平台资源的浪费。比如在虚拟机停止活动后 12 小时后将其自动关机，用户连续两周不使用云平台服务，便将该虚拟机清除，其工作文件备份存到一个文件仓库中，保存两个月。用户在此期间可以从网站中下载自己的工作文件，也可以创建一个新虚拟机导入这些文件，相当于立刻恢复自己的工程。
- 为用户提供在线的云服务，用户可以直接在网站上控制远程虚拟机上的 solidworks。一方面减少用户的操作复杂度，同时可以节约虚拟机的容量，提高网站的服务总量。缺点在于这样服务器与用户的通信量过大，可以考虑不直接使用服务器提供服务，转为其它通信媒介为用户提供实时服务。

## 4 成果展示

在网络服务器启动以后，确保用户连接了校园网或者挂载了交大 VPN。使用浏览器进入网址 10.119.5.66, 就可以看见网站的登陆页面（图 6）。

用户首先需要注册一个账号，注册的时候包含用户名重复检测，密码一致性检测，密



The image shows a web interface for user login. At the top, the text "Log In" is displayed in a large, bold, black font. Below this, there are two links: "Log In" and "Register", both in a smaller, blue, underlined font. Further down, there are two input fields: "User ID:" followed by a text box, and "Password:" followed by a text box. At the bottom of the form, there is a button labeled "Log In" in a small, black, rectangular box.

图 6: 用户登陆界面

码强度检测等功能。用户注册好帐号以后返回，选择登陆。输入用户名和密码以后后台会检测用户名存在与否，以及密码正确与否。在确认成功以后用户即可以登陆进入使用界面（图 7）。

在使用界面有以下的功能，首先用户如果需要使用功能需要通过”Get Solidwork “创建一台虚拟机，否则使用其他服务的时候都会提示没有创建虚拟机。在完成创建后网站会提示需要修改密码。通过”Change Password” 并在旁边输入新密码即可以修改虚拟机中的用户密码。其余的功能包括”Delete Soliwork” 会将现在用户创建的虚拟机删除。然后也可以选择打开，关闭和重启虚拟机等操作。

当用户需要使用 solidwork 服务的时候，首先可以选择”Show address” 了解自己虚拟机的 ip 地址。然后用户使用 windows 系统自带的远程桌面功能。然后输入该 ip 地址。然后

## Get Your Solidwork

Welcome to the world of Solidwork! Click "Get Solidwork" to start!

Show address

### Service

Get Solidwork

Delete Solidwork

### Control

Open Solidwork

Close Solidwork

Soft Reboot

Hard Reboot

### Password

Change Password(of your solidwork)

Length >= 6

Back

图 7: 用户使用陆界面

登陆时用户名输入"Administrator", 密码为自己刚才所修改的用户密码。就可以连入远程桌面, 进入以后已经安装好了 SW19pro 软件, 用户直接使用上面的 solidwork 就可以享受服务了。

这里有一个简单的关于项目操作的视频<https://jbox.sjtu.edu.cn/1/J5z6ku> 密码 *vpkk*。欢迎使用!

## 5 难点及解决办法

### 5.1 Apache 服务

在为后端搭建 Web Server 的时候我们遇到了一些问题。

1. 首先为了保证后端服务的正确运行, 需要为 Apache 配置 Php 服务, 否则 Apache 只能处理静态页面。在下载了 Php 以后并不会直接链接服务, 而需要自己手动修改 httpd.conf 文件在其中添加 `LoadModule php_module modules/libphp.so` 来加载 php 的动态编译库。网络上很多博客都有一些错误或者过时, 也是仔细看了 php 的官网后知道有这样一个动态库, 然后可以进行加载, 给 httpd 服务提供 php 解释器。
2. 另外遇到的一个问题是, 在本地调试网页代码可以成功运行, 但是在 Apache 服务器上就会不能成功调用 php 函数 `exec` 来执行 shell 命令。在看了 Apache 社区以后发现这和文件的权限设置有关系。Apache 在提供服务的时候是以 `apache:apache` 为用户和组身份进行执行。因此对于脚本我们使用 `chown` 和 `chmod` 指令修改 `read` 和可执行权限。

## 5.2 Openstack API 调用

因为在项目开始的时候我们对 openstack 是完全陌生的，所以在很多次尝试中走了很多弯路。我们遇到了很多问题，也解决了很多问题，但很多时候解决了问题并没有对我们的项目进展有任何帮助，反而让我们在错误的道路上越走越远。我们实时记录了一部分代码报错，如何实现的问题，但在完成项目之后，发现其中很大一部分问题与我们的最终的实现方法毫无关系。因此，我们只在这里列出一些与我们的实现紧密相关又让我们头疼很久的困难，尽管这只是我们遇到一系列困难中很小的一部分。

