

“天河星光” 用户使用手册

版本号	0.60
作者	黄鑫鑫
审核	
初始发布日期	2019/12/19
最新更新日期	2020/05/18
版权所有	国家超级计算广州中心
联系人	liang.du@nscc-gz.cn; shuaishuai.zhao@nscc-gz.cn
备注	

目录

“天河星光” 用户使用手册.....	1
一、 天河星光简介.....	4
二、 登录与注册.....	4
2.1 注册.....	5
2.1.1 通过邮箱注册.....	5
2.1.2 通过手机注册.....	6
2.2 登录.....	8
2.2.1 重置登录密码.....	8
2.2.2 申请系统账号.....	10
2.2.3 绑定已有系统账号.....	11
2.3 首页总览.....	12
2.4 忘记密码.....	13
2.3.1 通过邮箱找回.....	14
2.3.2 通过手机找回.....	15
三、 应用中心.....	16
3.1 应用列表.....	16
3.2 应用运行.....	16
3.2.1 使用默认配置.....	16
3.2.2 使用高级配置.....	22
四、 作业管理.....	24
4.1 作业列表.....	25
4.2 作业详情.....	25
4.2.1 基本信息.....	25
4.2.2 访问入口.....	26
五、 存储管理.....	35
5.1 传统方式.....	36
5.1.1 界面简介.....	36
5.1.2 上传文件.....	37

5.1.3 下载文件.....	38
5.1.4 复制文件.....	39
5.1 UDT 方式.....	42
5.1.1 切换方法.....	42
5.1.2 界面简介.....	43
5.1.3 上传文件.....	44
5.1.4 下载文件.....	46
5.1.5 复制文件.....	47
5.1.6 切换 TCP.....	47
六、在线提问.....	48
6.1 反馈问题.....	48
6.2 我的问题.....	48
七、用户中心.....	49
7.1 查看申请记录.....	49
7.2 查看机时充值记录.....	51
7.3 绑定微信.....	51
7.4 消息管理.....	52
八、镜像中心.....	53
8.1 镜像仓库.....	53
8.2 镜像管理.....	54
8.2.1 镜像配额.....	55
九、集群与分区.....	56
9.1 用户配额.....	56
9.2 集群与分区.....	56
9.2.1 通过 WebSSH 登录 Slurm 集群.....	57
十、积分兑换.....	58
十一、深度学习/大数据应用使用说明.....	59
11.1 加载平台已安装软件.....	60
11.1.1 查看已安装软件列表.....	60

11.1.2 加载/卸载已安装软件.....	60
11.2 网络代理设置.....	60
11.2.1 通过 module.....	60
11.2.2 通过环境变量.....	61
11.3 数据上传下载.....	61
11.4 pip 安装用户 python 包.....	65
11.5 自定义源码安装软件.....	65
11.6 常见应用使用说明.....	66
11.6.1 MongoDB 使用说明.....	66
11.6.2 RStudio 使用说明.....	72
11.6.3 Galaxy 使用说明.....	77
11.6.4 Tensorflow 使用说明.....	79
11.6.5 Mxnet 使用说明.....	82
11.6.6 PaddlePaddle 使用说明.....	84
11.6.7 Spark 使用说明.....	86

一、天河星光简介

“天河星光”平台是国家超级计算广州中心推出的融合高性能计算、大数据分析 with 人工智能应用的运行、支撑环境。该平台通过 Web Portal 提供服务统一入口，后端则对接广州超算中心内部的计算资源。

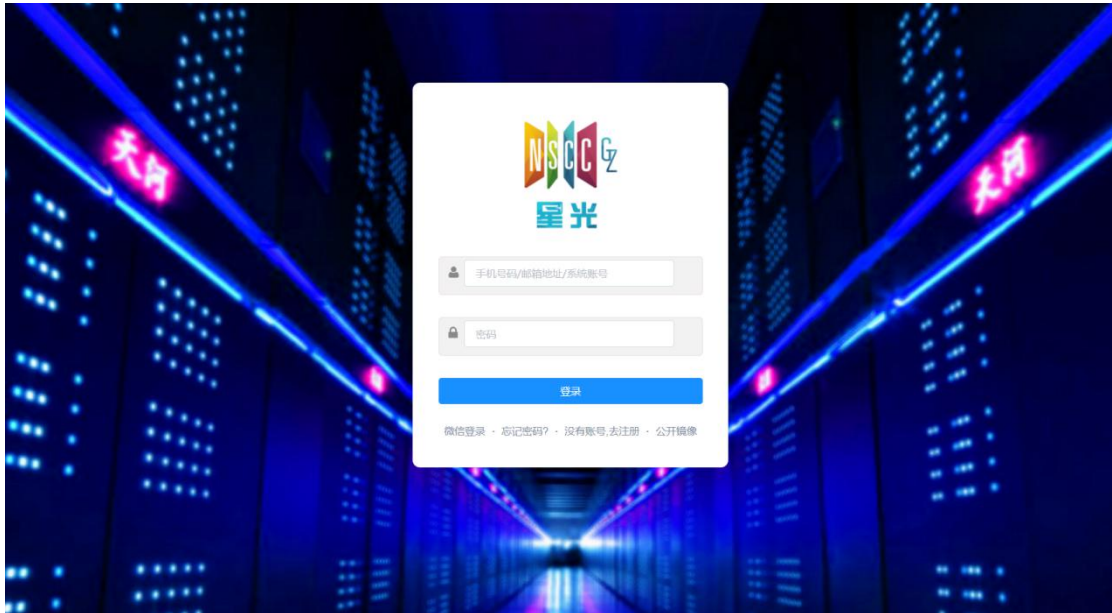
“星光”平台以应用为中心，为用户提供应用服务。用户可以直接在网页上填写应用表单，完成应用输入参数的设定并提交作业。“星光”既包含了传统的高性能计算应用(如工业仿真应用)，也涵盖了大数据、人工智能应用。用户可以在星光上便捷地创建自定义（自定义 CPU、GPU、内存数目）的深度学习集群。在“星光”上进行人工智能研究，无需用户自己手动配置复杂的深度学习框架和环境，方便用户快速进入数据分析过程。

“星光”的后端计算资源包含了多套集群，系统会根据用户提交的应用，智能地将作业引导至合适的集群。用户可以在“作业管理”中统一地对多套集群下的所有作业进行管理。用户的机时信息查询、配额调整申请则可以在“资源中心”中完成。

二、登录与注册

访问地址：<https://starlight.nscg-gz.cn/>

推荐使用 Google Chrome 浏览器访问星光系统。



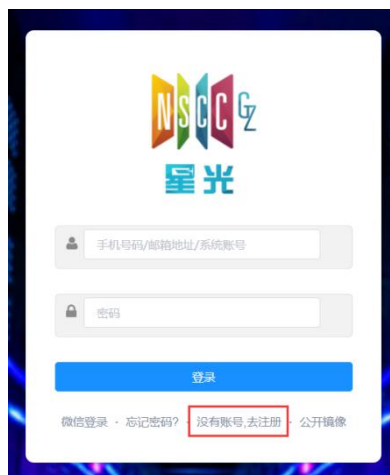
星光系统登录界面

无论用户是否已经拥有“天河二号”超级计算机的账号，第一次使用“星光”平台的用户需要通过手机或者邮箱注册一个星光账号。注册成功后，使用星光账号登录平台。

原星光平台（星光 v3 版本）用户可使用原来的星光账号直接登录新版本星光平台（星光 v4 版本），无需重新注册。

2.1 注册

第一次使用星光平台的用户，通过点击星光登录输入框下面的“没有账号，去注册”链接，可以进入到星光账号注册页面：



2.1.1 通过邮箱注册

进入账号注册界面后，默认的注册方式为邮箱注册，如下：

NSCC GZ
星光

邮箱注册 手机注册

邮箱地址

密码

确认密码

注册码

我已阅读并同意遵守《星光服务条款》

已有账号? [立即登录](#)

填写注册星光账号使用的邮箱地址和密码后，点击“获取注册码”按钮，系统会向指定邮箱发送一个六位数的注册码，如下：



注册码的有效时间为3分钟，请尽快完成注册。将收到的注册码输入后，点击“立即注册”按钮即可完成注册，若注册成功系统会自动跳到登录界面。

2.1.2 通过手机注册

在账号注册界面，点击上方的“手机注册”选项，可以切换到手机注册方式：

填写注册星光账号使用的手机号码和密码后，点击“获取注册码”按钮，系统会向指定手机号码发送一个六位数的验证码，如下：



注册码的有效时间为3分钟，请尽快完成注册。将收到的验证码输入后，点击“立即注册”按钮即可完成注册，若注册成功系统会自动跳到登录界面。

2.2 登录

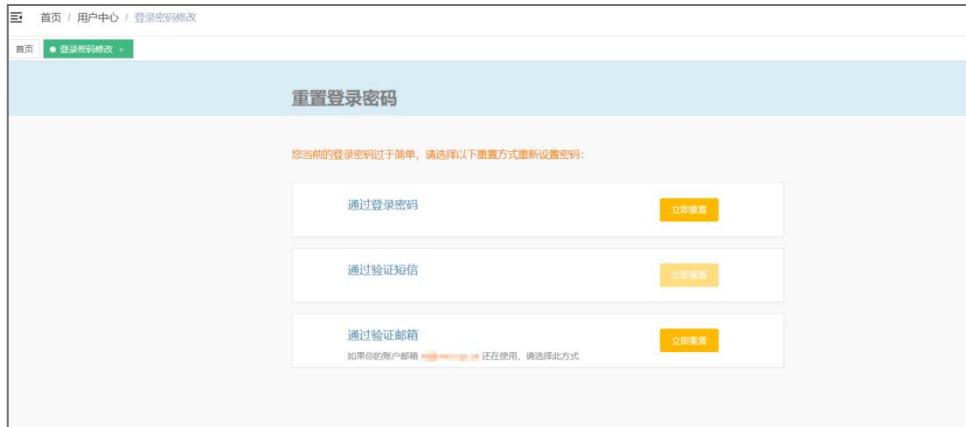
在星光的登录界面输入账号和密码后，点击登录按钮即可登录星光系统。



第一次登录时，若用户使用的是老版本的星光账号，则会进入重置登录密码界面，建议用户重置登录密码，重置方式见 [2.2.1](#)；若用户使用的是新注册的星光账号，会进入到首页，要求用户绑定一个已有的系统账号（见 [2.2.3](#)）或申请一个系统账号（见 [2.2.2](#)）。当绑定了微信账号后，还可以通过扫描微信二维码的方式登录星光平台，详见 [7.3](#)。

2.2.1 重置登录密码

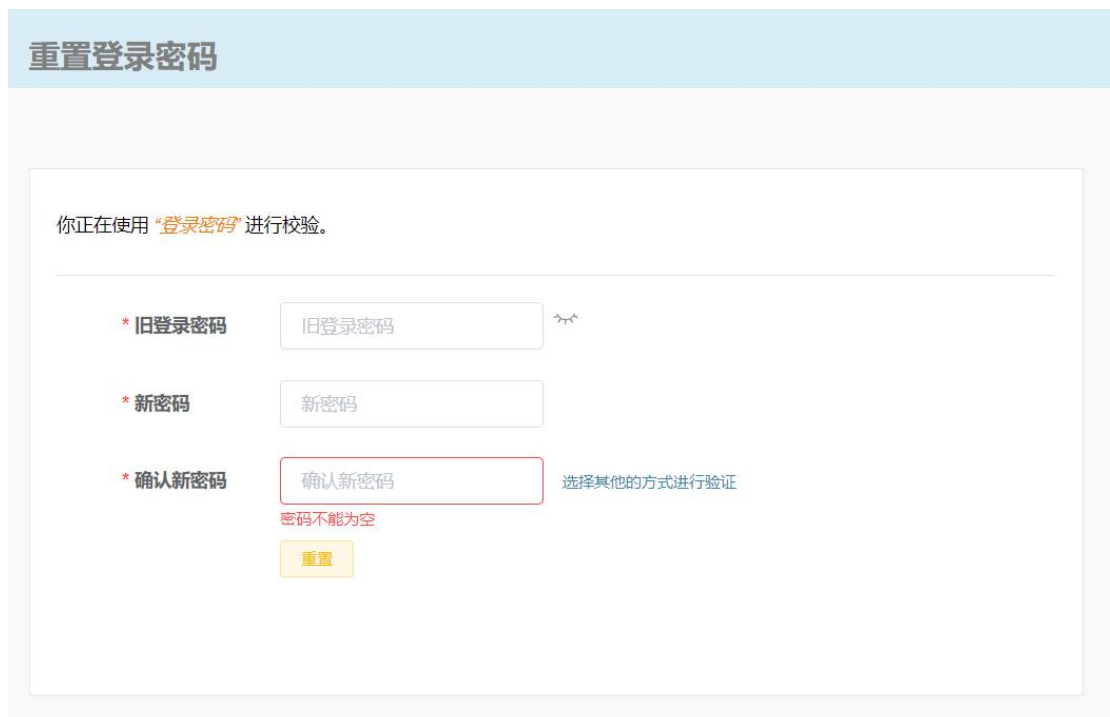
老版本星光（星光 v3）用户在第一次登录新版本的星光系统时，需要重置登录密码，系统会自动进入重置密码界面，如下：



重置密码有以下 3 种方式：通过原登录密码、验证短信和验证邮箱。

2.2.1.1 通过原登录密码

在重置密码界面，点击“通过登录密码”后面的“立即重置按钮”，进入重置登录密码界面：



输入原密码和新密码信息后，点击重置按钮，即可完成密码重置。

2.2.1.2 通过验证邮箱

在重置密码界面，点击“通过验证邮箱”后面的“立即重置按钮”，进入重置登录密码界面：



输入和账号关联的邮箱收到的验证码和新密码信息后，点击重置按钮，即可完成密码重置。

2.2.2 申请系统账号

新注册的星光用户需要绑定一个“天河二号”的系统账号，如果用户从来没有使用过“天河二号”，请点击首页右上角的“申请系统账号”：



点击“申请系统账号”后进入申请环节，填写申请表进行系统账号的申请：

审批名称	审批负责人	审批状态	审批意见
用户系统账号申请环节一	...	审批中	

用户申请表

* 用人单位

* 客户类型

* 应用平台

业务处理人

* 手机号码

* 联系人

* 邮箱地址

软件需求

额外需求

* 通讯地址

以 * 标识的为必须填写项。请尽量对每一项信息进行填写，有助于中心员工充分了解用户需求。

填完表后点击“提交”按钮提交系统账号申请。此时回到首页，可以看到审批正在进行中，耐心等待审批完成即可。



机时:

作业:

指南

天河星光使用手册v4 2019-12-09

用户申请表

您的用户申请表正在进行审核，请耐心等待审核结果

申请通过之后，便可以正常登录使用“星光”平台。在申请的过程中，用户可以登录平台，查看申请的状态。

2.2.3 绑定已有系统账号

新注册的星光用户需要绑定一个“天河二号”的系统账号，如果用户已经拥有“天河二号”系统账号，请点击首页右上角的“绑定系统账号”：



将“天河二号”系统账号和密钥文件填入，点击“绑定”按钮即可完成绑定：

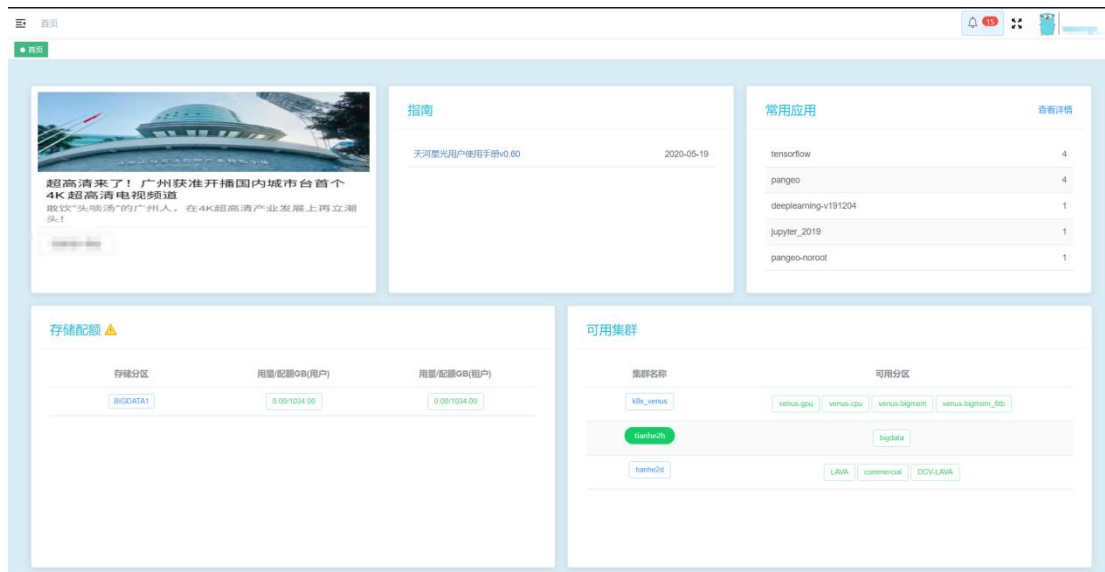
绑定说明

- **天河hpc账号** 是用户已经获取的天河二号高性能计算系统的账号，**不是** 您注册星光所用的邮箱或手机。不清楚的用户请检查“客户服务- service.nscg@nscg-gz.cn”发给您的**开户信息邮件**，格式一般如 xyz_npq_1
- **密钥文件** 一般在**开户信息邮件**的附件中，格式一般是 <天河hpc账号>.id。如果您通过putty 等软件转换过密钥文件的格式，会导致绑定失败，请使用原始的 RSA 格式。
- **私钥口令** 默认都是空值，除非您重新生成了SSH KEY 并进行了相关的设置

天河hpc账号	系统账号名(天河hpc账号)
请选择密钥文件	选择文件 未选择任何文件
私钥口令	私钥口令(phrase),默认为空

2.3 首页总览

首页集中展示用户相关信息，如下图：



各个信息包括：

1. 机时、作业信息。包括已用机时和剩余机时、正在运行中和排队中的作业。
2. 用户指南。包括用户手册等文档或教程。
3. 常用应用。显示用户常用的应用，点击相关应用项可快速访问该应用。
4. 存储配额。显示用户可用的存储分区及各个分区的配额信息。
5. 可用集群。显示用户可用的集群和集群分区。
6. 资源配额。显示用户的 Slurm 集群配额（包括作业提交数、作业运行数和节点数）以及 Kubernetes 集群配额（包括 CPU、GPU 和内存配额）。

星光平台拥有多个集群资源，根据用户使用的应用不同，星光将作业调度到适合应用运行且用户拥有使用权限的资源。星光提供对这些集群的统一管理服务，用户提交的作业可以在“作业管理”中统一管理。如果用户直接使用终端登录这些集群，只能看到登录的集群上运行的作业。统计信息目前不是实时信息，一般为按天汇总（如机时、存储信息）。