1. 在连接云平台的过程中，原本需要使用从网站上获取的 token 通过 api 的鉴权。但一是因为 jcloud 上 token 的有效期只有 6 个小时，定期更新不太方便，二是因为我们在 openstacksdk 的文档对于如何使用 token 通过 keystone 的验证并没有说明，我们不确定这个方法的是否可行。所以我们向助教申请重新创建了 jcloud 上的账号（不用 jaaccount 登陆），用于通过 keystone 的鉴权，来实现与 jcloud 的连接。
2. 在创建云主机时，如果不执行修改密码的操作，创建出的云主机随机生成一个密码（该处是与助教确认之后我们才了解），然而我们在创建云主机的函数中试图修改初始密码为固定的“123456”时却得到如下的报错（图 8）

```
http_status=http_status, request_id=request_id
openstack.exceptions.ConflictException: ConflictException: 409: Client Error for url: https://home.jcloud.sjtu.edu.cn:8774/v2.1/2f6d334cc33648e5b38b5ac19e018776/servers/0dbfef6e-675d-4d62-89be-533d07be50f1/action, Failed to set admin password on 0dbfef6e-675d-4d62-89be-533d07be50f1 because error setting admin password
```

图 8: 修改密码报错

我们暂时没有弄明白出现该报错的原因，我们解决的办法是放弃了在创建云主机的函数中直接修改密码的想法，用同样的函数实现了一个单独的修改密码的 python 脚本，通过对该脚本的调用可以实现密码的修改。

3. 在绑定浮动 ip 时，创建的浮动 ip 需要与一个外部网络关联。在可选的网络列表中只有“public”，“base”和我们创建的私有网络“ydx”（图 9）

```

List Networks:
Public Network
openstack.network.v2.network.Network(ipv6_address_scope=None, revision_number=16,
ef934, router:external=True, availability_zone_hints=[], availability_zones=['nova
070e47b397a8731909682b14, status=ACTIVE, subnets=['6327ec90-d198-4514-91b6-f9151d
-408c-94a8-66b68b6c6edd'], description=, tags=[], updated_at=2019-12-09T09:39:55Z
min_state_up=True, created_at=2018-11-06T06:53:12Z, mtu=1500, location=Munch({'cl
object': Munch({'id': '37bac654070e47b397a8731909682b14', 'name': None, 'domain_id
base
openstack.network.v2.network.Network(ipv6_address_scope=None, revision_number=3,
fd9c, router:external=False, availability_zone_hints=[], availability_zones=['nova
70e47b397a8731909682b14, status=ACTIVE, subnets=['e288fdb6-3a7e-4572-a3fc-d8d1e8d
, name=base, qos_policy_id=None, admin_state_up=True, created_at=2018-11-18T15:23
me': 'RegionOne', 'zone': None, 'project': Munch({'id': '37bac654070e47b397a87319
}})))
ydx
openstack.network.v2.network.Network(ipv6_address_scope=None, revision_number=3,
75c8, router:external=False, availability_zone_hints=[], availability_zones=['nova
c33648e5b38b5ac19e018776, status=ACTIVE, subnets=['4bfa3606-4634-4c67-a49c-b3b2a3
Z, name=ydx, qos_policy_id=None, admin_state_up=True, created_at=2020-10-04T12:39
me': 'RegionOne', 'zone': None, 'project': Munch({'id': '2f6d334cc33648e5b38b5ac1
main_name': 'Default'}})))

```

图 9: 网络列表

如果我们使用网络 id 尝试将浮动 ip 绑定到 “base” 与 “ydx” 两个网络上的时候，会有报错指出该网络不是一个外部网络（10）

```
Bad floatingip request: Network 7b9a3af8-f32b-4b3d-9b8d-3553917efd9c is not a valid external network.
```

图 10: 外部网络报错

我们发现 “base” 网络和我们创建的 “ydx” 网络中有一个名为 “is\_router\_external” 的参数值都为 False，而 “public” 网络改参数的值为 True。因此我们猜测三个网络中只有 “public” 可以连通外网，可以被浮动 ip 绑定。将浮动 ip 绑定到该网络上后我们发现默认绑定的网络就是校园网。

## 6 拓展调查

### 6.1 高并发服务

经过思考，如果真正完全实现云服务最重要的一点还要使得网络具有高并发服务的能力，当用户数量非常多，如何同时处理大量的请求，能够同时相应用户需求而不带来明显的延时。当用户长时间或者服务器长时间没有相应该如何处理。因此我们对于高并发服务的问题进行了简单的调查。

4 大具有代表性的并发模型：PPC，TPC 模型，select 模型和 poll 模型、Epoll 模型。PPC 模型是多进程模型，每当有一个连接时就开辟一个进程进行处理，这是比较简单的模型，但是实际并不能承载很高的模型数，并且在 UNIX 中开辟进程的开销也是相当大的。

而 TPC 则是多线程模型，虽然没有了进程的复杂性，但是线程之间的共享和互斥也是重要的问题，如何合理访问共享资源也是需要思考的一个问题。处理线程互相影响和性能问题，我们可以使用 prethread 模式，在用户发送请求之前就提前开辟好线程，当有用户请求的时候再分享即可。Apache 服务对两种模型都是支持的。另外需要考虑的就是 I/O 访问模式，我们应当使用非阻塞式同步 I/O 或异步 I/O 来操作，因为如果同步的 I/O 可能会因为线程数量过多导致阻塞，影响性能，具体实现可以用 linux 内置的 epoll 服务。

## 6.2 openstack 与个性化服务

虽然项目中我们主要使用 openstack API，没有使用 openstack 但是我们对 openstack 平台也做了一些简单的了解。Openstack 是一个集成了多工具的平台大概各个组件部分如图 11所示。

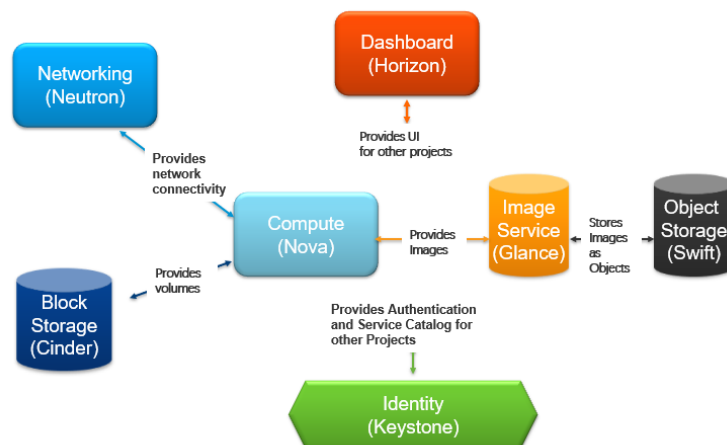


图 11: Openstack 组成

另外我们关注到的另外一个问题是个性化服务，我们可以提供更多不同 solidwrok 版本例如 lite 和 pro 两种，以及不同的内核数目，硬盘空间和内存大小满足不同的需求，当然这里面没有太多的技术拓展，我们也就不全部一一实现了。

## 7 总结感想

在本次工程设计实践中，我们成功地设计了一个可以为用户提供云 solidworks 服务的网站，从 solidworks 在云虚拟机上的装载，调用 API 创建和控制远程虚拟机，到设计网站为用户提供交互式服务，我们在本次实践中学到了太多平日课堂上不会传授的知识。所谓实践是最好的老师，无论是通过查阅文档来实现调用 openstack API 的脚本，还是制作在线镜像维护网络服务器，抑或构建网站访问控制逻辑和后端数据库的维护与更新，如果不加

以实践就只是纸上功夫，落不到实处。我们在工程中，既是建设者，同时也是体验者，每次完成工程的一部分后，都在作为用户体验着自己设计的服务。我们在体验中琢磨自己的设计是否还有更进一步的空间，自己作为用户还希望从哪些方面获取更好的服务，基于此进一步磨砺自己的设计。这是我们在交大第一次完成可以直接面向用户提供服务的工程，在完成后我们在使用线上的云服务，甚至只是访问网页的时候，看到的事物都已不再一样了。我们会在享受他人提供的云服务的时候思考他们是怎么优化服务的，他们的工作和我们的有什么相同之处，相异之处，能不能从中寻找到启发，为我们现在的，以及未来许许多多的工程工作提供指导呢？感谢为我们提供了本次实践机会的老师，我们会将此作为起点，从服务用户的角度，向工程设计这个集技术与服务于一体的广阔世界迈出崭新的一步。

## 致谢

感谢上海交通大学网络信中心的罗萱、何斐老师对我们项目的指导和意见，以及管海兵老师对工程实践与科技创新 *iii-D* 课程的指导！

## 参考文献

- [1] <https://www.solidworks.com/>
- [2] <http://lic.si.sjtu.edu.cn/Softs/good/id/1609>
- [3] [https://docs.openstack.org/zh\\_CN/](https://docs.openstack.org/zh_CN/)
- [4] [https://docs.openstack.org/zh\\_CN/api-quick-start/](https://docs.openstack.org/zh_CN/api-quick-start/)