2.4 忘记密码

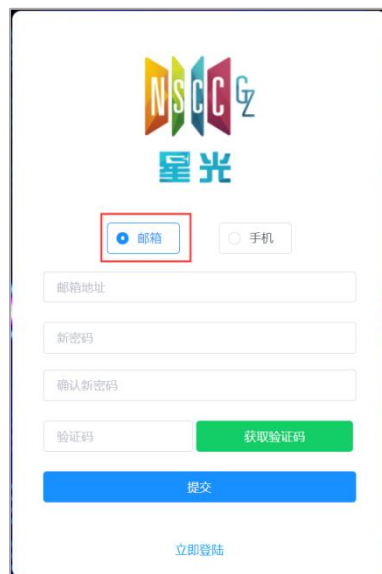
当忘记密码时，可通过登录界面下方的“忘记密码？”链接进入找回密码界面：



找回密码需要通过和账号绑定的邮箱或手机来找回。

2.3.1 通过邮箱找回

进入找回密码界面后，默认的找回方式即为通过邮箱找回，如下：



填写和星光账号绑定的邮箱地址和密码后，点击“获取验证码”按钮，系统会向指定邮箱发送一个六位数的验证码，如下：



输入收到的验证码和新密码信息后，点击“提交”按钮即可将密码重置为新密码，若重置成功系统会自动跳到登录界面。

2.3.2 通过手机找回

在找回密码界面，点击上方的“手机”选项，可以切换到手机找回方式：



填写星光账号绑定的手机号码，点击“获取验证码”按钮，系统会向指定手机号码发送一个六位数的验证码，如下：

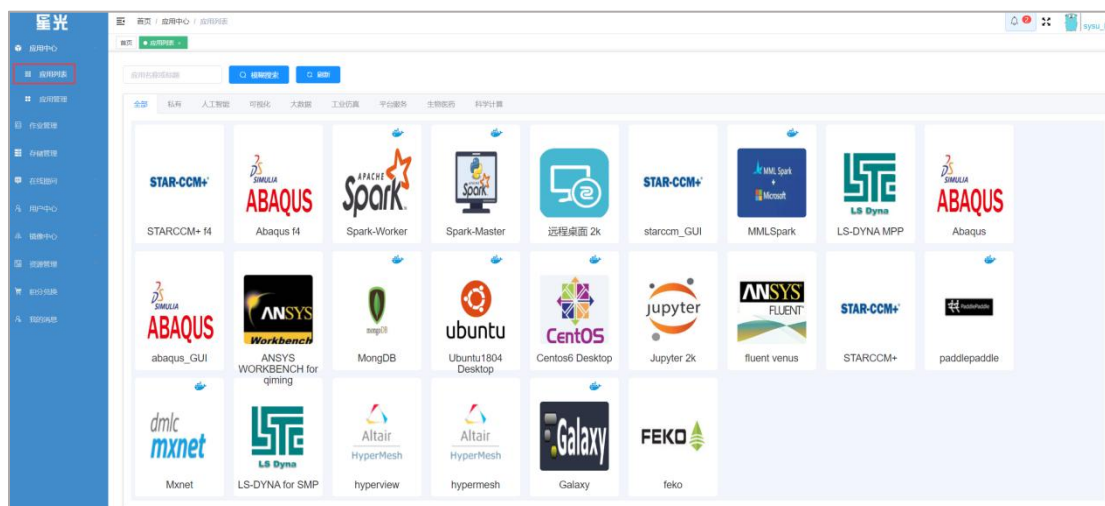


输入收到的验证码和新密码信息后，点击“提交”按钮即可将密码重置为新密码，若重置成功系统会自动跳到登录界面。

三、应用中心

3.1 应用列表

点击导航栏“应用中心” —> “应用列表”，将显示当前用户所有可用的应用（如下图）。右上角带蓝鲸标志的为容器应用（如深度学习和大数据应用），其与经典的 HPC 应用 在操作和展示上略有不同，文档后续涉及到的相关内容会做特别说明。

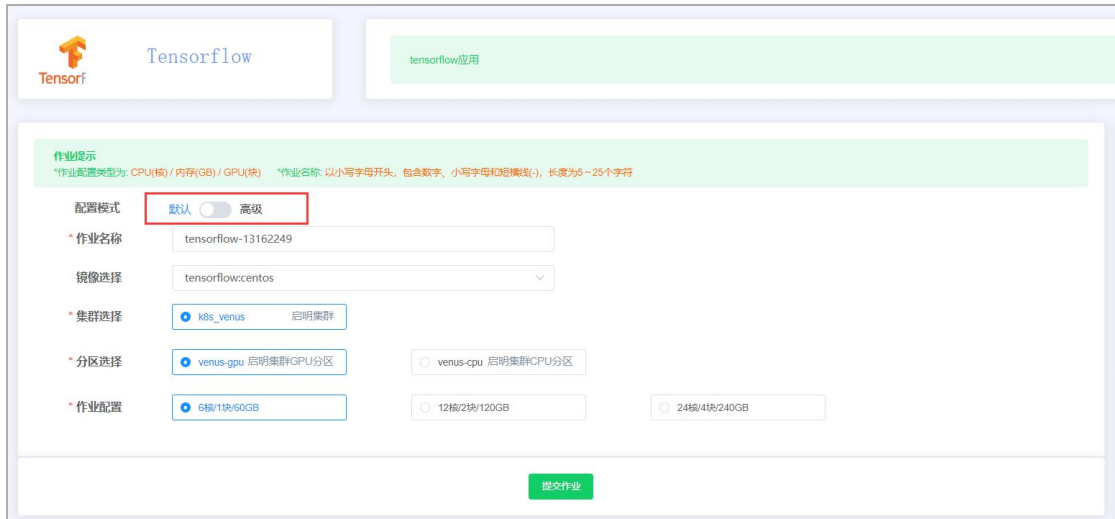


3.2 应用运行

用户在“应用列表”中直接点击应用图标即可进入作业提交界面，填写相应的配置后点击“提交作业”即可将应用提交到集群中运行。在提交作业时，对于容器应用，有“默认”和“高级”两种配置方式可选，详见 [3.2.1](#)、[3.2.2](#)。

3.2.1 使用默认配置

从“应用列表”点击应用图标进入作业提交界面后展示的是默认的配置方式：



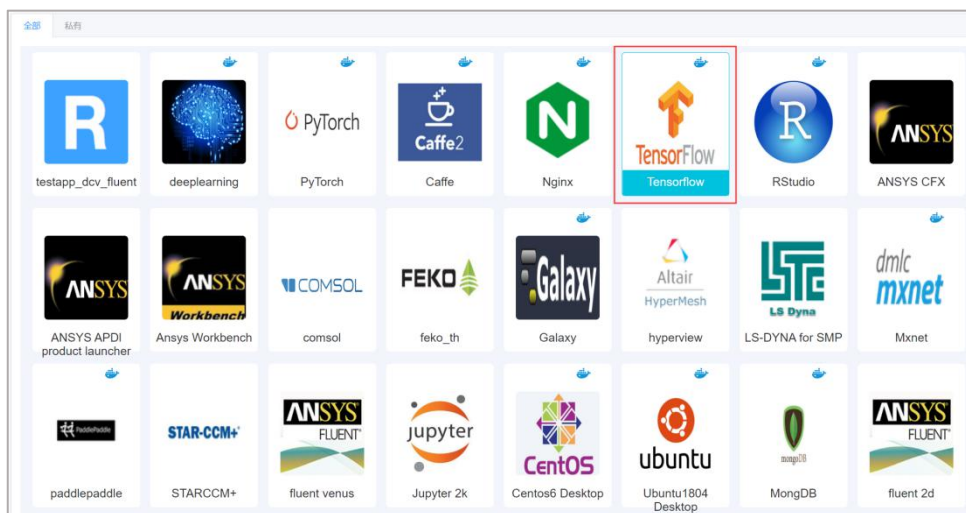
默认配置方式下用户只需选择应用的镜像版本、目标集群和分区以及作业的配置即可快速完成应用运行的配置,如需指定更多的配置项,则需使用“高级”配置方式(详见 [3.2.2](#))。

3.2.1.1 运行容器应用

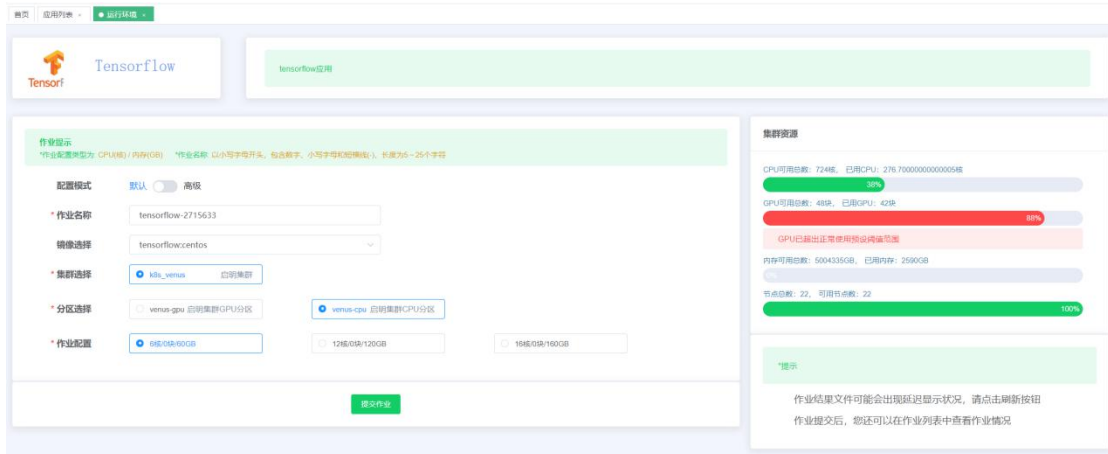
在“应用列表”中,带有蓝鲸标志的应用表示该应用为容器应用,本节以容器应用“Tensorflow”为例,演示容器应用的运行步骤,更多的应用运行说明见: [常见应用使用说明](#)。

运行“Tensorflow”应用步骤:

1. 点击应用列表中的“Tensorflow”图标,进入应用运行配置表单页面:



2. 配置应用的运行环境信息:



上图为默认的配置模式，各个主要的配置项说明如下：

- (1) 作业名称：提交的作业（在星光平台中，一个运行中的应用一般称之为作业）的名称，默认系统会自动生成一个
- (2) 镜像选择：选择该应用的 Docker 镜像，一般会有 centos 和 ubuntu 两个类型的镜像可供选择
- (3) 集群选择：选择作业要运行的目标集群
- (4) 分区选择：选择作业要运行的目标集群分区
- (5) 作业配置：选择运行作业的容器的配置，包括 CPU、GPU 和内存配置

系统会根据当前用户的配额和机时判断用户是否可以提交该作业。

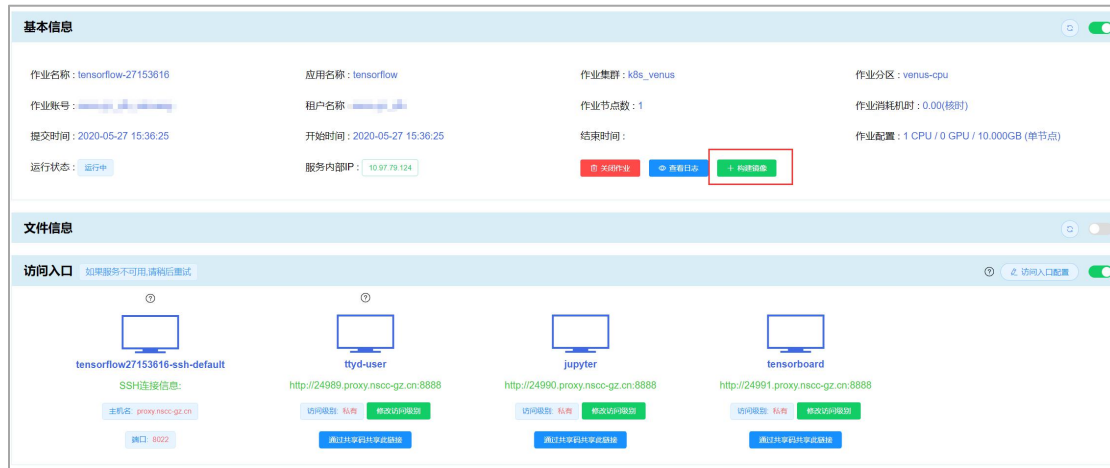
3. 点击“提交作业”按钮，提交作业，系统会自动跳转到提交的作业详情界面：



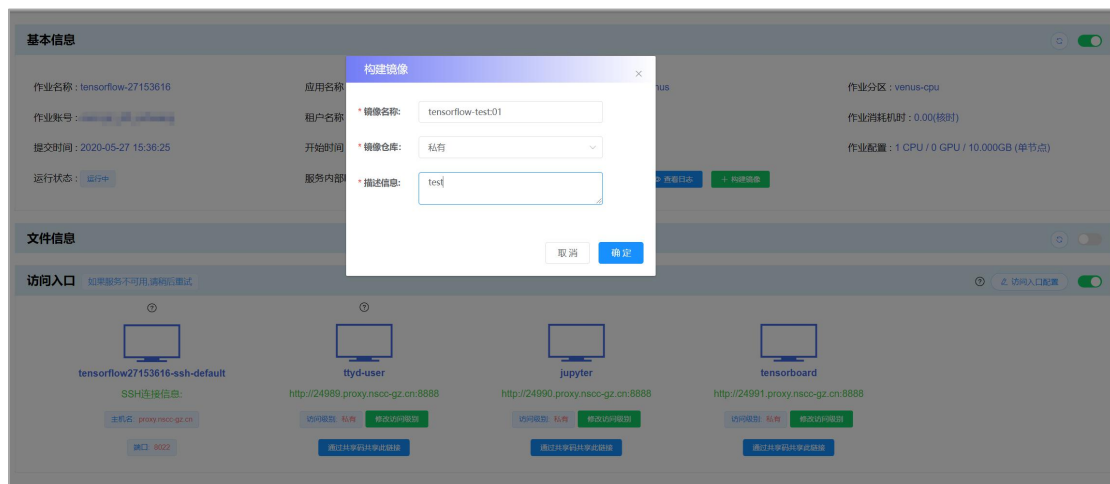
关于作业详情的更多信息，请参考：[4.2 节](#)。

3.2.1.2 构建容器镜像

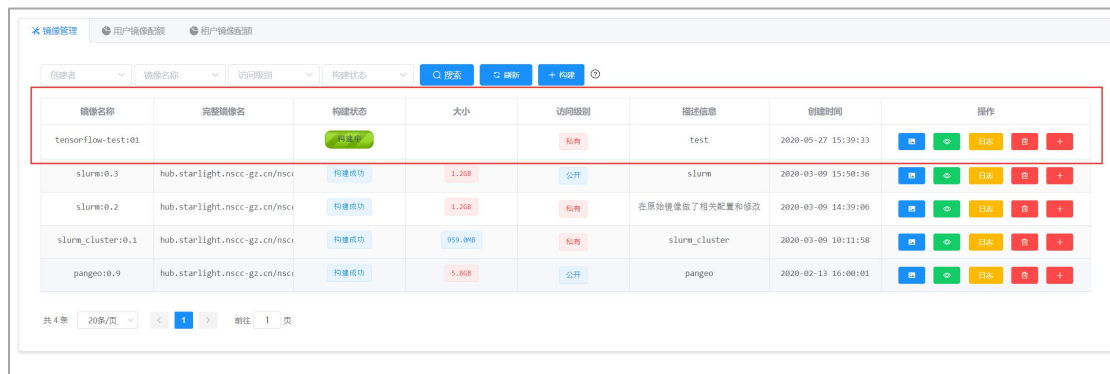
在容器应用的作业详情页面，可以点击“构建镜像”按钮，将当前运行的容器提交到镜像仓库中：



点击后填写镜像的相关信息：



点击“确定”按钮会跳转到“镜像管理”界面，显示构建镜像的状态：



构建成功后状态变为“构建成功”，如下：

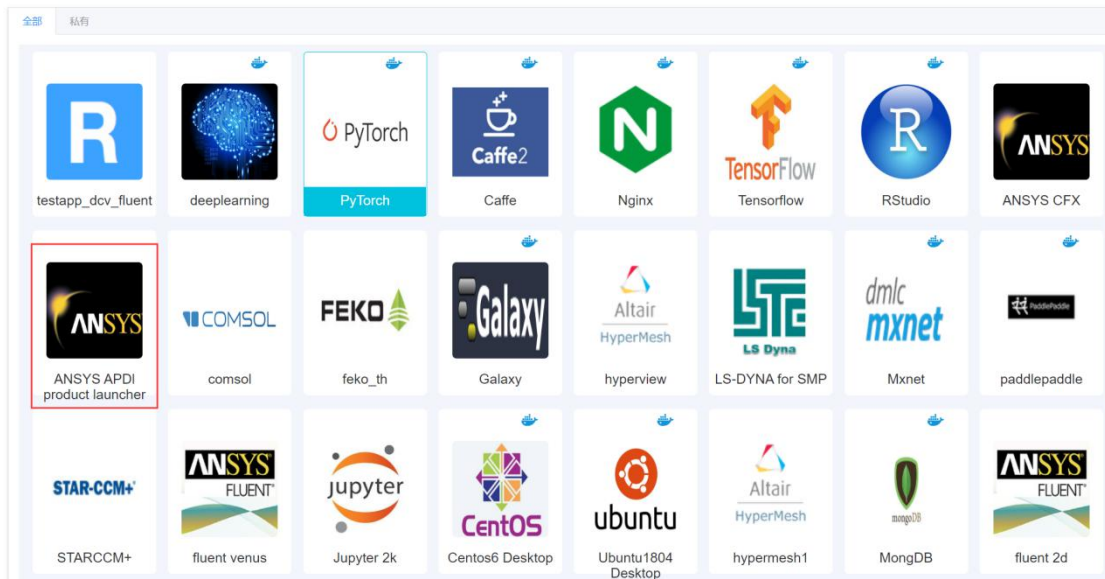
镜像名称	完整镜像名	构建状态	大小	访问级别	描述信息	创建时间	操作
tensorflow-test:01	hub.starlight.nssc-gz.cn/nssc	构建成功	956.0MB	私有	test	2020-05-27 15:39:33	[启动] [停止] [删除] [刷新]
slurm:0.3	hub.starlight.nssc-gz.cn/nssc	构建成功	1.2GB	公开	slurm	2020-03-09 15:50:36	[启动] [停止] [删除] [刷新]
slurm:0.2	hub.starlight.nssc-gz.cn/nssc	构建成功	1.2GB	私有	在原镜像做了相关配置和修改	2020-03-09 14:39:06	[启动] [停止] [删除] [刷新]
slurm_cluster:0.1	hub.starlight.nssc-gz.cn/nssc	构建成功	959.0MB	私有	slurm_cluster	2020-03-09 10:11:58	[启动] [停止] [删除] [刷新]
pangeo:0.9	hub.starlight.nssc-gz.cn/nssc	构建成功	5.8GB	公开	pangeo	2020-02-13 16:00:01	[启动] [停止] [删除] [刷新]

3.2.1.3 运行 HPC 应用

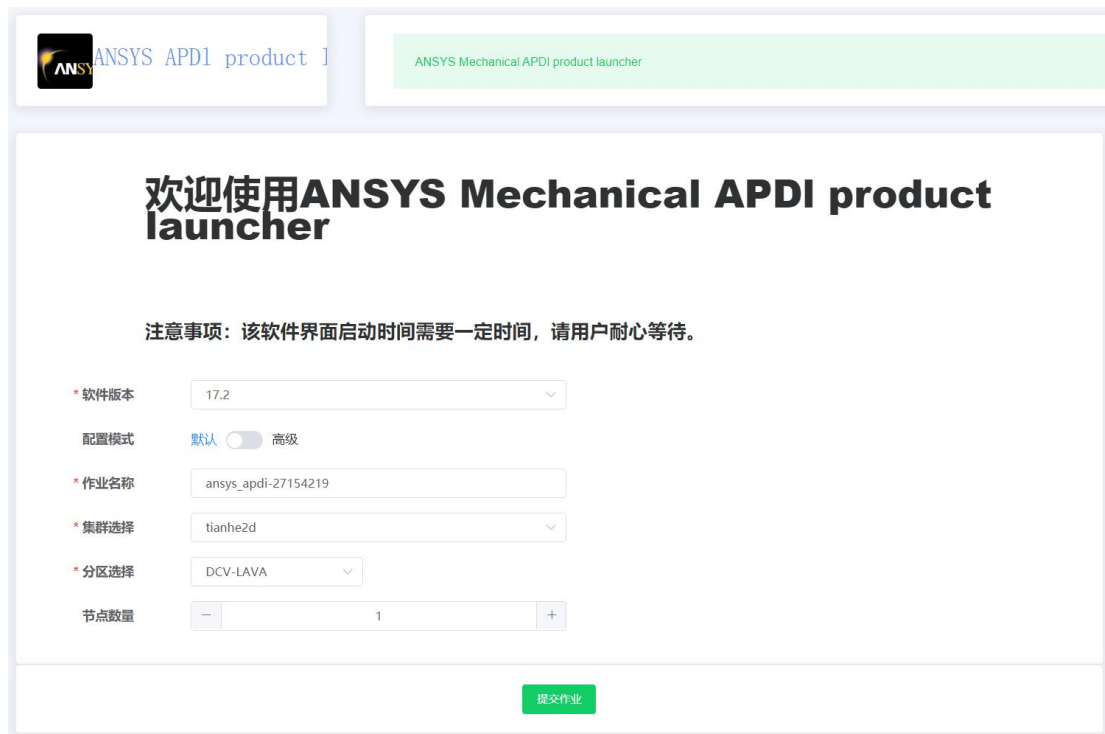
在“应用列表”中，不带蓝鲸标志的应用表示该应用为普通 HPC 应用，本节以 HPC 应用“ANSYS APDI product launcher”为例，演示 HPC 应用的运行步骤，更多的应用运行说明见：[常见应用使用说明](#)。