## 附录

### A. 同学对软件安装使用困难的吐槽



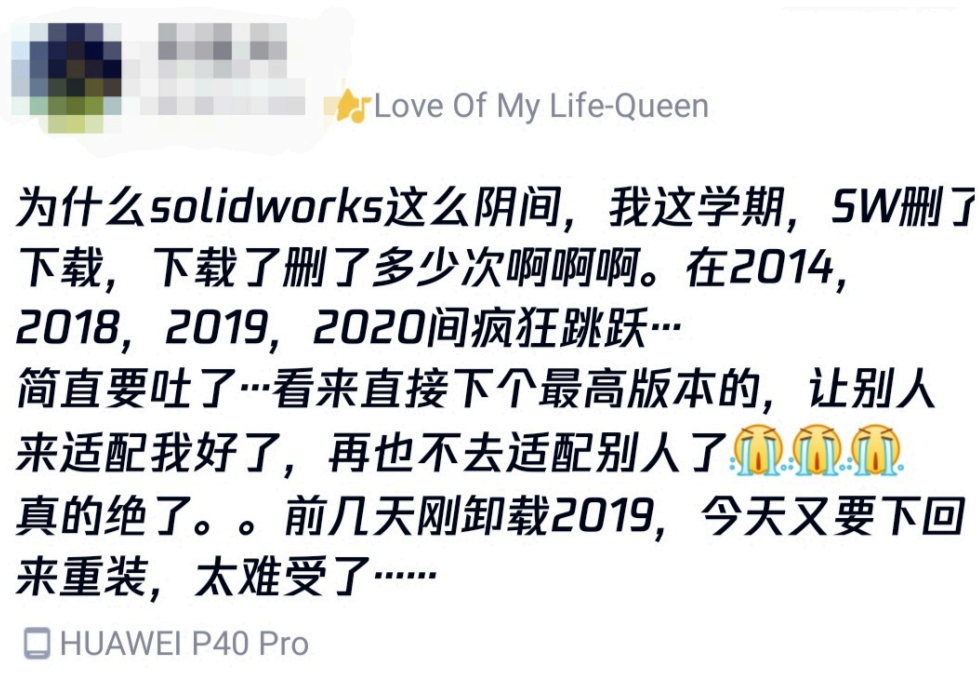


图 12: 同学反映

#### B. Python 后端脚本调用云主机创建

```
from openstack import connection
import os
import errno
from keystoneauth1.identity import v3
from keystoneauth1 import session
from keystoneclient.v3 import client
import sys
import pandas as pd

auth = v3.Password(auth_url='https://home.jcloud.sjtu.edu.cn:5000/v3',
                   username='Jason',
                   password='200414Lhx',
                   project_name='jimmyyao18_project',
                   user_domain_name='Default',
                   project_domain_name='Default'
)

sess = session.Session(auth=auth)
```

```
conn = connection.Connection(
    session = sess,
    region_name='RegionOne',
    identity_interface='public',
)
conn.authorize()

def list_servers(conn):
    print("List Servers:")

    for server in conn.compute.servers():
        print(dict(server)['name'])

def list_images(conn):
    print("List Images:")

    for image in conn.compute.images():
        print(dict(image)['name'])

def list_flavors(conn):
    print("List Flavors:")

    for flavor in conn.compute.flavors():
        print(dict(flavor)['name'])
        #print(flavor)

def list_networks(conn):
    print("List Networks:")

    for network in conn.network.networks():
        print(dict(network)['name'])
        print(network)

def create_keypair(conn):
    keypair = conn.compute.find_keypair('solidworks')

    if not keypair:
        print("Create Key Pair:")

        keypair = conn.compute.create_keypair(name='solidworks')
```



```

print(keypair)

try:
    os.mkdir('./ssh_dir')
except OSError as e:
    if e.errno != errno.EEXIST:
        raise e

with open('./ssh_dir/private_keypair_file', 'w') as f:
    f.write("%s" % keypair.private_key)

os.chmod('./ssh_dir/private_keypair_file', 0o400)

return keypair

def create_server(conn):
    print("Create Server:")

    image = conn.compute.find_image("solidworks_19pro")
    flavor = conn.compute.find_flavor("research.4C.16G")
    network = conn.network.find_network("ydx")
    keypair = create_keypair(conn)

    i = 0
    for server in conn.compute.servers():
        i = i + 1
        if(i<5):
            with open("/usr/local/httpd/htdocs/pages/tmp/tmp_limit.txt","w")
                as f:
                    f.write("0")
        if(i >= 5):
            with open("/usr/local/httpd/htdocs/pages/tmp/tmp_limit.txt","w")
                as f:
                    f.write("1")
            return

    server = conn.compute.create_server(
        name=sys.argv[1], image_id=image.id, flavor_id=flavor.id,
        networks=[{"uuid": network.id}], key_name='solidworks')

    server = conn.compute.wait_for_server(server)

```

```

#conn.compute.change_server_password(server, '123456')

print("ssh -i {key} root@{ip}".format(
    key='./ssh_dir/private_keypair_file',
    ip=server.access_ipv4))

return server

def create_float_ip(conn):
    floatip = conn.network.create_ip(
        floating_network_id = '0145335c-c9b0-40bd-bc4b-73cd530ef934'
    )
    print("floating_ip:")
    address=floatip.floating_ip_address
    print(address)

    return floatip,address

def add_float_ip(conn, server, floatip):
    conn.compute.add_floating_ip_to_server(server, floatip.floating_ip_address
    )

def main():
    list_servers(conn)
    #list_images(conn)
    #list_flavors(conn)
    #list_networks(conn)
    #list_serverste_keypair(conn)
    server = create_server(conn)
    floatip,address = create_float_ip(conn)
    add_float_ip(conn, server, floatip)

    file = open(r'/usr/local/httpd/htdocs/pages/tmp/tmp_address.txt', 'w')
    file.write(str(address))
    file.close()

    file = open(r'/usr/local/httpd/htdocs/pages/tmp/current_ip.txt', 'w')
    file.write(str(address))
    file.close()

    file = open(r'/usr/local/httpd/htdocs/pages/tmp/current_username.txt', 'w'
    )