运行“ANSYS APDI product launcher”应用步骤：

1. 点击应用列表中的“ANSYS APDI product launcher”图标，进入应用运行配置表单页面：



2. 配置应用的运行环境信息：

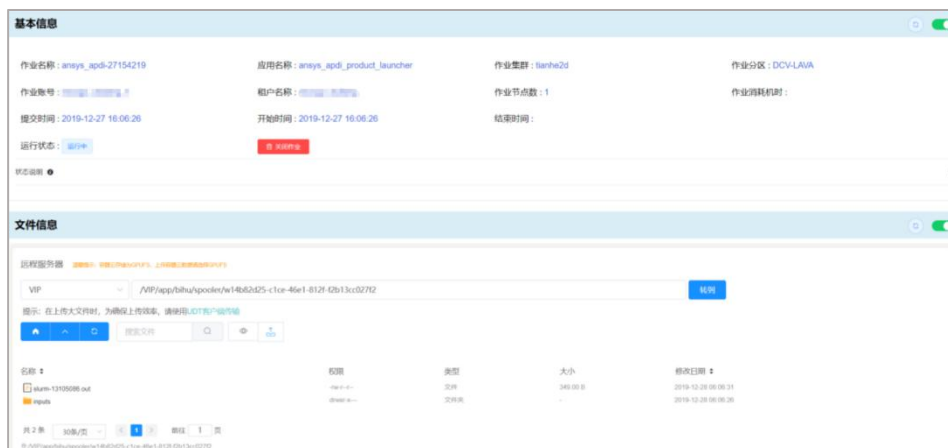


上图为默认的配置模式，各个主要的配置项说明如下：

- (1) 软件版本：选择相应的软件版本
- (2) 作业名称：提交的作业（在星光平台中，一个运行中的应用一般称之为作业）的名称，默认系统会自动生成一个
- (3) 集群选择：选择作业要运行的目标 Slurm 集群
- (4) 分区选择：选择作业要运行的目标 Slurm 集群分区
- (5) 作业配置：选择运行作业所需的节点数

系统会根据当前用户的配额和机时判断用户是否可以提交该作业。

3. 点击“提交作业”按钮，提交作业，系统会自动跳转到提交的作业详情界面：



关于作业详情的更多信息，请参考：[4.2 节](#)。

3.2.2 使用高级配置

在提交容器应用作业时，可以选择高级配置，来指定容器的挂载存储、环境变量等参数：

The screenshot shows the Tensorflow application configuration interface. At the top, there is a 'Tensorflow' logo and a 'tensorflow应用' tab. Below this, there is a '作业提示' section with instructions. The main configuration area includes:

- 配置模式:** Radio buttons for '默认' (selected) and '高级'. A red arrow points to the '高级' button with the text '点击使用高级配置模式'.
- * 作业名称:** Input field containing 'tensorflow-13175533'.
- 镜像选择:** Dropdown menu showing 'tensorflow:centos'.
- * 集群选择:** Radio buttons for 'k8s_venus' (selected) and '启明集群'.
- * 分区选择:** Radio buttons for 'venus-gpu 启明集群GPU分区' (selected), 'venus-cpu 启明集群CPU分区', and '24核/4块/240GB'.
- * 作业配置:** Radio buttons for '6核/1块/60GB' (selected), '12核/2块/120GB', and '24核/4块/240GB'. A red arrow points to this section with the text '高级配置项'.

Below these options is a red-bordered box containing advanced settings:

- 用户模式:** Dropdown menu with '缺省: 请选择用户模式'.
- 工作目录:** Input field with '点击设置工作目录' and a right arrow.
- 镜像挂载存储:** Input field with '点击设置镜像挂载存储' and a right arrow.
- 环境变量设置:** Input field with '点击设置环境环境变量' and a right arrow.
- 镜像启动入口:** Input field with '点击设置镜像启动入口' and a right arrow.
- 访问入口设置:** Input field with '点击设置访问入口' and a right arrow.

At the bottom right, there is a green '提交作业' button.

3.2.2.1 设置镜像挂载存储

设置镜像挂载存储的步骤如下图所示：

配置模式 默认 高级

* 作业名称

镜像选择

* 集群选择 kds_venus 启明集群

* 分区选择 venus-gpu venus-cpu

* 作业配置 6核/60GB 12核/120GB 16核/160GB 1核/10GB

镜像挂载存储 ② [点击设置镜像挂载存储](#)

存储名称	系统目录	容器内目录	操作
user-vol0	/GPUFS:.....mp	/volume1	移除

[添加存储映射](#)

环境变量设置 ② [点击设置环境变量](#)

镜像启动入口 [点击设置镜像启动入口](#)

访问入口设置 ② [点击设置访问入口](#)

[提交作业](#)

设置完成后提交运行作业，进入运行的作业中可以看到挂载的存储：

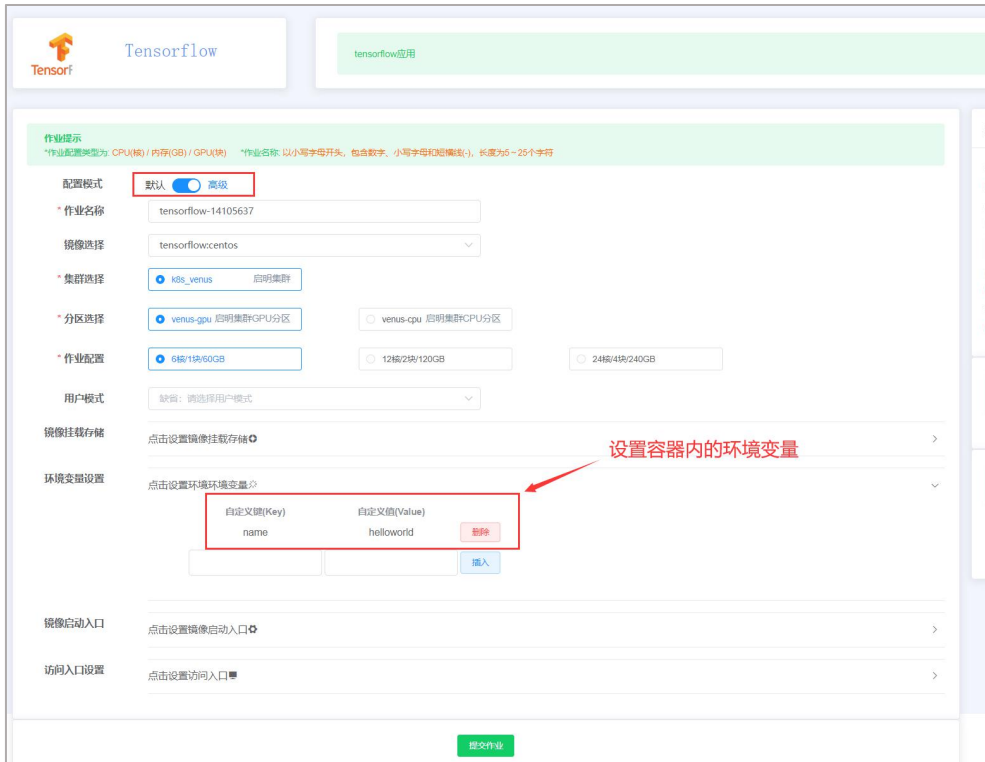
```

← → ↻ 🏠 ⚠️ 不安全 | 24206.proxy.nscg-gz.cn:8888
bash-4.2$ mount |grep volume
10.20.10.100@o2ib4:10.20.10.101@o2ib4:/L2 on /volume1 type lustre (rw,nosuid,localflock,lazystatfs)
bash-4.2$

```

3.2.2.2 设置环境变量

环境变量的设置和镜像挂载存储的设置类似，如下：



提交完作业可以进入到作业中查看设置的环境变量：



可以看到容器中已经成功设置了该环境变量。

四、 作业管理

作业管理用于管理历史和正在运行的作业。

4.2.2 访问入口

“访问入口”提供访问资源的入口。对于深度学习/大数据作业，系统会默认为作业创建若干个访问入口，如下为深度学习应用“deeplearning”默认的作业访问入口：



对于可视化作业，系统默认会为作业创建一个访问入口用于访问作业的图形界面，如下为可视化应用“Centos6 Desktop”默认的作业访问入口：



4.2.2.1 浏览器远程登录作业

用户可以通过作业的访问入口对用户创建的作业进行访问。除 SSH 协议的访问入口(显示“SSH 连接信息”)外，其它类型的访问入口使用鼠标点击相应的访问入口链接即可在浏览器上访问作业。

以下深度学习应用“deeplearning”系统默认创建的访问入口“ttyd-user”为例，首先点击相应的访问入口：



浏览器会弹出一个新的标签页，以用户的身份登录到该作业的终端上，可以在该终端上执行常规的 Shell 命令：

```
← → ↻ 🏠 不安全 | 24211.proxy.nscg-gz.cn:8888
bash-4.2$ ls
aaaa AQI_shiny Centos7.iso core.177915 dask-worker-space dataset example git log nodebook R rstudio scripts tmp tools
bash-4.2$
```

对于 SSH 协议的访问说明请参考下一节。

4.2.2.2 SSH 远程登录作业

用户可以通过常用的 SSH 工具如 Putty、MobaXterm、XShell 等来登录作业，各个工具的下载链接如下：

Putty: <https://en.softonic.com/download/putty/windows/post-download>

MobaXterm: <https://mobaxterm.mobatek.net/download-home-edition.html>

XShell: <https://www.netsarang.com/zh/xshell-download/>

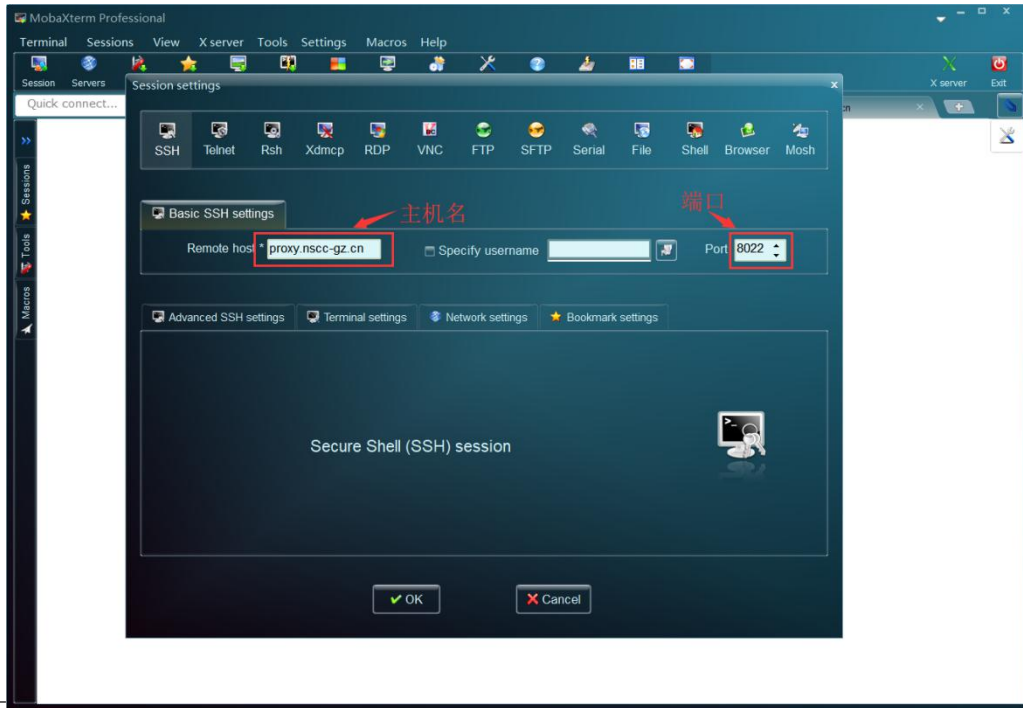
以下以 MobaXterm 和 XShell 工具为例。

使用 MobaXterm 登录作业

首先，在作业详情的访问入口查询远程登录的主机名和端口：



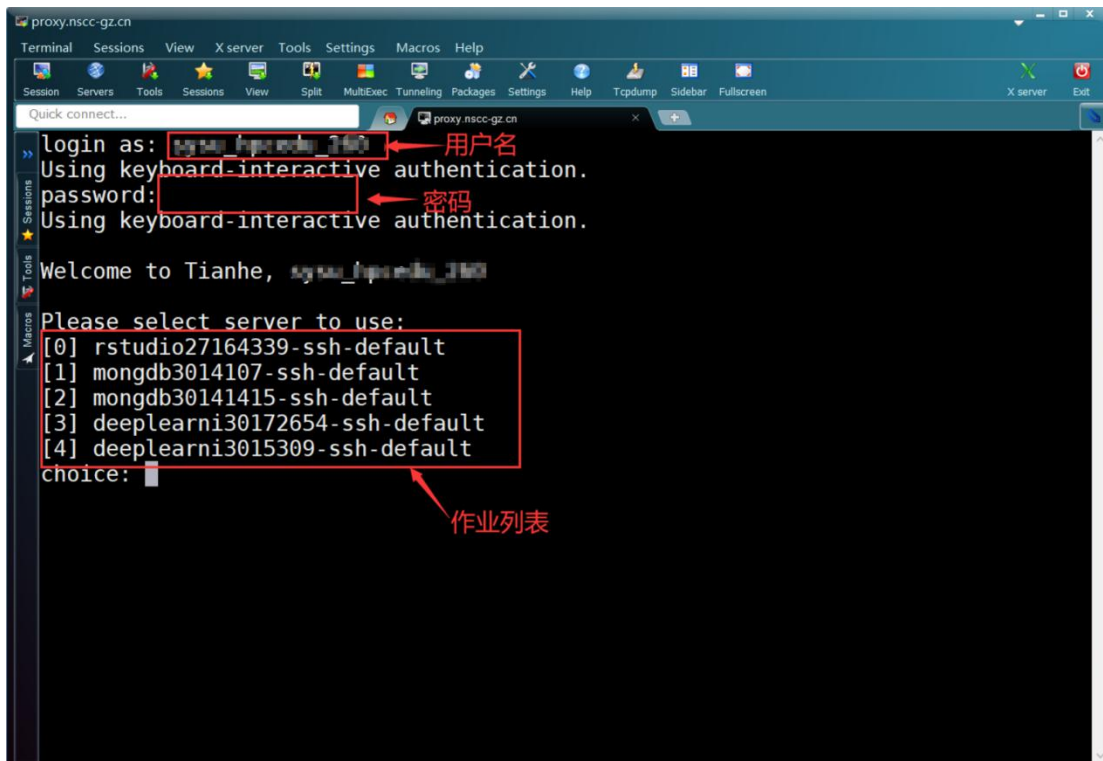
打开 MobaXterm 工具，将主机名和端口信息输入进去：



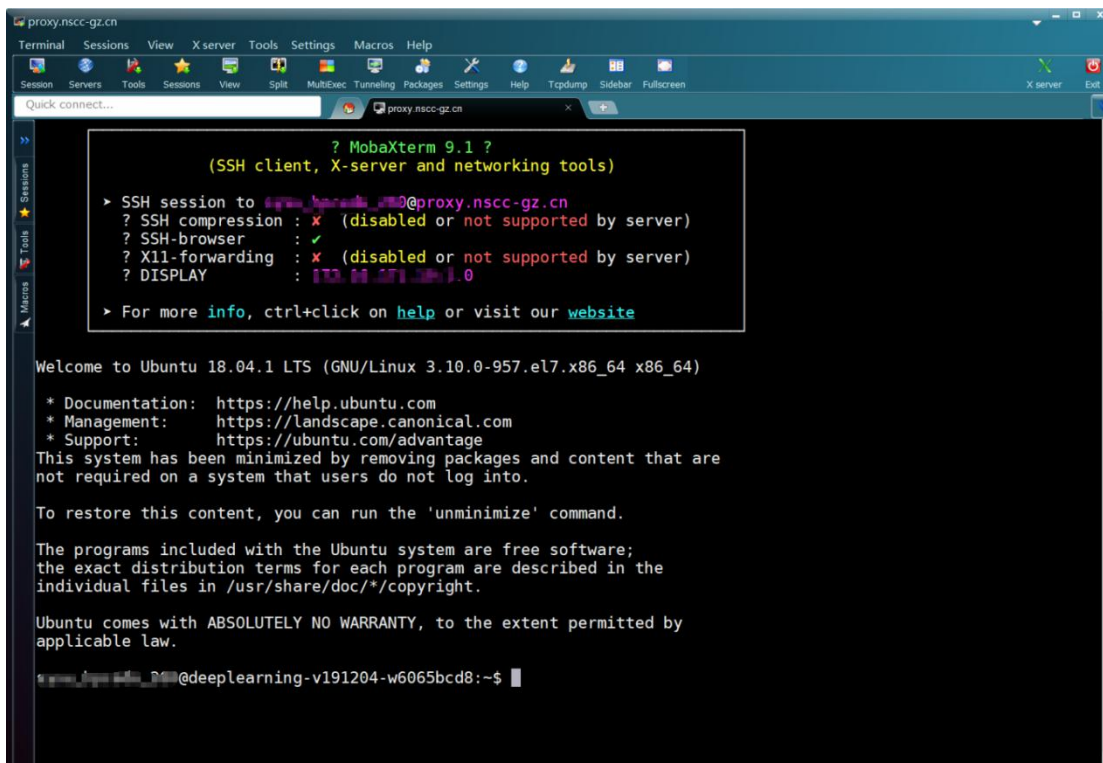
点击“OK”按钮确定后，会要求输入要登录的用户名和密码，用户名使用[作业详情-基本信息](#)中的作业账号（如下），密码为[星光平台的登录密码](#)：



将作业账号和登录密码输入终端，用户密码验证通过后，会显示正在运行的作业列表：



选择要登录的作业编号按回车即可登录到作业运行的容器里:

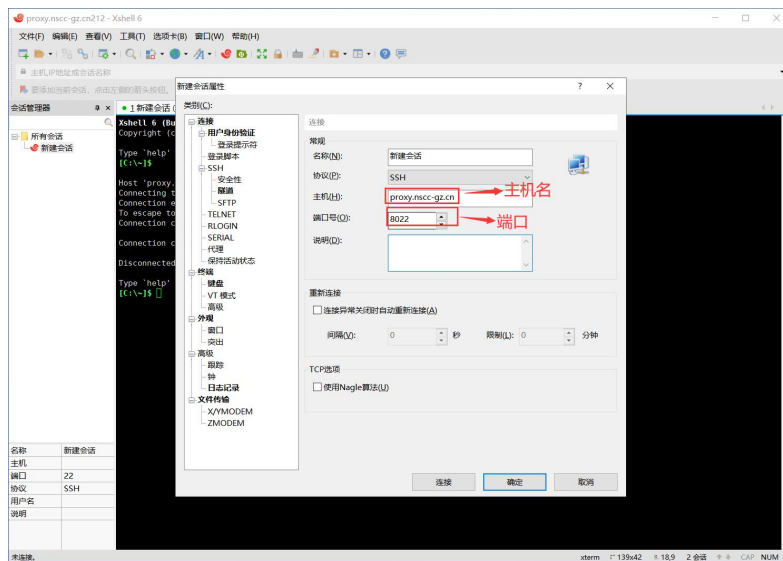


使用 XShell 登录作业

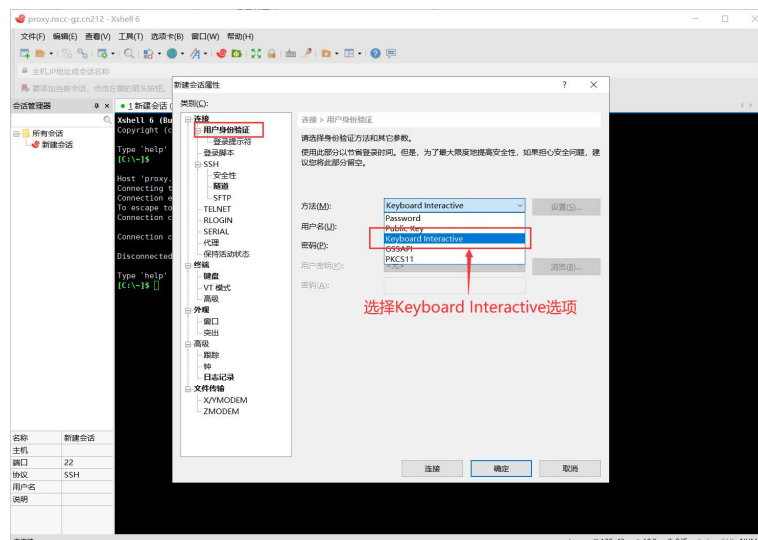
首先，在作业详情的访问入口查询远程登录的主机名和端口:



打开 XShell 工具，将主机名和端口信息输入进去：



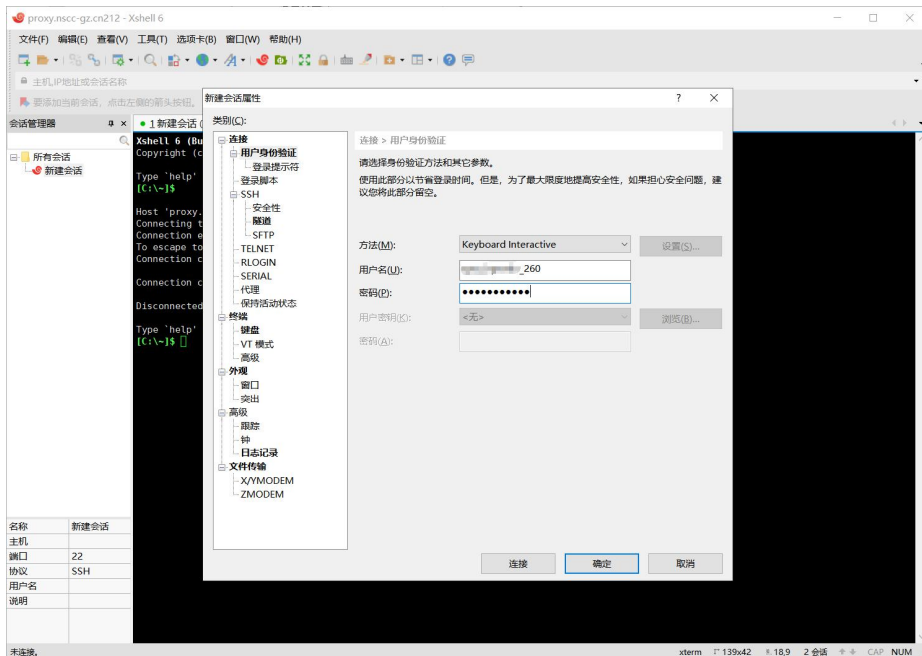
点击右侧菜单栏的“用户身份验证”选项，方法选项选择“Keyboard Interactive”：



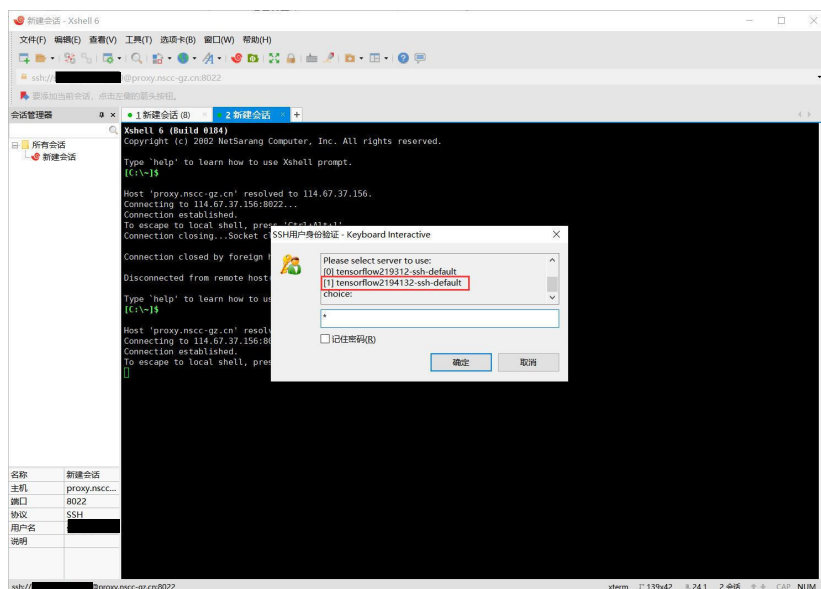
用户名使用[作业详情-基本信息](#)中的作业账号（如下），密码为星光平台的登录密码：



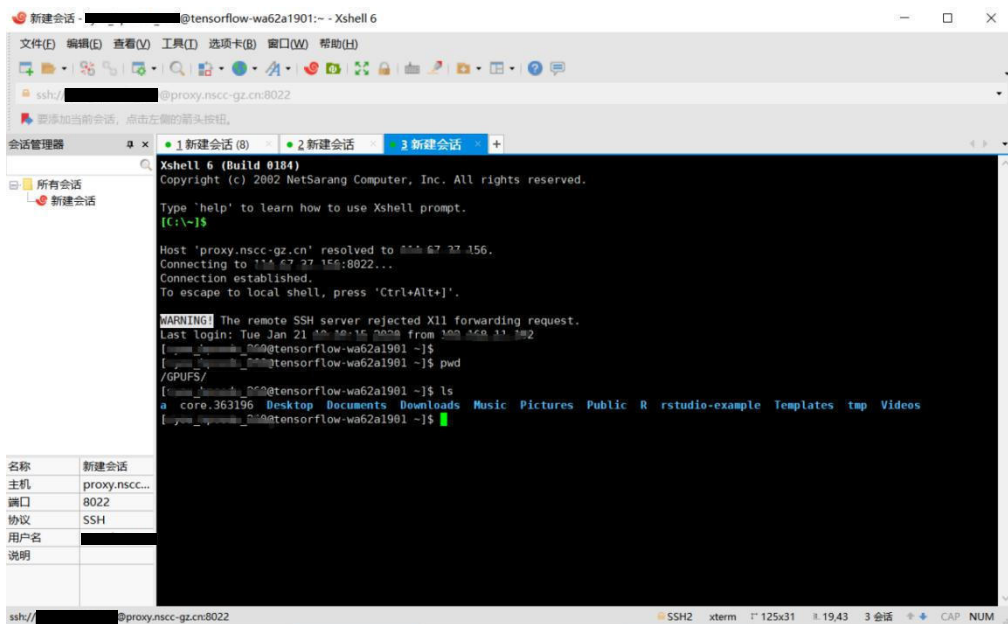
将用户名和密码信息输入相应的输入框:



用户密码验证通过后，若用户有多个正在运行中的作业，会显示正在运行的作业列表，在输入框中输入要登录的作业编号即可进入作业中；若用户只有一个运行中的作业，则会直接登录到该作业中：

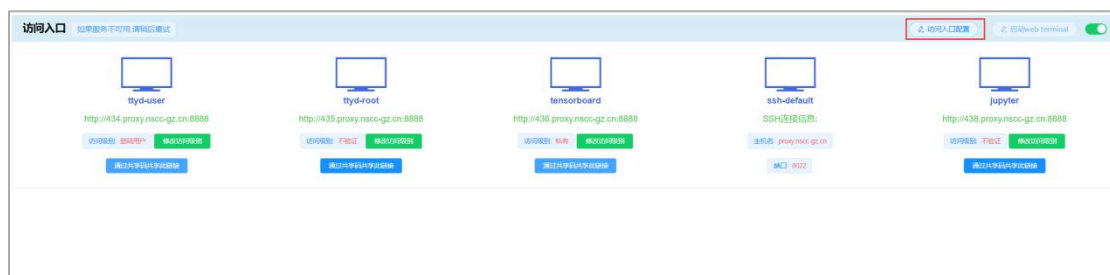


出现如下界面则表明登录作业成功:



4.2.2.3 添加访问入口

要添加新的访问入口，可以点击访问入口右上角的“访问入口配置”按钮：



在弹出的“修改访问入口”页面点击添加按钮：



在弹出的“添加访问入口”页面填写入口相关信息：



添加访问入口

名称: http1

协议: http 使用非协议标准端口 (如选中, 将从32768-65535之间随机选择一个作为外部访问端口)

自定义域名: 建议不填写 注: 如填写, 请使用您已注册备案过的域名, 并将您的域名映射到IP[114.67.37.156].

验证级别: 私有

目的端口: 8888 注: 容器内部服务所监听的端口

取消 确定

各项信息说明如下：

- 名称：访问入口的名称，取个容易记的即可，注意不要和已有访问入口重复
- 协议：代理入口使用的协议，可选 http、https、ssh 和 tcp。每个协议会有默认的固定端口，如 http 代理入口的默认端口一般为 8888，勾选“使用非协议标准端口”可以让系统随机选择一个端口，而不是使用默认的
- 自定义域名：该项仅在选择 http 和 https 协议时出现，填写一个已注册备案过的域名，如没有，可不写
- 证书文件：该项仅在选择 https 协议时出现，用于上传 https 证书文件
- 密钥文件：该项仅在选择 https 协议时出现，用于上传 https 密钥文件
- 验证级别：生成的代理链接的验证级别，可选的级别有：私有、登录用户、不验证，各级别的区别如下：
 - 私有：该级别的访问入口只有作业运行者本人可以使用，但可以通过共享码的方式共享给其他人，点击“通过共享码共享此链接”，可以生成共享码。点击“复制共享码”，将共享码分享给需要的人，对方可以在浏览器中通过共享码访问分享的资源。点击“取消共享”，取消共享码，其他用户将不能再使用共享码登录入口。
 - 登录用户：该级别的访问入口所有的星光用户在登录后，都可以访问
 - 不验证：该级别的访问入口不需要验证，任何人都可以访问
- 目的端口：填写要访问的容器内部服务所监听的端口

填写完相关信息后点击确定即可创建新的访问入口。

以下以为容器中运行的 nginx 服务创建访问入口为例，演示添加访问入口的步骤：

1. 首先，在容器中查看 nginx 服务监听的端口：

```
root@deeplearning-v191204-wd06cf42b-sbin]# netstat -nap |grep nginx
tcp        0      0 0.0.0.0:80          0.0.0.0:*        LISTEN     2711/nginx: master
unix 3      [ ]          STREAM  CONNECTED      89609008 2711/nginx: master
unix 3      [ ]          STREAM  CONNECTED      89609007 2711/nginx: master
```

2. 点击“作业详情”界面的访问入口右上方的“访问入口配置”：



3. 点击弹出界面左下方的“添加”按钮：



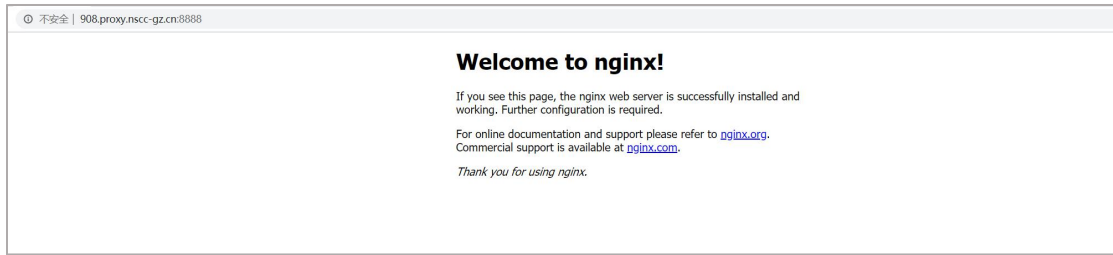
4. 按如下方式填写各项信息，注意这里的目的端口填的是步骤 1 中查出来的 nginx 服务监听的端口



5. 点击“确定”按钮后，再点击“确定”按钮，可以看到访问入口中多了一个名为“nginx”的访问入口：



6. 点击名为“nginx”的访问入口，即可访问容器中的 nginx 服务：



4.2.2.4 删除访问入口

点击访问入口右上角的“访问入口配置”按钮：



点击要删除的访问入口项后面的移除按钮后，再点击确认即可删除相关访问入口：

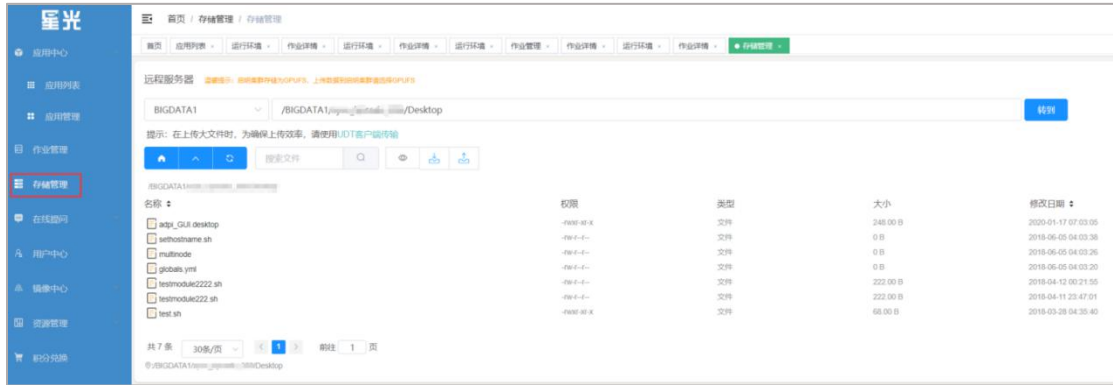


五、 存储管理

存储管理支持用户对具有访问权限的存储进行文件操作，支持在线上传、下载、复制、移动、删除等操作。

5.1 传统方式

点击导航栏的“存储管理”选项，进入到存储管理界面。若安装了 UDT 客户端，会进入到使用 UDT 的存储管理界面（见下节），若未安装 UDT 客户端，则进入到传统方式的存储管理界面，如下：



5.1.1 界面简介

传统方式的存储管理界面显示了远程服务器上的文件信息，各部分信息如下：



该界面还支持对文件和目录进行一些常用操作，如下所示：

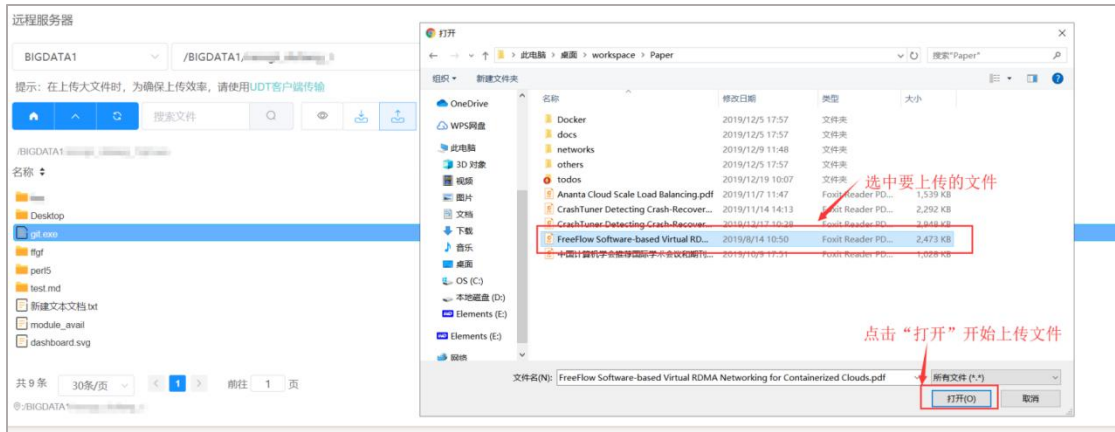


5.1.2 上传文件

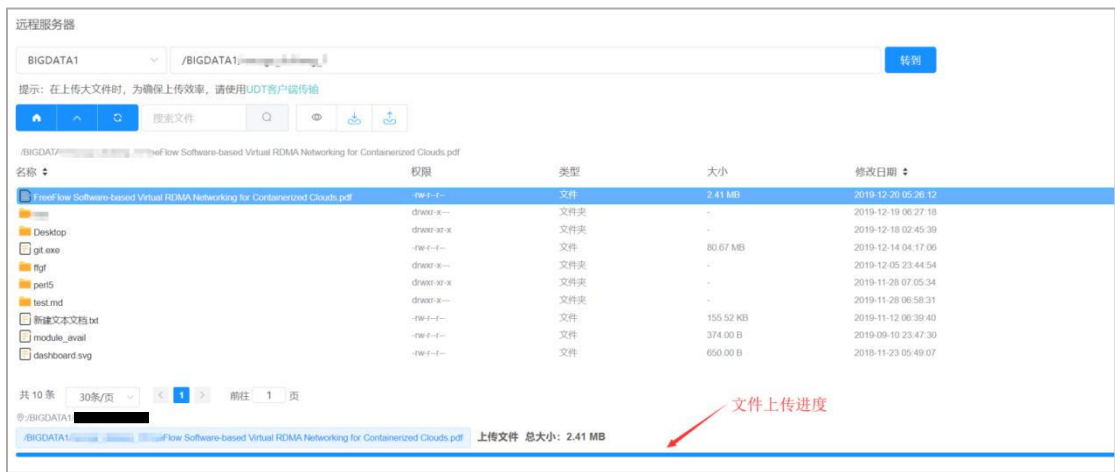
要上传文件到当前目录，点击上传文件按钮，如下图所示：



点击上传文件按钮后，会弹出一个文件选择框，如下图所示：



点击文件选择框的“打开”按钮开始文件上传后，在存储管理界面下方会显示文件上传的进度，上传完成后即可在当前目录下看到该文件：

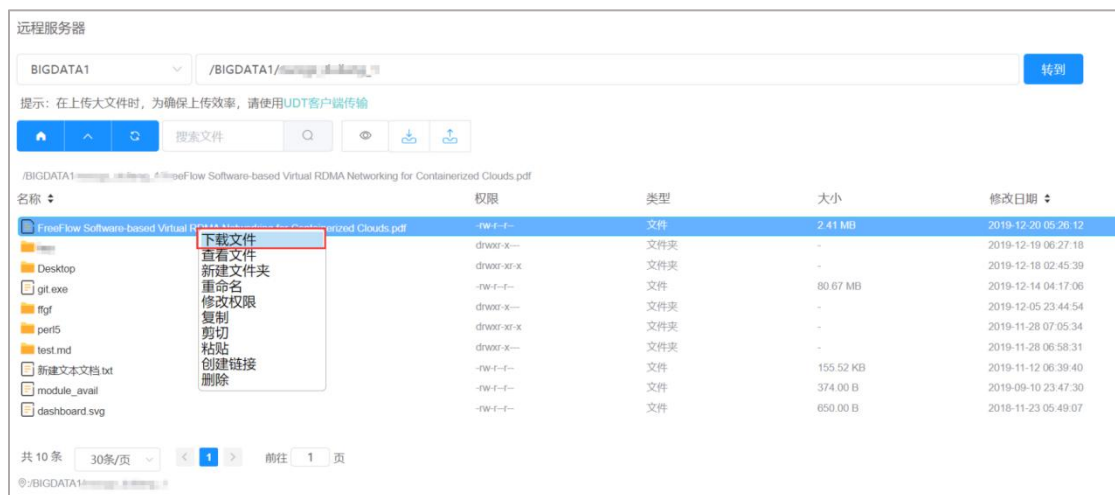


5.1.3 下载文件

点击选中要下载的文件后，界面上会出现一个下载按钮，点击该按钮即可将该文件下载下来：



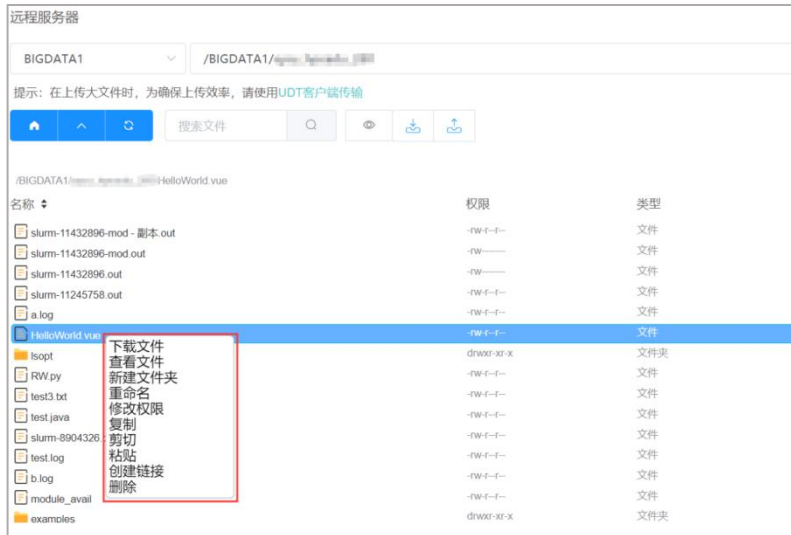
还有另外一种下载方法, 在选中要下载的文件后, 点击鼠标右键弹出操作菜单, 再点击“下载文件”同样可以把文件下载下来:



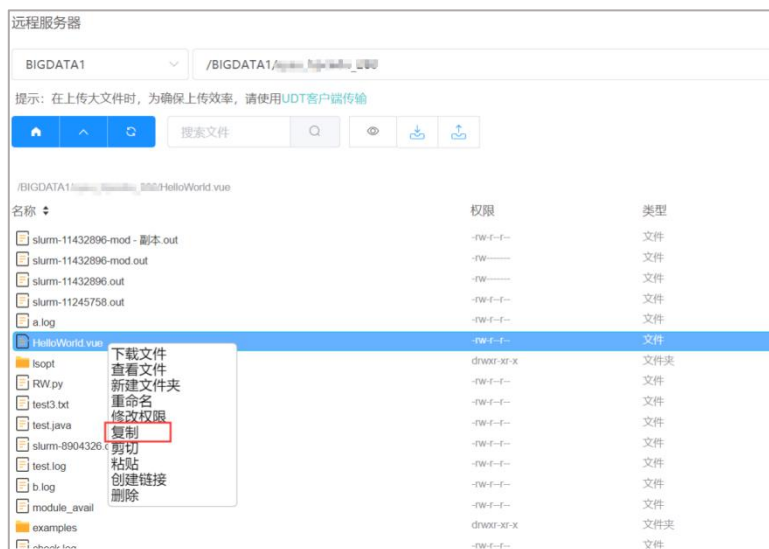
5.1.4 复制文件

存储管理支持将文件或文件夹复制到同个文件系统下的其它目录, 也支持跨文件系统的文件或文件夹复制。以下演示跨文件系统复制文件的步骤, 文件夹的复制和文件复制的步骤基本一致, 同文件系统的复制忽略步骤 3, 只需要选择同个文件系统的目录作为目标位置即可。

1. 选中要复制的文件, 右击弹出操作菜单:



2. 点击操作菜单的“复制”选项:



3. 切换到目标文件系统:



4. 进入目标位置所在的目录:



5. 右击弹出操作菜单, 点击“粘贴”选项:



6. 输入目标文件名, 点击“确定”按钮:



7. 完成复制:



5.1 UDT 方式

UDT 是一种基于 UDP 的数据传输协议，用于支持高速的海量数据传输，在上传大文件时，UDT 方式会有更高的效率。

5.1.1 切换方法

通过点击下图所示的“UDT 客户端传输”，可从传统方式切换到 UDT 方式的界面：



使用 UDT 方式进行数据传输时，需要在本地运行 UDT 客户端。第一次使用 UDT 方式时，需要先下载安装 UDT 客户端，点击如下“UDT 客户端工具”链接，下载 UDT 客户端工具：

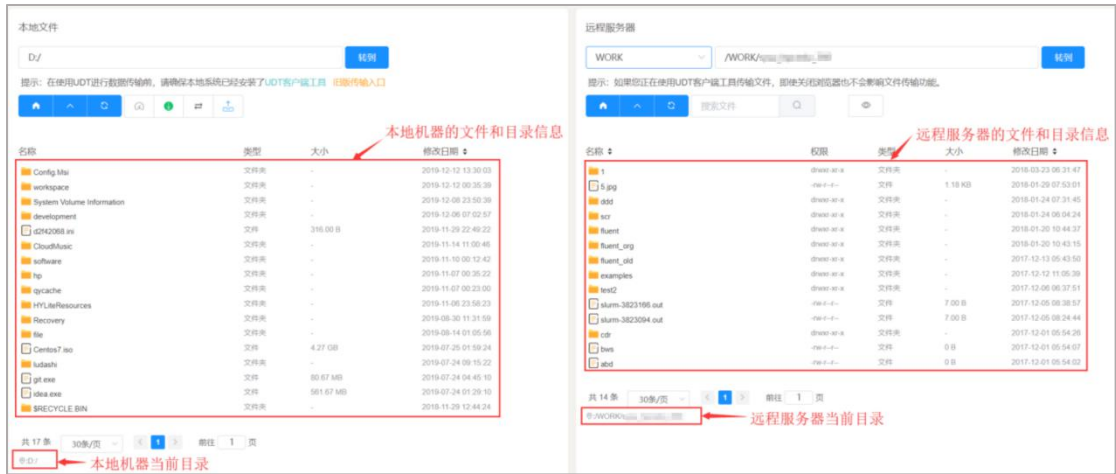


下载得到的是一个 zip 格式的压缩包，解压后得到一个可执行文件：“星光网络传输客户端.exe”，执行该可执行文件进行安装并运行后，再刷新一下界面，若此时本地文件列表中可以看到本地机器上的文件及目录信息（如下图），则表明 UDT 客户端运行成功，接下来可以使用 UDT 进行文件传输：

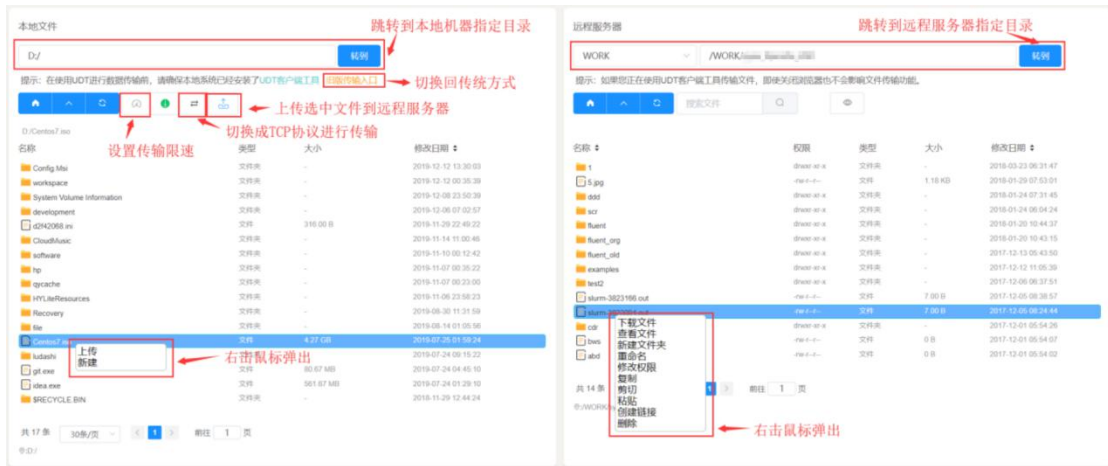


5.1.2 界面简介

UDT 存储管理界面和传统方式的管理界面布局有所不同，左侧区域为用户本地机器上的文件及目录信息，右侧区域为远程服务器上的文件及目录信息：



各个主要操作如下图所示：

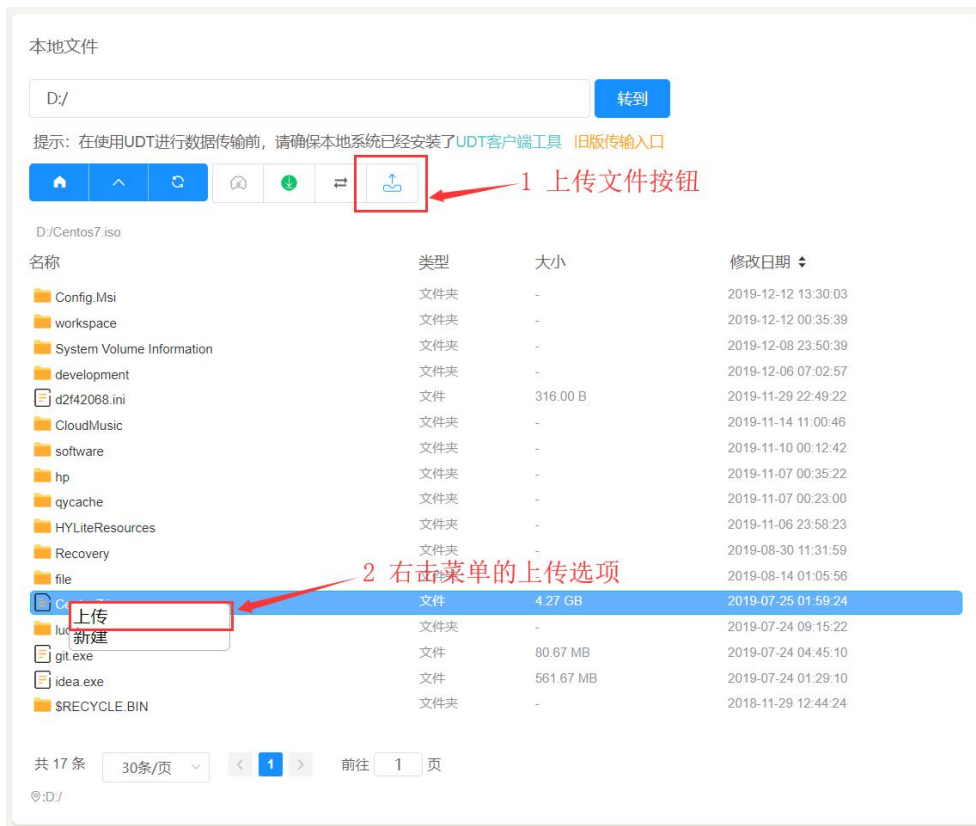


5.1.3 上传文件

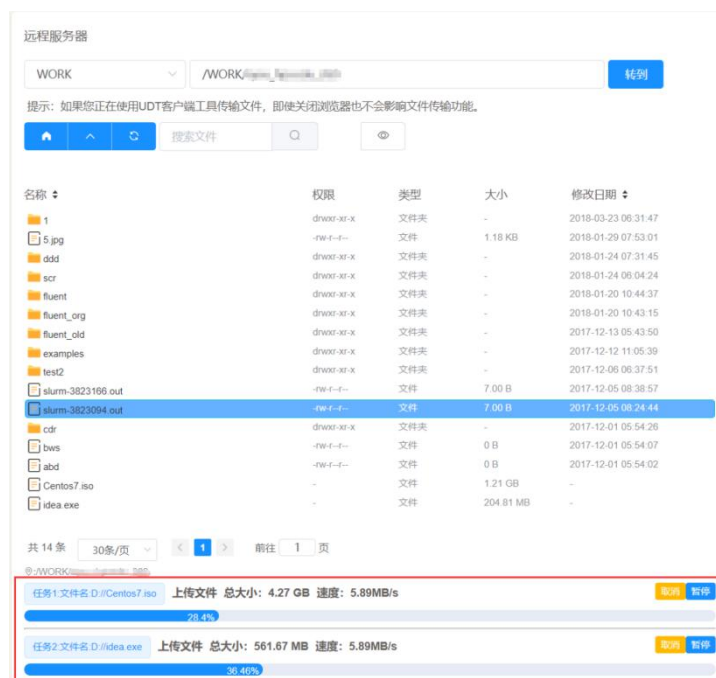
上传文件有两种方式：

1. 点击选中本地机器上要上传的文件，点击上方的文件上传按钮
2. 点击选中本地机器上要上传的文件，右击鼠标，点击弹出框的“上传”选项

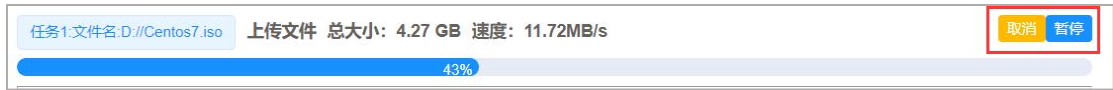
如下图所示：



上传文件时, 在远程服务器界面下方会显示文件上传进度, 并且可以同时上传多个文件:



点击文件传输进度条右上角的“暂停”按钮可以暂停文件传输, 点击“取消”按钮可以取消文件传输:



暂停文件传输后点击文件传输进度条右上角的“恢复”按钮可以继续文件的传输：

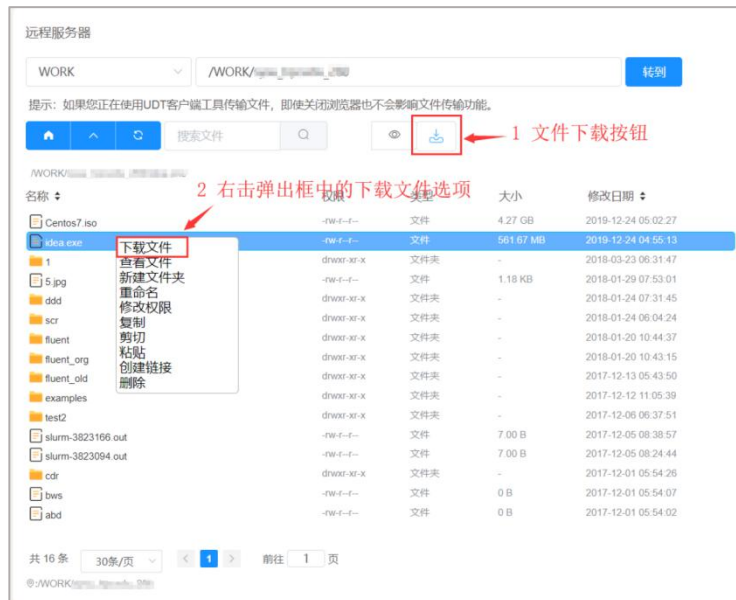


5.1.4 下载文件

下载文件有两种方式：

1. 点击选中远程服务器上要下载下来的文件，点击上方的文件下载按钮
2. 点击选中远程服务器上要下载下来的文件，右击鼠标，点击弹出框的“下载”选项

如下图所示：



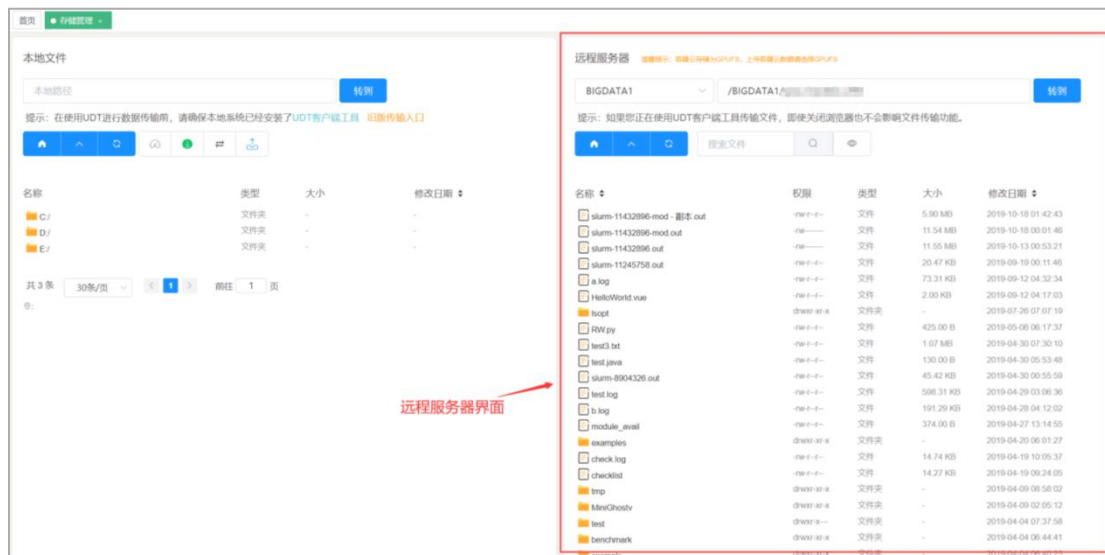
下载文件时，在远程服务器界面下方会显示文件下载进度，并且可以同时下载多个文件：



和上传文件类似，点击文件传输进度条右上角的“暂停”按钮可以暂停文件传输，点击“取消”按钮可以取消文件传输，详情参考上一小节。

5.1.5 复制文件

UDT 方式的存储管理的文件复制方式和传统方式基本一致，仅“远程服务器”子界面的位置改到界面右侧，如下：



复制操作在远程服务器界面中完成，详见：[5.1.4](#)

5.1.6 切换 TCP

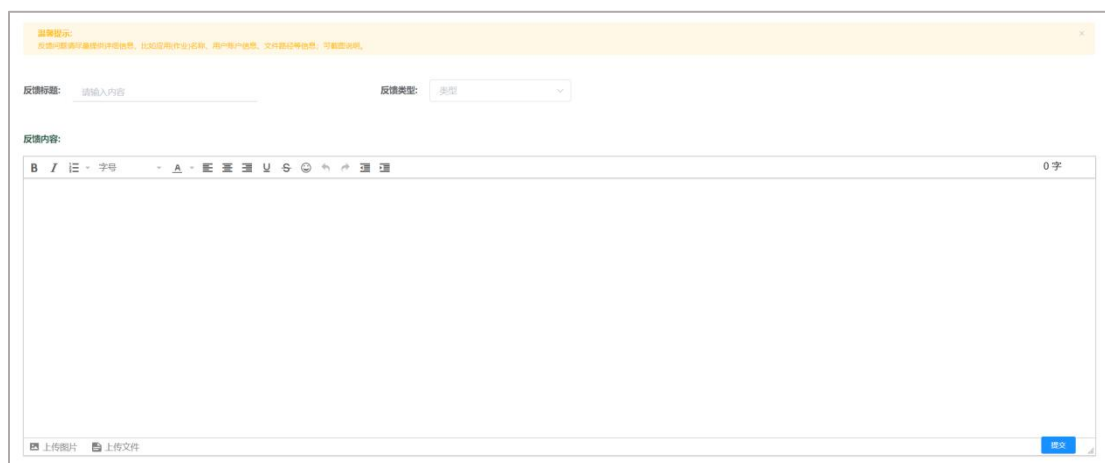
UDT 客户端默认使用 UDP 和服务器之间进行数据传输，但由于某些原因（如防火墙）导致 UDP 无法使用或者很慢时，可以将 UDT 客户端切换成使用 TCP 协议进行数据传输：