```

```

file.write(sys.argv[1])
file.close()

file = open(r'/usr/local/httpd/htdocs/pages/tmp/current_userpassword.txt',
            'w')

file.write('Unknown')
file.close()

file = open(r'/usr/local/httpd/htdocs/pages/tmp/current_account.txt', 'r')
account = file.read()
file.close()

df = pd.read_csv(r'/usr/local/httpd/htdocs/pages/inf/user_data.txt')
for i in range(len(df)):
    if str(df.iloc[i].account) == account:
        break

df1=df.copy()
df1.ip[i] = str(address)
df1.username[i] = sys.argv[1]
df.userpassword[i] = 'Unknown'

df1.to_csv(r'/usr/local/httpd/htdocs/pages/inf/user_data.txt', index=False
          )

if __name__=="__main__":
    main()

```

云主机删除

```

from openstack import connection
import os
import errno
import sys

conn = connection.Connection(
    auth_url='https://home.jcloud.sjtu.edu.cn:5000/v3',
    username='Jason',
    password='200414Lhx',
    project_name='jimmyao18_project',

```

```

        user_domain_name='Default',
        project_domain_name='Default',
        region_name='RegionOne',
        identity_interface='public'
    )
conn.authorize()

def delete(conn):
    if(len(sys.argv)<3):
        print("Error")
    else:
        server = conn.compute.find_server(sys.argv[1])
        conn.compute.delete_server(server)
        ip = conn.network.find_ip(sys.argv[2])
        conn.network.delete_ip(ip)

delete(conn)

```

## 修改云主机密码

```

from openstack import connection
import os
import errno
from keystoneauth1.identity import v3
from keystoneauth1 import session
from keystoneclient.v3 import client
import sys

auth = v3.Password(auth_url='https://home.jcloud.sjtu.edu.cn:5000/v3',
                   username='Jason',
                   password='200414Lhx',
                   project_name='jimyyao18_project',
                   user_domain_name='Default',
                   project_domain_name='Default'
                  )

sess = session.Session(auth=auth)
sess.get('https://home.jcloud.sjtu.edu.cn:5000/v3')

conn = connection.Connection(
    session = sess,

```

```

        region_name='RegionOne',
        identity_interface='internal',
    )
conn.authorize()

def change(conn):
    if(len(sys.argv)<3):
        print("Error")
    else:
        server = conn.compute.find_server(sys.argv[1])
        conn.compute.change_server_password(server, sys.argv[2])

change(conn)

```

## 云主机开机

```

from openstack import connection
import os
import errno
from keystoneauth1.identity import v3
from keystoneauth1 import session
from keystoneclient.v3 import client
import sys

auth = v3.Password(auth_url='https://home.jcloud.sjtu.edu.cn:5000/v3',
                   username='Jason',
                   password='200414Lhx',
                   project_name='jimyyao18_project',
                   user_domain_name='Default',
                   project_domain_name='Default'
                  )

sess = session.Session(auth=auth)
sess.get('https://home.jcloud.sjtu.edu.cn:5000/v3')

conn = connection.Connection(
    session = sess,
    region_name='RegionOne',
    identity_interface='internal',
)

```

```

conn.authorize()

def open_server(conn):
    if(len(sys.argv)<2):
        print("Error")
    else:
        server = conn.compute.find_server(sys.argv[1])
        conn.compute.start_server(server)

open_server(conn)

```

## 云主机关机

```

from openstack import connection
import os
import errno
from keystoneauth1.identity import v3
from keystoneauth1 import session
from keystoneclient.v3 import client
import sys

auth = v3.Password(auth_url='https://home.jcloud.sjtu.edu.cn:5000/v3',
                    username='Jason',
                    password='200414Lhx',
                    project_name='jimyyao18_project',
                    user_domain_name='Default',
                    project_domain_name='Default'
                )

sess = session.Session(auth=auth)
sess.get('https://home.jcloud.sjtu.edu.cn:5000/v3')

conn = connection.Connection(
    session = sess,
    region_name='RegionOne',
    identity_interface='internal',
)

conn.authorize()

def stop(conn):
    if(len(sys.argv)<2):

```

```
        print("Error")
    else:
        server = conn.compute.find_server(sys.argv[1])
        conn.compute.stop_server(server)

stop(conn)
```

## 云主机硬重启

```
from openstack import connection
import os
import errno
from keystoneauth1.identity import v3
from keystoneauth1 import session
from keystoneclient.v3 import client
import sys

auth = v3.Password(auth_url='https://home.jcloud.sjtu.edu.cn:5000/v3',
                    username='Jason',
                    password='200414Lhx',
                    project_name='jimmyao18_project',
                    user_domain_name='Default',
                    project_domain_name='Default'
                    )

sess = session.Session(auth=auth)
sess.get('https://home.jcloud.sjtu.edu.cn:5000/v3')

conn = connection.Connection(
    session = sess,
    region_name='RegionOne',
    identity_interface='internal',
)

conn.authorize()

def reboot_HARD(conn):
    if(len(sys.argv)<2):
        print("Error")
    else:
        server = conn.compute.find_server(sys.argv[1])
        conn.compute.reboot_server(server,"HARD")
```

```
reboot_HARD(conn)
```