六、在线提问

6.1 反馈问题

若用户在使用天河星光平台中出现任何问题，均可以在平台中的“在线提问”->“反馈问题”页面进行问题反馈或寻求技术人员的支持。问题内容的编辑支持 Markdown 语法。建议用户在提交问题时，尽量详细地描述问题发生的情景，以及尽量提供问题现场截图贴图以便技术支持人员可以更快的定位和排查问题。



温馨提示：
反馈内容将直接公开给其他用户，且会应用在日志系统、用户帐户系统、文件附件系统等，可能记录在案。

反馈标题: 反馈类型:

反馈内容:

B / 字号 - A - 背景 列表 链接 插入 删除 0字

6.2 我的问题

历史提问信息可在“在线提问”->“我的问题”中查看（如下图），如果有来自管理员的回复，可以看到提示信息。



我的问题 公共问题

问题数量 编辑进度 创建时间 更新时间 编辑

问题标题	问题类型	编辑进度	创建时间	更新时间	查看详情
some questions	系统bug	已解决	2019-12-23 17:38	2019-12-23 17:38:21	<input type="button" value="详情"/> <input type="button" value="回复"/> <input type="button" value="结束"/>

共 1 条 20条/页 < 1 > 前往 1 页

公共问题显示了平台上所有公开的问题：



我的问题 公共问题

全文搜索

标题	内容	时间
test	123 天河星光4.0数据库设计.vsdx	2019-12-23 17:41
请问请问	问起我	2019-12-23 17:40
1321	123	2019-11-18 18:25

共 3 条 20条/页 < 1 > 前往 1 页

七、用户中心

用户中心展现用户的基本资料，包括用户个人账号信息和邮箱、手机号码等信息，用户可以在“用户中心”页进行用户个人信息的修改，如登录密码的修改。建议绑定邮箱和手机，以便获得更完整的平台功能。

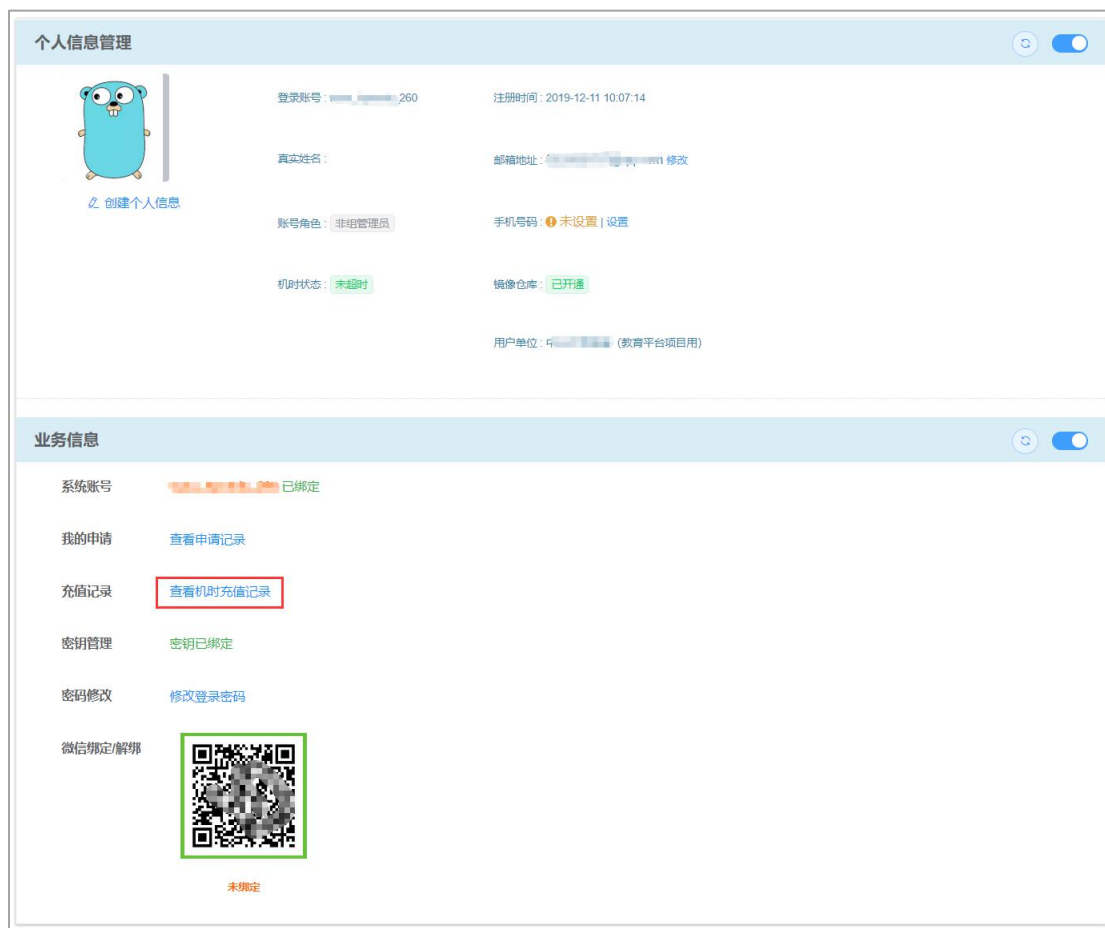


7.1 查看申请记录

记录用户分区开通、配额变更等申请的审批进度和审批记录。用户可点击“查看申请记录”，查看正在进行中和已结束的申请记录：

7.2 查看机时充值记录

查看用户的机时充值记录。用户可点击“查看机时充值记录”查看历史充值记录：



用户机时充值记录：

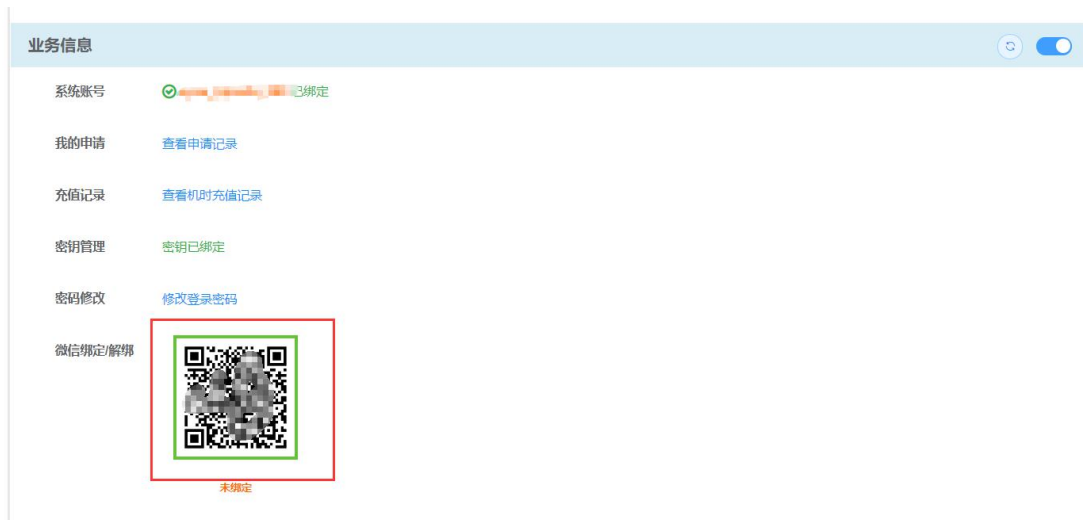


7.3 绑定微信

用户可将星光账号与微信账号绑定，绑定后可通过微信扫描二维码的方式登录星光平台。

绑定步骤：

1. 点击导航栏的“用户中心”，使用微信扫描以下二维码：



2. 在微信界面中点击关注公众号即可完成绑定。

绑定后在星光平台登录界面，点击微信登录可进入到微信二维码登录界面：

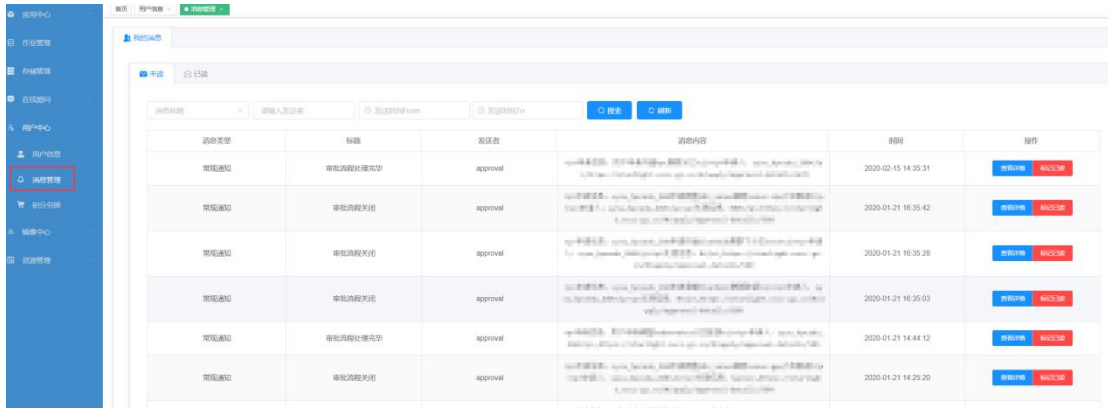


微信二维码登录界面如下，通过微信扫描二维码即可登录星光平台：

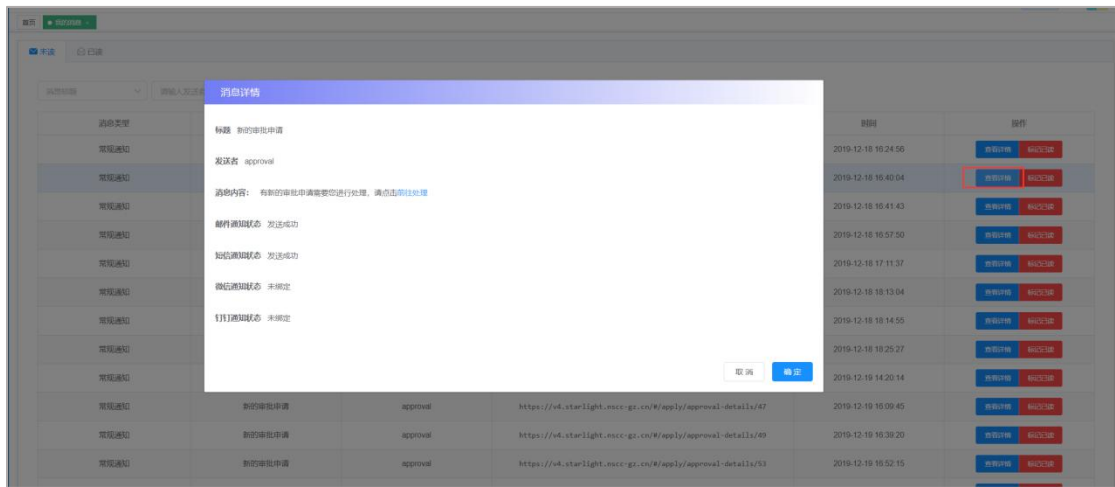


7.4 消息管理

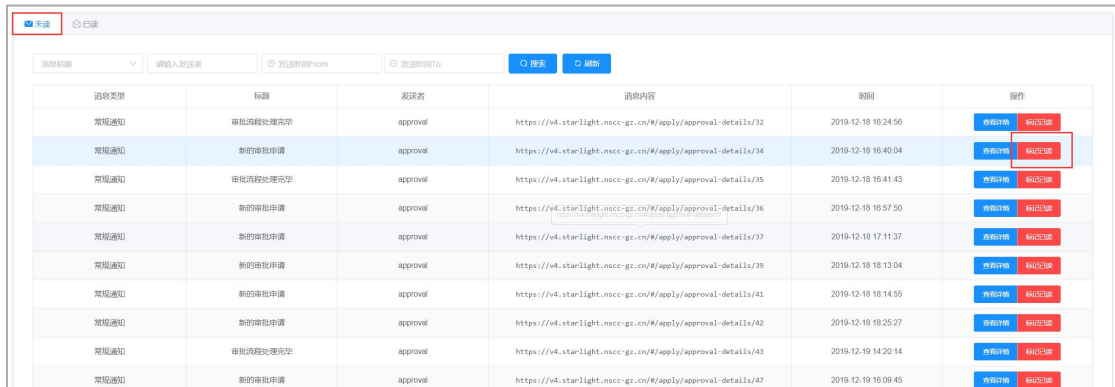
点击导航栏的“消息管理”选项，可以显示收到的消息列表：



点击消息后面的“查看详情”按钮，可以查看消息的详情：



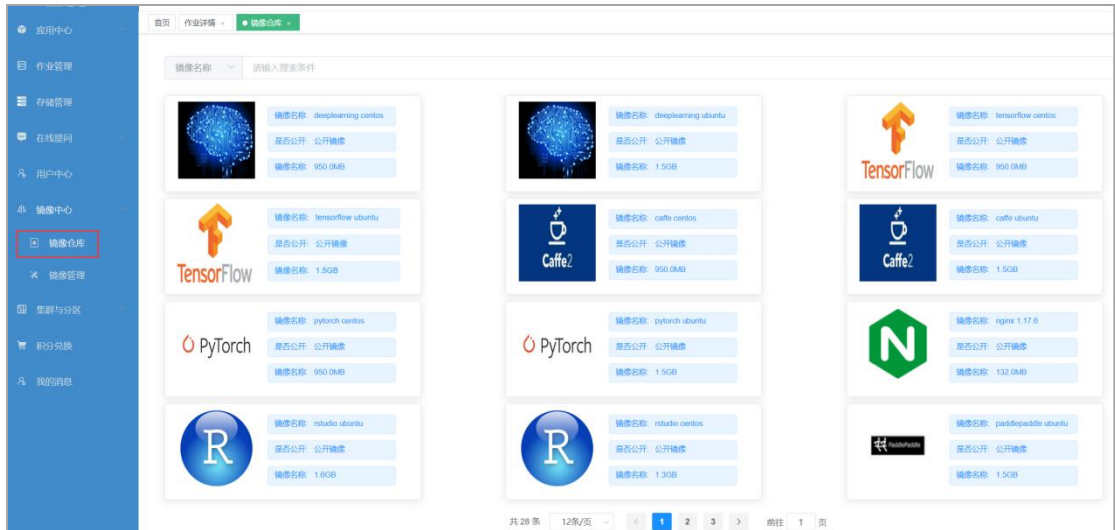
消息查看完可以点击“标记已读”按钮，把该消息移到已读消息中去：



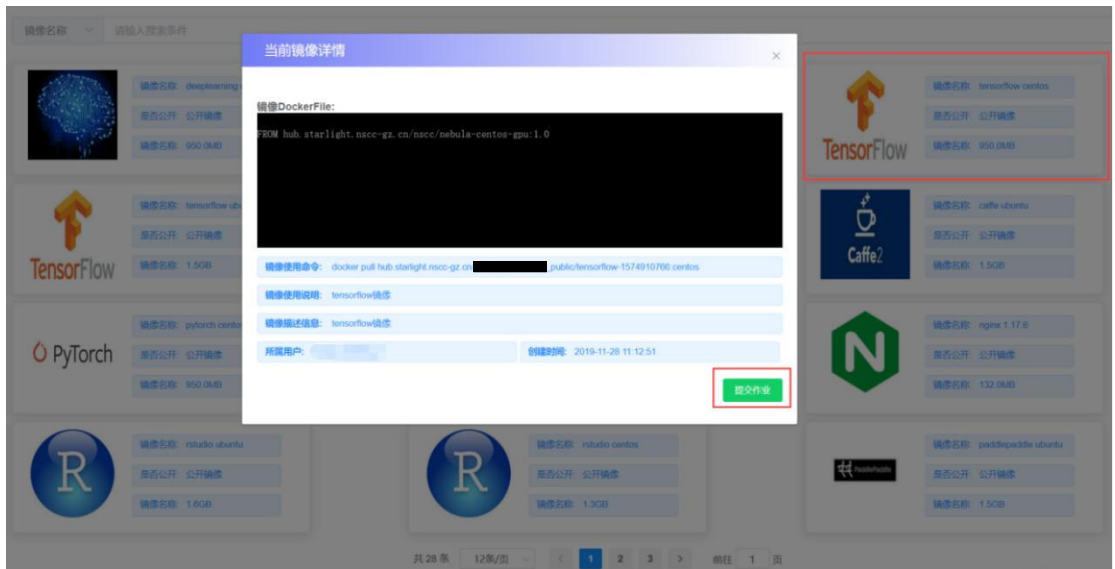
八、镜像中心

8.1 镜像仓库

镜像仓库显示了用户可用 Docker 镜像列表：



点击镜像图标可查看镜像详情，点击详情界面右下角的“提交作业”按钮可以将镜像提交到集群中运行：



8.2 镜像管理

镜像管理界面用于对镜像进行管理：



列出的镜像包括了镜像的一些信息，如状态、大小、创建时间等，最后面的操作包括了5个操作：修改镜像图标、查看镜像构建脚本、查看镜像日志，删除镜像和提交作业：



8.2.1 镜像配额

用户镜像配额指用户可以创建的镜像个数和创建的镜像可以占用的存储容量：

用户名称	项目名称	可用个数	已用个数	可用容量(GB)	已用容量(GB)	更新时间	审批状态	操作
admin	private	49	1	49.71	0.29	2019-12-31 15:12:41	可申请	申请调整
admin	public	49	1	49.71	0.29	2019-12-31 15:12:48	可申请	申请调整

用户可以点击“申请调整”按钮来申请更多的镜像配额，进入申请调整配额的申请页面后，填写想要申请的存储配额和数量配额，点击“提交”即可提交申请，申请需要管理员通过确认，请耐心等待：

申请调整镜像仓库配额

用户申请调整镜像配额

审批名称	审批负责人	审批状态	审批意见
镜像配额申请环节一	admin	审批中	

* 镜像存储配额(GB)

* 镜像数量配额

* 镜像仓库类型

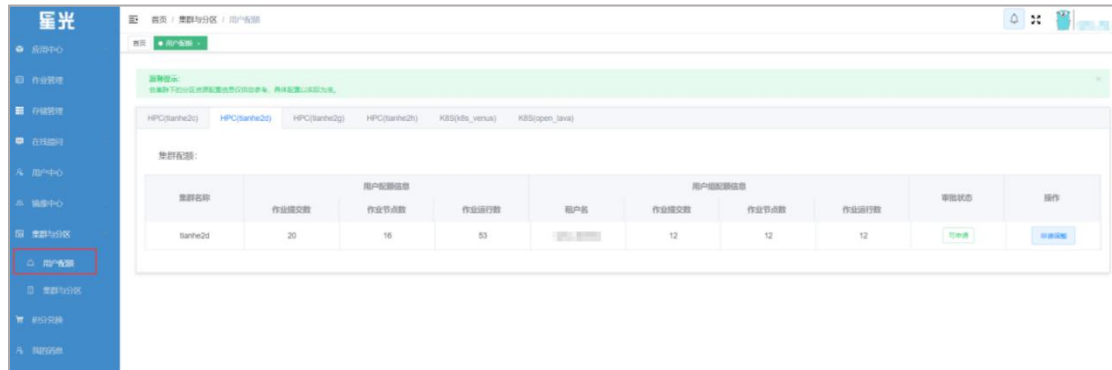
租户镜像配额限制了所有用户的可用配额总数，如下，该租户可用镜像数是 50，包括了用户私有镜像和公有镜像：

用户组	可用存储(GB)	已用存储(GB)	可用镜像	已用镜像
admin	48.11	1.87	50	2

九、 集群与分区

9.1 用户配额

点击导航栏“集群与分区”->“用户配额”，可以查看用户和用户组的配额信息：



配额信息包括：

- 1) 作业提交数：用户能够提交的最大作业数量，包括运行中和排队中的作业数量，不包括已结束的作业数量
- 2) 作业节点数：用户最多能够使用的最大节点数量
- 3) 作业运行数：用户能够同时运行（即处于“运行中”状态）的作业数量