## 云主机软重启

```
from openstack import connection
import os
import errno
from keystoneauth1.identity import v3
from keystoneauth1 import session
from keystoneclient.v3 import client
import sys

auth = v3.Password(auth_url='https://home.jcloud.sjtu.edu.cn:5000/v3',
                   username='Jason',
                   password='200414Lhx',
                   project_name='jimyyao18_project',
                   user_domain_name='Default',
                   project_domain_name='Default'
                  )

sess = session.Session(auth=auth)
sess.get('https://home.jcloud.sjtu.edu.cn:5000/v3')

conn = connection.Connection(
    session = sess,
    region_name='RegionOne',
    identity_interface='internal',
)

conn.authorize()

def reboot_HARD(conn):
    if(len(sys.argv)<2):
        print("Error")
    else:
        server = conn.compute.find_server(sys.argv[1])
        conn.compute.reboot_server(server,"SOFT")

reboot_HARD(conn)
```

## C. 网页后端实现细节



html 指令通过 post 指令进行调用

php 收到 post 指令，进行后端操作通过函数 exec 执行 shell 语句。E;. 网页后端 php 文件源码 main.php

```
<?php
```

```
    $command=$_POST['command'];
    if ($command=="Get Solidwork"){
        $handle = fopen("./pages/tmp/current_ip.txt", 'r');
        $ip=fread($handle,1024);
        $handle = fopen("./pages/tmp/current_account.txt", 'r');
        $ac=fread($handle,1024);

$flag=0;

        if ($ip == "Undefined"){
            exec("python3 ./pages/script/create.py $ac");
            sleep(40);
            $flag='1';
        }
        $handle = fopen("./pages/tmp/current_ip.txt", 'r');
        $ip=fread($handle,1024);
        echo "Your address is $ip\n";
        if($flag=='1')echo "Please change password when
first time using";
        exit;
    }
    if ($command=="Delete Solidwork"){
        $handle = fopen("./pages/tmp/current_ip.txt", 'r');
        $ip=fread($handle,1024);
        $handle = fopen("./pages/tmp/current_username.txt", 'r');
        $usr=fread($handle,1024);
        $handle = fopen("./pages/tmp/current_account.txt", 'r');
        $ac=fread($handle,1024);
        if ($ip != "Undefined"){
            exec("python3 ./pages/script/delete.py $usr $ip");
            exec("python3 ./pages/utlis/refresh_inf.py");
            echo "Deleted";
        }
    }
}
```

```

}
else{
    echo "It seems that you have not created a
solidwork yet.";
}

exit;
}
if ($command=="Open Solidwork"){
    $handle = fopen("./pages/tmp/current_ip.txt", 'r');
    $ip=fread($handle,1024);
    $handle = fopen("./pages/tmp/current_username.txt", 'r');
    $usr=fread($handle,1024);
    if ($ip != "Undefined"){
        exec("python3 ./pages/script/open.py $usr");
        echo "Opened";
    }

    else

        echo "It seems that you have not created a
solidwork yet.";

    exit;
}
if ($command=="Close Solidwork"){
    $handle = fopen("./pages/tmp/current_ip.txt", 'r');
    $ip=fread($handle,1024);
    $handle = fopen("./pages/tmp/current_username.txt", 'r');
    $usr=fread($handle,1024);
    if ($ip != "Undefined"){
        exec("python3 ./pages/script/close.py $usr");
        echo "Closed";
    }

    else

        echo "It seems that you have not created a
solidwork yet.";

    exit;
}

```

```

}
if ($command=="Soft Reboot"){
    $handle = fopen("./pages/tmp/current_ip.txt", 'r');
    $ip=fread($handle,1024);
    $handle = fopen("./pages/tmp/current_username.txt", 'r');
    $usr=fread($handle,1024);
    if ($ip != "Undefined"){
        exec("python3 ./pages/script/reboot_soft.py $usr");
        echo "Rebooted";
    }
    else
        echo "It seems that you have not created a
        solidwork yet.";
    exit;
}
if ($command=="Hard Reboot"){
    $handle = fopen("./pages/tmp/current_ip.txt", 'r');
    $ip=fread($handle,1024);
    $handle = fopen("./pages/tmp/current_username.txt", 'r');
    $usr=fread($handle,1024);
    if ($ip != "Undefined"){
        exec("python3 ./pages/script/reboot_hard.py $usr");
        echo "Rebooted";
    }
    else
        echo "It seems that you have not created a
        solidwork yet.";
    exit;
}
if ($command=="Show address"){
    $handle = fopen("./pages/tmp/current_ip.txt", 'r');
    $ip=fread($handle,1024);
    if($ip=="Undefined")echo "It seems that you have
    not created a solidwork yet.";
}