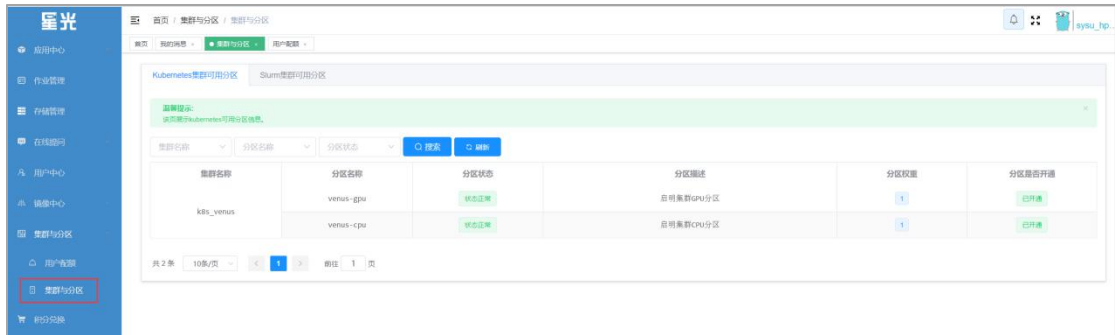
每个用户的配额、机时等资源，不仅受用户自身的资源限制，也不能超过用户组的资源限制。同一用户组下的所有用户使用的资源总量不能超过用户组的配额。

如需调整某个集群的用户配额，点击该集群后面的“申请调整”按钮，即可进入申请配额调整：

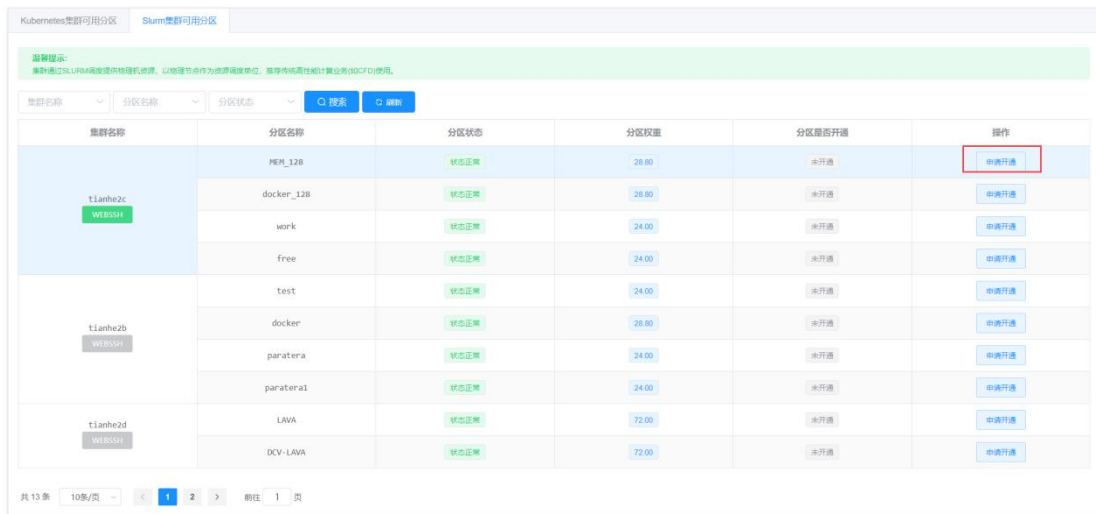


9.2 集群与分区

展示用户可以连接的高性能计算集群和容器云(k8s)平台分区。用户在具备使用权限的集群上可以看到可用的分区资源：

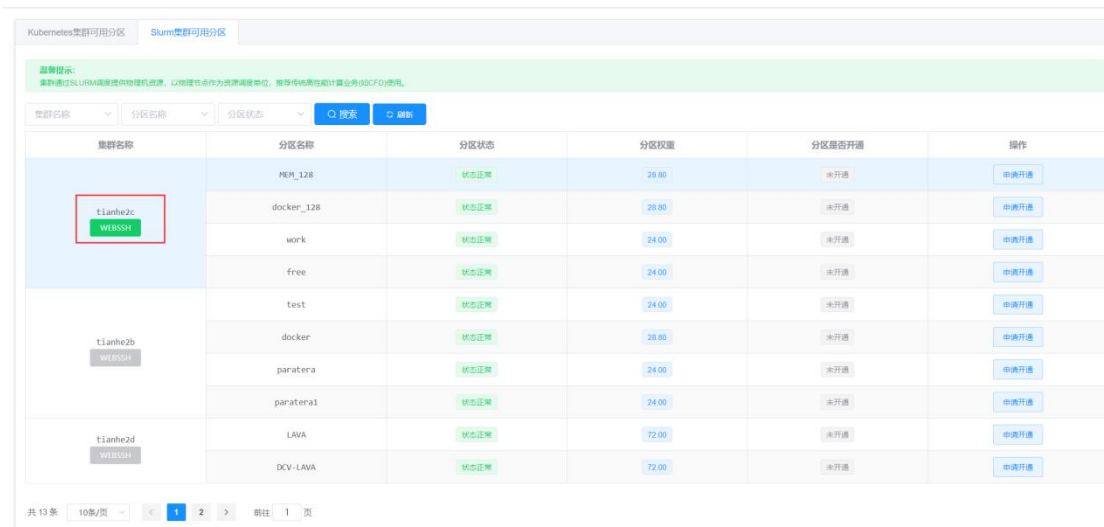


对于没有开通的分区，用户可以点击“申请开通”按钮提交分区开通的申请，管理员审批通过后则可以使用相应的资源分区：

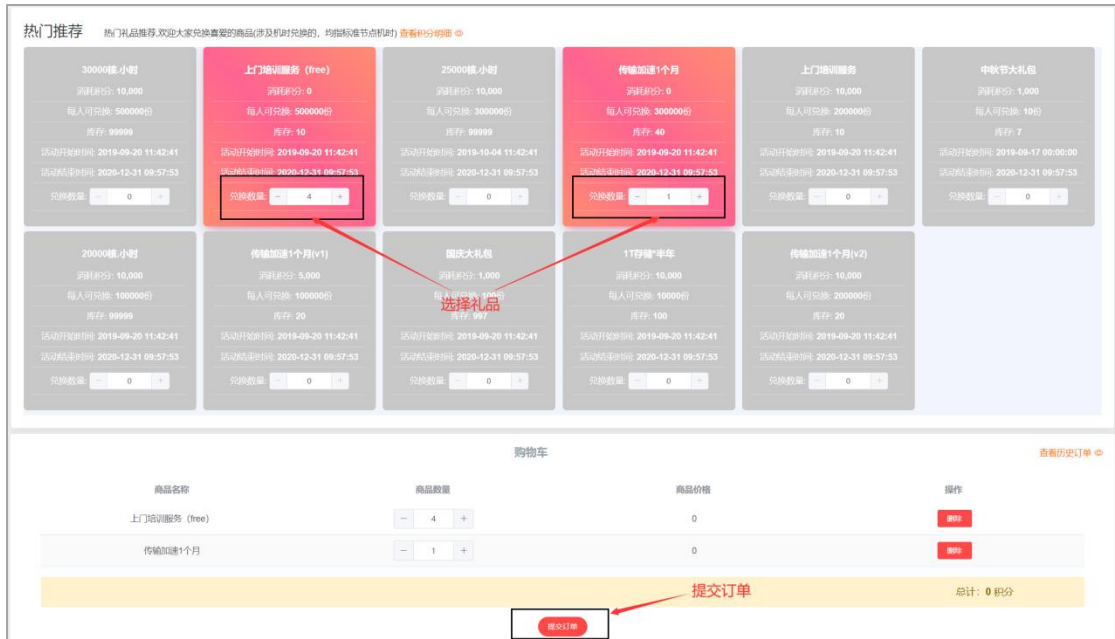


9.2.1 通过 WebSSH 登录 Slurm 集群

在 Slurm 集群名称下面显示“WEBSSH”按钮的集群支持直接通过浏览器登录到 Slurm 集群：



点击“WEBSSH”按钮后即可登录到该集群：



提交后点击购物车右上角的“查看历史订单”按钮，可以查看订单状态：



订单需要经过平台管理员审批通过后才生效，请耐心等待。

十一、深度学习/大数据应用使用说明

本节主要说明用户所创建的大数据、深度学习应用如何进行远程登录、连接互联网、安装需要的软件、进行数据的上传下载等。

目前上线的有 Rstudio 等大数据分析工具，Tensorflow、Caffe、CNTK、Torch、Mxnet、PaddlePaddle 等深度学习框架、Galaxy 生物分析平台等，更多的应用也正在不断的发布和完善过程中。

11.1 加载平台已安装软件

星光平台预安装了许多软件包供用户应用加载使用，并且还在不断完善中。这些应用使用 module 进行管理。module 是集群的软件环境管理工具，通过 module 相关命令，可以将集群已经安装好的软件环境加载到当前用户环境里。

11.1.1 查看已安装软件列表

应用可以通过 module avail 命令查看平台已安装的软件列表：

```
[root@deeplearning-v191204-wefd60bcf ~]# module avail
anaconda2/5.3.0  anaconda3/5.3.0  apr/1.6.5  cuda/9.0  cudnn/7.5-v6.0  cudnn/8.0-v7.1  dcatk/3.6.4  gcc/8.3.0  java/1.8.0_191  maven/3.6.0  proxy/1.0.0  sox/14.4.2  sqlite/3.25.3  vfstools/1.0.0
anaconda3/4.2.0  ant/1.10.5  cjson/1.7.9  cuda/10.0  cudnn/8.0-v5.1  cudnn/9.0-v7.1  ffmpeg/4.0.2  gcc/9.2.0  libconfuse/3.2.2  openssl/1.7.5  R/3.2.2  spark/2.3.2  subversion/1.11.0
anaconda3/5.0.0  apr-util/1.6.3  cmake/3.5.2  cuda/8.0  cudnn/7.5-v5.1  cudnn/8.0-v6.0  cudnn/10.0-v7.3  gcc/5.4.0  gradle/4.10.2  lua/5.3.5  openssl/1.10.7  scala/2.10.7  spark/2.3.2-alluxio  torch/7
```

注：使用 ubuntu 基础镜像的应用若 module 命令不可用，可以尝试执行 source /etc/profile.d/module.sh 加载一下 module 工具。

11.1.2 加载/卸载已安装软件

应用可以通过 module load/unload 来加载/卸载平台已安装的软件，如下图以加载/卸载 java 为例：

```
[root@deeplearning-v191204-wefd60bcf ~]# java -version
bash: java: command not found
[root@deeplearning-v191204-wefd60bcf ~]# module load java/1.8.0_191
[root@deeplearning-v191204-wefd60bcf ~]# java -version
java version "1.8.0_191"
Java(TM) SE Runtime Environment (build 1.8.0_191-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.191-b12, mixed mode)
[root@deeplearning-v191204-wefd60bcf ~]# module unload java/1.8.0_191
[root@deeplearning-v191204-wefd60bcf ~]# java -version
bash: java: command not found
```

未加载java, 命令不可用

加载java后, 命令可正常使用

卸载java后, 命令不可用

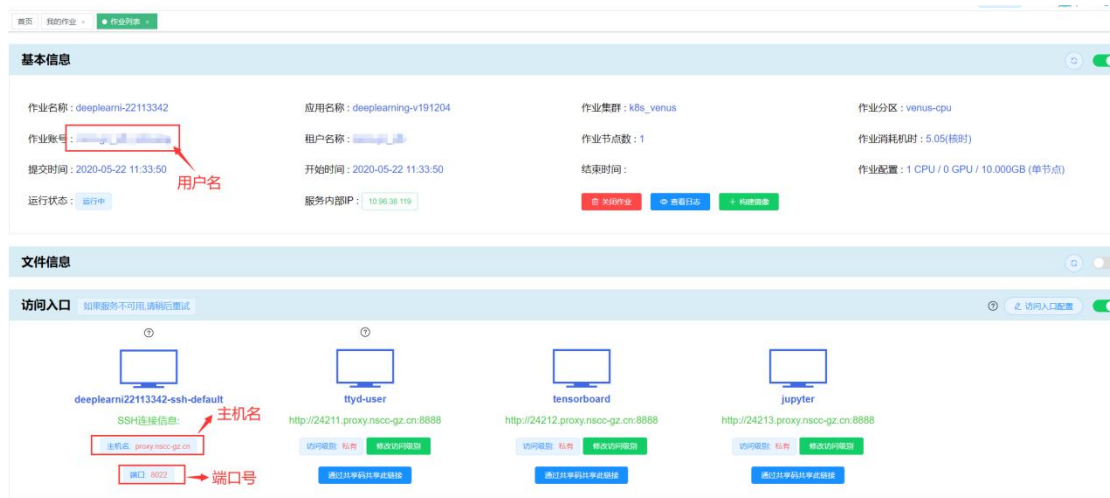
11.2 网络代理设置

用户登录作业之后，如果需要访问互联网，需设置访问代理。可通过以下两种方式进行配置。

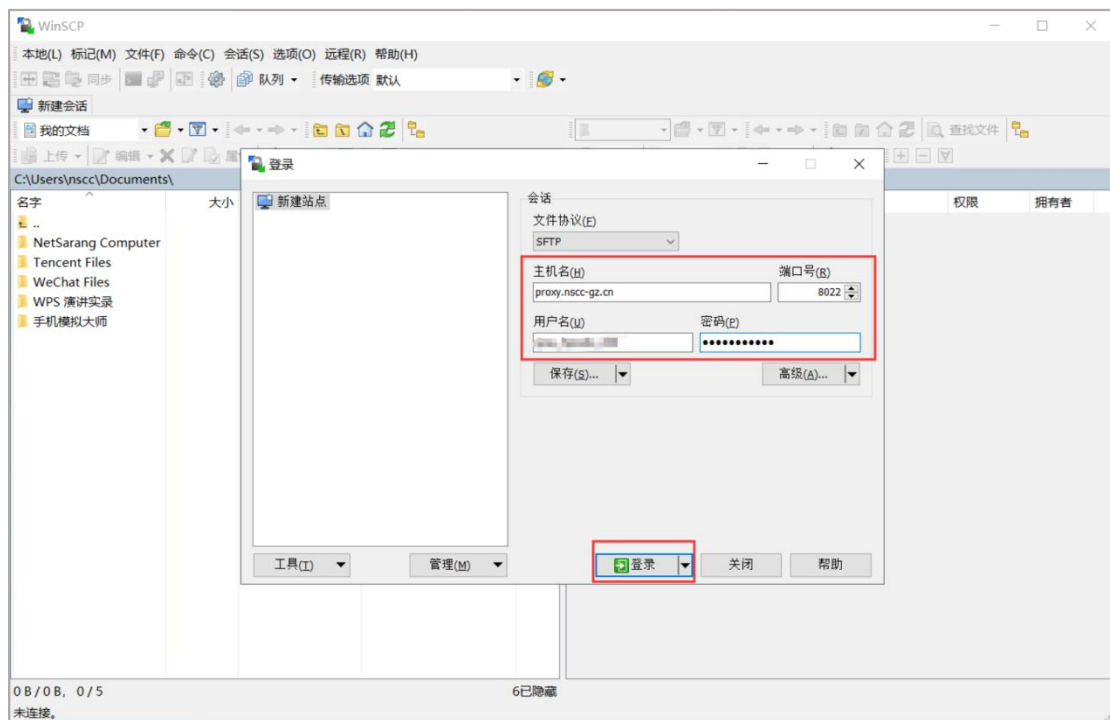
11.2.1 通过 module

在终端界面中输入命令：module load proxy/1.0.0 即可访问互联网，如下图所示：

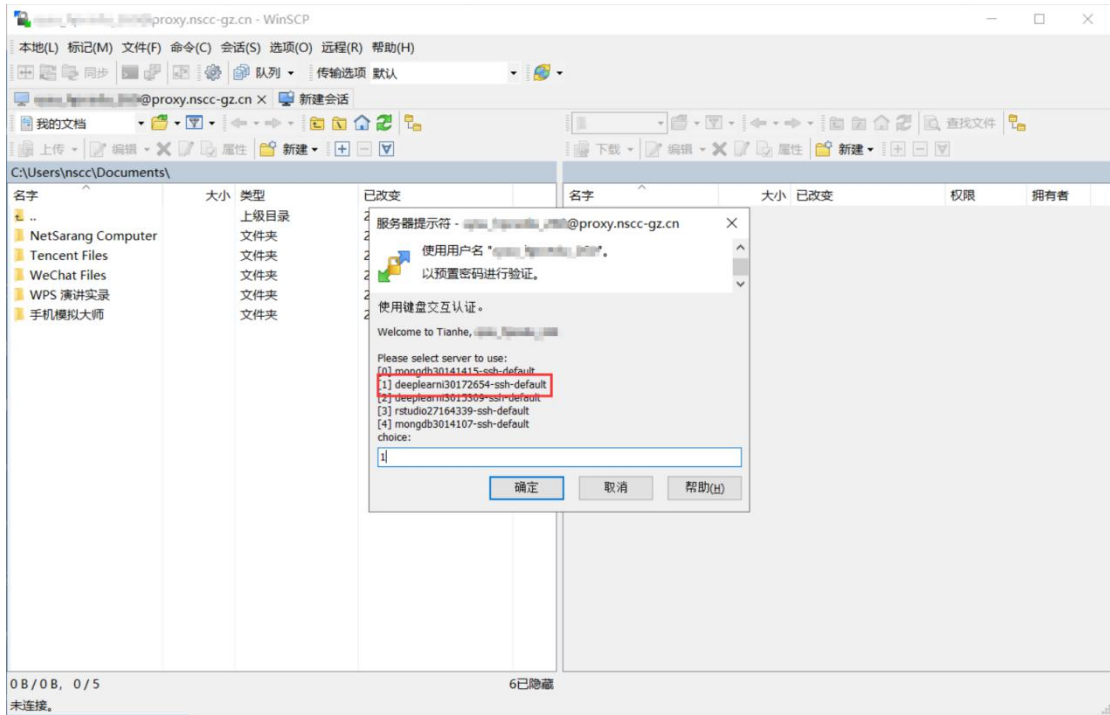
1) 在作业详情界面查看作业的主机名、端口号和用户名信息：



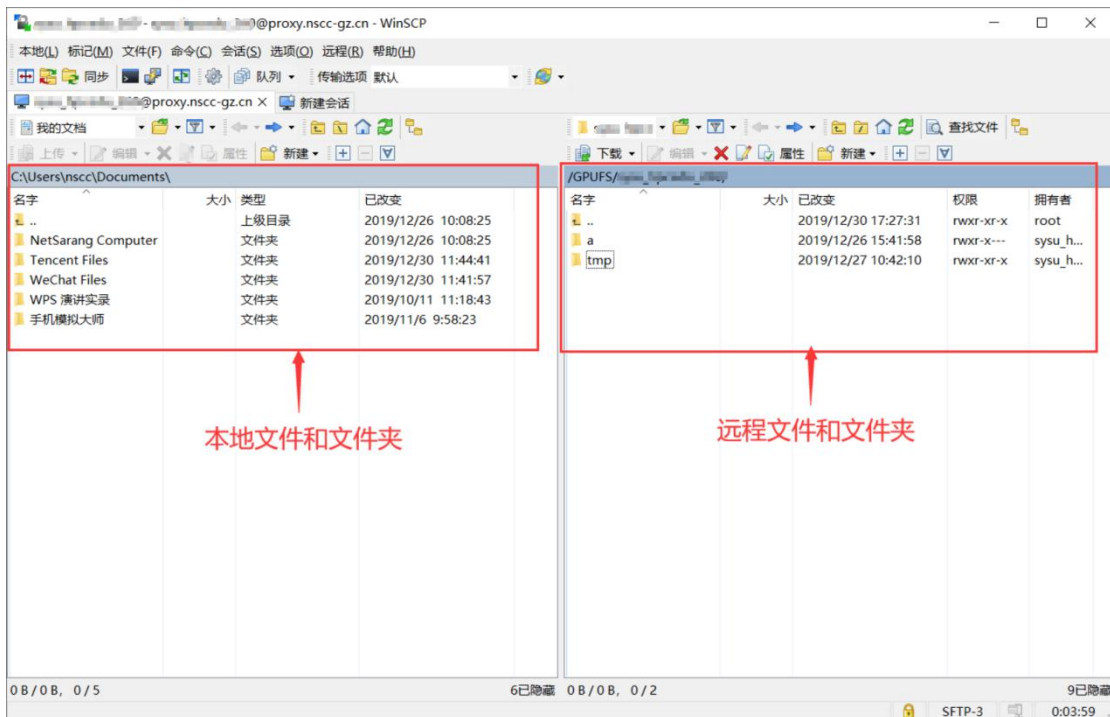
2) 打开 WinSCP，根据步骤 1 的信息填写主机名、端口号和用户名信息，密码使用星光平台的登录密码，输入完成后点击“登录”按钮：