```

```

        else echo "Your address is $ip";
        exit;
    }
    if ($command=="Change Password(of your solidwork)") {
        $new_password = $_POST["password"];
        if (strlen($new_password) >= 6) {
            $handle = fopen("./pages/tmp/current_ip.txt", 'r');
            $ip = fread($handle, 1024);
            $handle = fopen("./pages/tmp/current_username.txt",
                'r');
            $usr = fread($handle, 1024);
            if ($ip != "Undefined") {
                exec("python3 ./pages/script/
                    change_password.py $usr $new_password");
                exec("python3 ./pages/utils/
                    refresh_password.py $new_password");
                echo "Changed";
            }
            else echo "It seems that you have not created
                a solidwork yet.";
        }
        else
            echo "It seems that your password is too short.
                (length < 6)";
        exit;
    }
}
?>

```

login.php

<?php

```

$userid = $_POST['userid'];
$password = $_POST['password'];
exec("python3 ./pages/utils/login_check.py $userid", $output, $ret);
sleep(1);
$handle = fopen('./pages/tmp/tmp_login_check.txt', 'r');

```

```

if ($userid=='')$flag='1';
else $flag=fgets($handle);
if ($flag=='0'){
    exec("python3 ./pages/utils/login_get.py
    $userid $password");
    sleep(1);
    $handle = fopen( './pages/tmp/tmp_login_get.txt ', 'r ');
    if ($password=='')$flag2='0';
    else $flag2=fgets($handle);
    if ($flag2=='1'){
        exec("python3 ./pages/utils/write_inf.py $userid");
        header('Location: Main.html');
    }
    else{
        echo 'Sorry, it seems that your password is
        wrong.';
    }
}
else{
    echo 'Sorry, it seems that your User Id doesnt exist?';
}

```

?>

regiser.php

<?php

```

$password =$_POST["password"];
$confirm = $_POST["password2"];
$userid = $_POST["userid"];
if ($password == $confirm)
{
    if (strlen($userid)<6||strlen($password)<6)
        echo "Sorry, your User Id or Password is
        too short.(length<6)";
    else
        {

```

```

        exec("python3 ./pages/Utils/
        register_check.py $userid");
        sleep(1);
        $handle = fopen('./pages/tmp
        /tmp_register_check.txt', 'r');
        $flag=fgets($handle);
        if ($flag == "1")
        {
            exec("python3
            ./pages/Utils/
            register_insert.py $userid
            $password");

            echo "Your account has been
            created, now get your
            Solidwork!";
        }
        else
            echo "Sorry, this User Id
            has already been
            registered.";
    }
}

else
    echo "Error: Your Password should be same as your
    Confirmation.";
?>

```

E. php 中所调用的非 openstack 交互脚本, utils 工具脚本 login\_check.py

```

import sys
import pandas as pd

df = pd.read_csv(r'/usr/local/httpd/htdocs/pages/inf/user_data.txt')
account = sys.argv[1]
if account in df.account.values:
    f = 0

```

```
else:
    f = 1
file = open(r'/usr/local/httpd/htdocs/pages/tmp/tmp_login_check.txt', 'w')
file.write(str(f))
file.close()
```

login\_get.py

```
import sys
import pandas as pd

df = pd.read_csv(r'/usr/local/httpd/htdocs/pages/inf/user_data.txt')
account, password = sys.argv[1], sys.argv[2]
f = 0
for i in range(len(df)):
    if str(df.iloc[i].account) == account and str(df.iloc[i].password) == password:
        f = 1
        break

file = open(r'/usr/local/httpd/htdocs/pages/tmp/tmp_login_get.txt', 'w')
file.write(str(f))
file.close()
```

refresh\_inf.py

```
import sys
import pandas as pd

file = open(r'/usr/local/httpd/htdocs/pages/tmp/tmp_address.txt', 'w')
file.write("Undefined")
file.close()

file = open(r'/usr/local/httpd/htdocs/pages/tmp/current_ip.txt', 'w')
file.write("Undefined")
file.close()

file = open(r'/usr/local/httpd/htdocs/pages/tmp/current_username.txt', 'w')
file.write("Undefined")
file.close()

file = open(r'/usr/local/httpd/htdocs/pages/tmp/current_userpassword.txt', 'w')
)
```

```

file.write("Undefined")
file.close()

file = open(r'/usr/local/httpd/htdocs/pages/tmp/current_account.txt', 'r')
account = file.read()
file.close()

df = pd.read_csv(r'/usr/local/httpd/htdocs/pages/inf/user_data.txt')
for i in range(len(df)):
    if str(df.account[i]) == account:
        break

df1=df.copy()
df1.ip[i] = "Undefined"
df1.username[i] = "Undefined"
df1.userpassword[i] = "Undefined"

df1.to_csv(r'/usr/local/httpd/htdocs/pages/inf/user_data.txt', index=False)

```

refresh\_password.py

```

import sys
import pandas as pd

file = open(r'/usr/local/httpd/htdocs/pages/tmp/current_userpassword.txt', 'w'
            )

file.write(sys.argv[1])
file.close()

file = open(r'/usr/local/httpd/htdocs/pages/tmp/current_account.txt', 'r')
account = file.read()
file.close()

df = pd.read_csv(r'/usr/local/httpd/htdocs/pages/inf/user_data.txt')
for i in range(len(df)):
    if str(df.iloc[i].account) == account:
        break

df1=df.copy()
df1.userpassword[i] = sys.argv[1]

df1.to_csv(r'/usr/local/httpd/htdocs/pages/inf/user_data.txt', index=False)

```



register\_check.py

```
import sys
import pandas as pd

df = pd.read_csv(r'/usr/local/httpd/htdocs/pages/inf/user_data.txt')
account = sys.argv[1]
if account in df.account.values:
    f = 0
else:
    f = 1
file = open(r'/usr/local/httpd/htdocs/pages/tmp/tmp_register_check.txt', 'w')
file.write(str(f))
file.close()
```

register\_insert.py

```
import sys
import pandas as pd

df = pd.read_csv(r'/usr/local/httpd/htdocs/pages/inf/user_data.txt')
account, password = sys.argv[1], sys.argv[2]
inf = pd.DataFrame({
    'account': account,
    'password': password,
    'username': 'Undefined',
    'userpassword': 'Undefined',
    'ip': 'Undefined',
}, index=[0])
df = df.append(inf, ignore_index=True)
df.to_csv(r'/usr/local/httpd/htdocs/pages/inf/user_data.txt', index=False)
```

write\_inf.py

```
import sys
import pandas as pd

df = pd.read_csv(r'/usr/local/httpd/htdocs/pages/inf/user_data.txt')
account = sys.argv[1]
for i in range(len(df)):
    if str(df.iloc[i].account) == account:
        file = open(r'/usr/local/httpd/htdocs/pages/tmp/current_account.txt', 'w')
        file.write(str(df.iloc[i].account))
        file.close()
```

```
file = open(r'/usr/local/httpd/htdocs/pages/tmp/current_username.txt', 'w')
file.write(str(df.iloc[i].username))
file.close()

file = open(r'/usr/local/httpd/htdocs/pages/tmp/current_userpassword.txt', 'w')
file.write(str(df.iloc[i].userpassword))
file.close()

file = open(r'/usr/local/httpd/htdocs/pages/tmp/current_ip.txt', 'w')
file.write(str(df.iloc[i].ip))
file.close()
```

#### F. 镜像制作过程

第一步，选择安装好 solidworks 的主机所对应的系统硬盘，创建快照

第二步，该快照创建镜像

第三步，镜像创建完成

jCloud Management Platform

home.jcloud.sjtu.edu.cn/storage/

jCloud jimmyao10\_project 控制台 消费 子项目 jimmyao10

+ 创建云硬盘 删除 点击这里过滤

| 名称                  | 容量     | 状态  | 类型  | 挂载到                   | 硬盘标签 | 已用上限(快照) | 自动快照 | 自动备份 | 创建于  |
|---------------------|--------|-----|-----|-----------------------|------|----------|------|------|------|
| webserver_root_disk | 50 GB  | 使用中 | 容量型 | /dev/vda on webserver | 系统盘  | 0 / 64   | 否    | 否    | 18天前 |
| ydx_root_disk       | 100 GB | 使用中 | 性能型 | -                     | 数据盘  | 0 / 64   | 否    | 否    | 26天前 |

< 1 / 1 > 总计: 2 每页: 10

jCloud Management Platform

home.jcloud.sjtu.edu.cn/storage/snapshots/?tab=instancesSnapshot

jCloud jimmyao10\_project 控制台 消费 子项目 jimmyao10

云硬盘快照 云主机快照

+ 删除 点击这里过滤

| 名称  | 容量       | 状态 | 云硬盘                 | 云主机       | 创建类型 | 创建于  | 保留时间 |
|-----|----------|----|---------------------|-----------|------|------|------|
| tmp | 9.002 GB | 可用 | webserver_root_disk | webserver | 手动   | 几秒钟前 | 长期保留 |

< 1 / 1 > 总计: 1 每页: 10

jCloud Management Platform

home.jcloud.sjtu.edu.cn/compute/images

jCloud jimmyao10\_project 控制台 消费 子项目 jimmyao10

+ 删除 点击这里过滤

| 名称                  | 使用类型 | 类型      | 状态  | 可见性 | 创建于  |
|---------------------|------|---------|-----|-----|------|
| solidworks_19pro    | 云主机  | Windows | 运行中 | 私有  | 25天前 |
| Debian10_2-64-bit   | 云主机  | Linux   | 运行中 | 公有  | 2月前  |
| win8_1_En-64-bit    | 云主机  | Windows | 运行中 | 公有  | 6月前  |
| win8_1_Ch-64-bit    | 云主机  | Windows | 运行中 | 公有  | 6月前  |
| win2012_En-64-bit   | 云主机  | Windows | 运行中 | 公有  | 6月前  |
| win2012_Ch-64-bit   | 云主机  | Windows | 运行中 | 公有  | 6月前  |
| win10_Ch-64-bit     | 云主机  | Windows | 运行中 | 公有  | 7月前  |
| win10_En-64-bit     | 云主机  | Windows | 运行中 | 公有  | 7月前  |
| win10LTSC_Ch-64-bit | 云主机  | Windows | 运行中 | 公有  | 7月前  |
| win10LTSC_En-64-bit | 云主机  | Windows | 运行中 | 公有  | 7月前  |