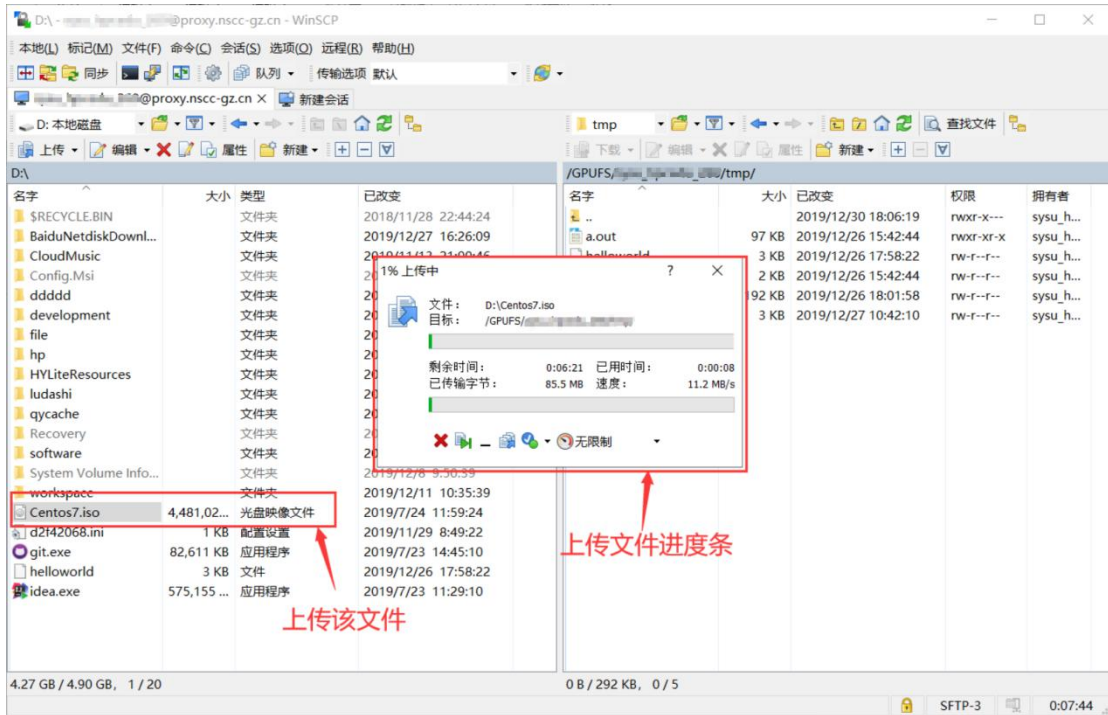
3) 选择要登录的作业对应的编号，如本例中的作业编号为 1，选择完成后点击“确定”按钮：



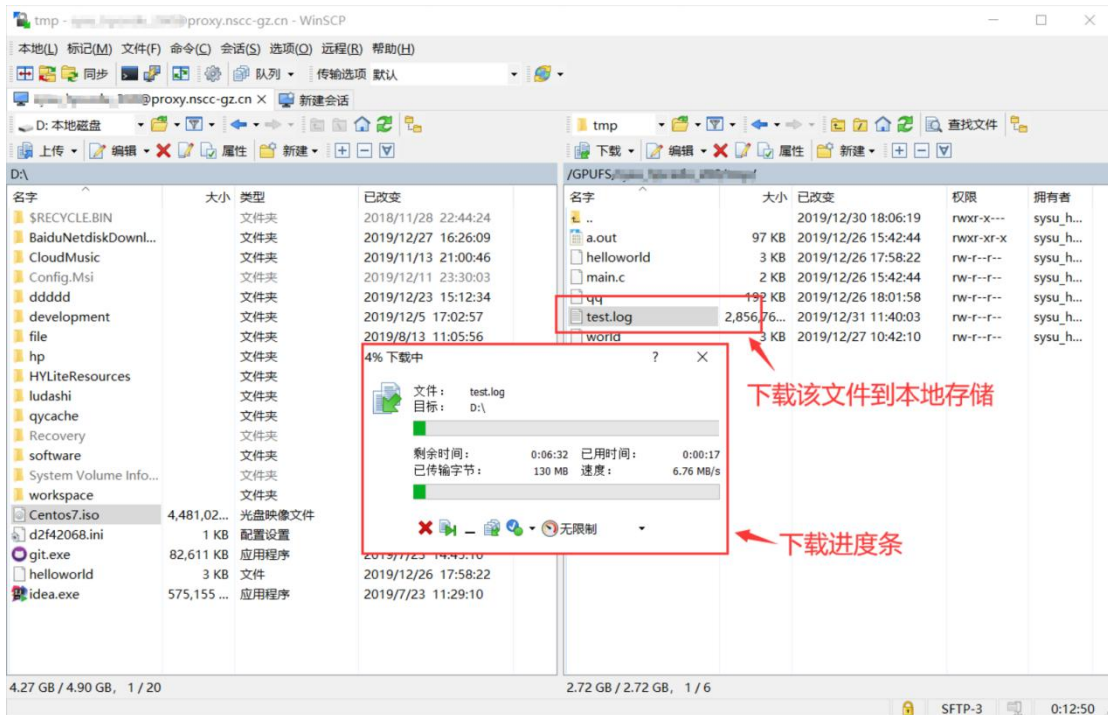
- 4) 进入 WinSCP 的文件传输界面，左侧是本地文件和文件夹，右侧是远程服务器的文件和文件夹：



- 5) 上传文件：选中一个本地文件，右击弹出操作菜单，点击“上传”即可上传文件到远程服务器：



6) 下载文件：选中一个远程文件，右击弹出操作菜单，点击“下载”即可下载文件到本地服务器：

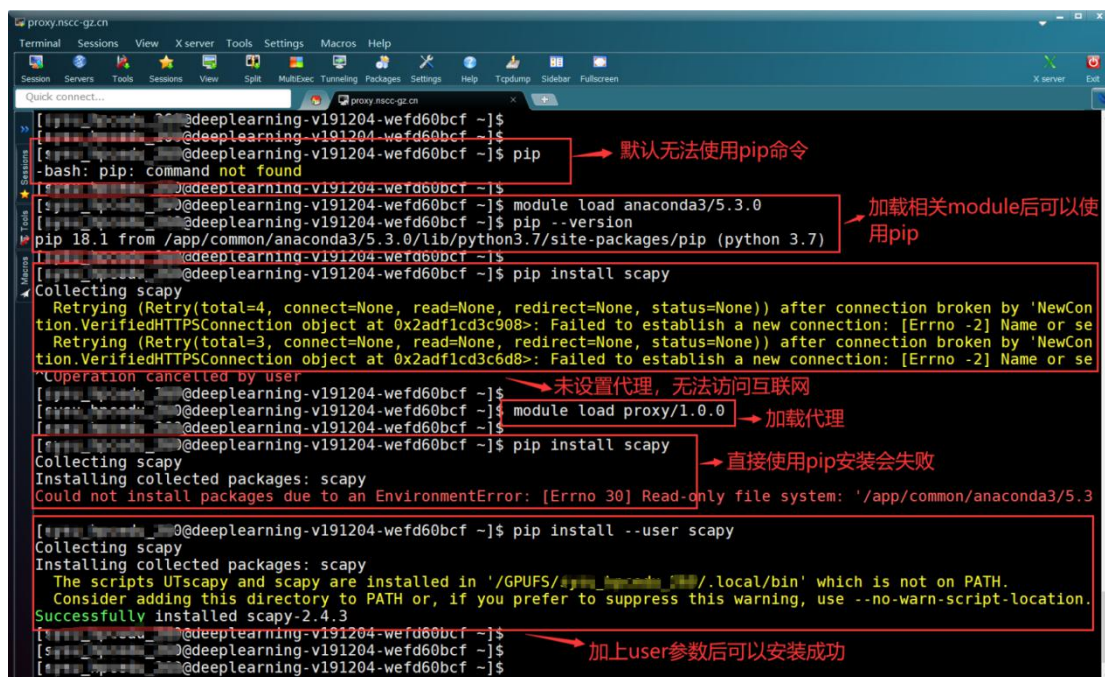


11.4 pip 安装用户 python 包

用户直接使用 pip 命令进行 python 包的安装，安装会失败。要在星光平台进行用户自定义 python 包安装时需要安装以下几个步骤进行：

- 1) 加载包含 pip 命令的相关 module，如：anaconda3/5.3.0
- 2) 加载互联网代理 module：proxy/1.0.0
- 3) 使用 pip 命令时要带上 user 参数，如：pip install --user xxx

下图演示了使用 pip 安装 scapy 的步骤：



```
proxy.nssc-gz.cn
Terminal Sessions View X server Tools Settings Macros Help
Session Servers Tools Sessions View Split MultiExec Tunneling Packages Settings Help Tcdump Sidebar Fullscreen
Quick connect... proxy.nssc-gz.cn
[ ] @deeplearning-v191204-wefd60bcf ~]$
[ ] @deeplearning-v191204-wefd60bcf ~]$ pip
-bash: pip: command not found
[ ] @deeplearning-v191204-wefd60bcf ~]$ module load anaconda3/5.3.0
[ ] @deeplearning-v191204-wefd60bcf ~]$ pip --version
pip 18.1 from /app/common/anaconda3/5.3.0/lib/python3.7/site-packages/pip (python 3.7)
[ ] @deeplearning-v191204-wefd60bcf ~]$ pip install scapy
Collecting scapy
  Retrying (Retry(total=4, connect=None, read=None, redirect=None, status=None)) after connection broken by 'NewCon
tion.VerifiedHTTPSConnection object at 0x2adflcd3c908': Failed to establish a new connection: [Errno -2] Name or se
  Retrying (Retry(total=3, connect=None, read=None, redirect=None, status=None)) after connection broken by 'NewCon
tion.VerifiedHTTPSConnection object at 0x2adflcd3c6d8': Failed to establish a new connection: [Errno -2] Name or se
  Operation cancelled by user
[ ] @deeplearning-v191204-wefd60bcf ~]$ module load proxy/1.0.0
[ ] @deeplearning-v191204-wefd60bcf ~]$ pip install scapy
Collecting scapy
Installing collected packages: scapy
Could not install packages due to an EnvironmentError: [Errno 30] Read-only file system: '/app/common/anaconda3/5.3
[ ] @deeplearning-v191204-wefd60bcf ~]$ pip install --user scapy
Collecting scapy
Installing collected packages: scapy
The scripts UTscapy and scapy are installed in '/GPUFS/.../local/bin' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed scapy-2.4.3
[ ] @deeplearning-v191204-wefd60bcf ~]$
```

默认无法使用pip命令

加载相关module后可以使用pip

未设置代理，无法访问互联网

加载代理

直接使用pip安装会失败

加上user参数后可以安装成功

Pip 使用手册链接为：https://pip.pypa.io/en/stable/user_guide/

11.5 自定义源码安装软件

用户在自有集群中编译安装其他第三方软件时，可选择自定义编译安装模式，对于可自定义编译安装的软件，其官方源码一般都会提供自定义安装选项，如在大部分官方源码文件都会提供 **configure** 文件，在自定义安装前可采用如下命令 **./configure --help** 列出相关信息，若提供有 **--prefix=\$HOME** 选项，则采用 **./configure --prefix=/user/\$dir/xxx** 形式提供自定义安装目录。在此种情况下即可以安装软件到用户自定义目录中，。

注意：每个软件包的安装形式不同，在自定义安装前，强烈建议首先查看该软件的安装指引文件。

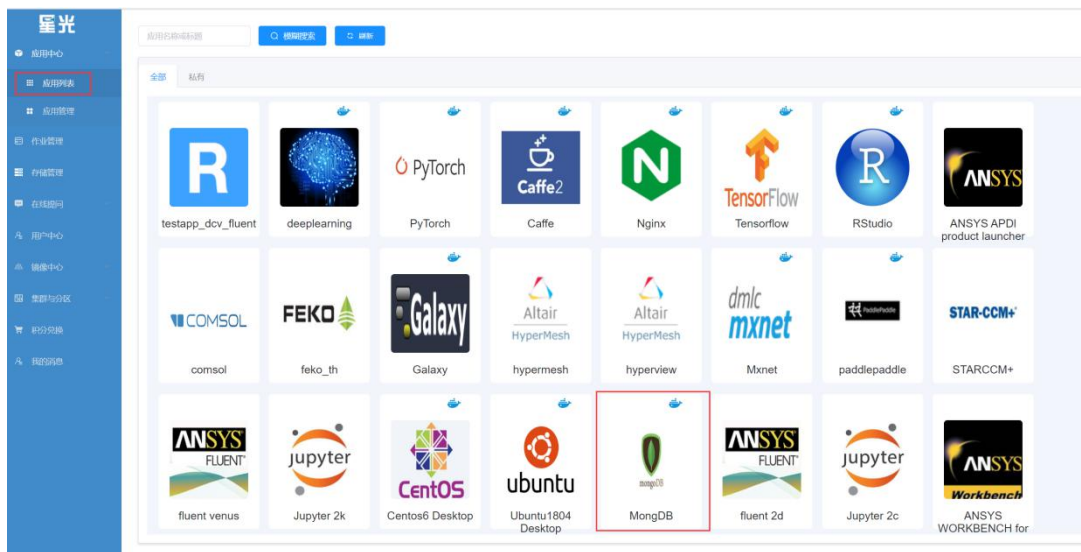
11.6 常见应用使用说明

11.6.1 MongoDB 使用说明

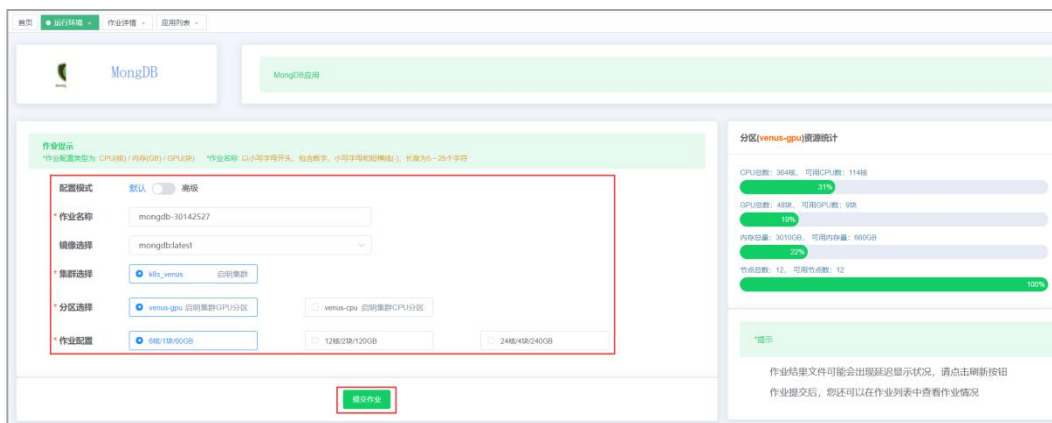
11.6.1.1 配置运行 MongoDB

在星光平台上配置运行 mongodb 的步骤如下：

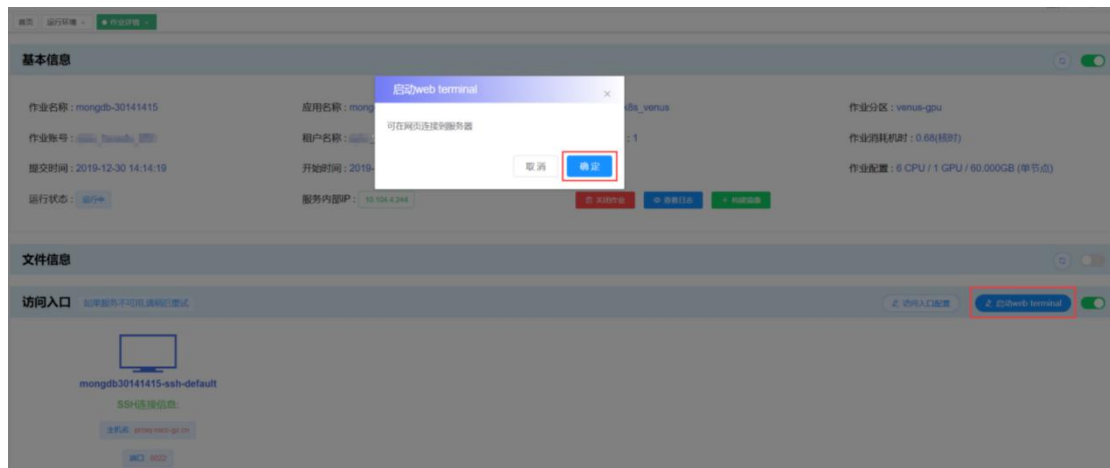
- 1) 点击“应用中心”→“应用列表”，选择 MongoDB 应用：



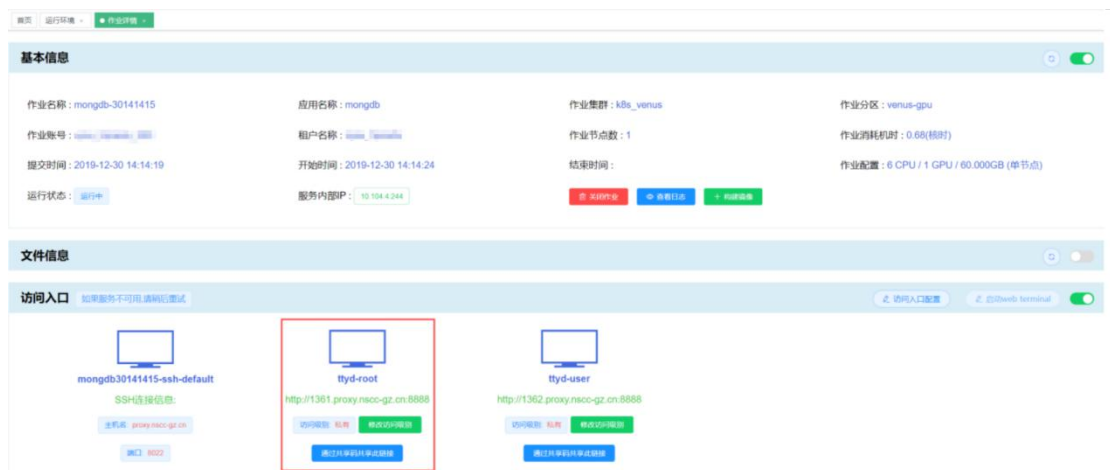
- 2) 选择相应的配置后，点击“提交作业”：



- 3) 在作业详情右下角点击“启动 web terminal”按钮，再点击弹出框中的“确定”添加访问入口：



4) 点击名为“ttyd-root”的访问入口，以 root 身份进入应用内部：



5) 在 mongodb 应用内部配置和运行 mongodb 服务：


```

root@vfce2f852-fa85-4b7a-b5e9-5ce16e17e154-5fbbd97695-1zrrb:~# mongo
MongoDB-shell-version: v4.2.2
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("7c1a7a95-73c4-43c9-a30b-50fa0c02cc13") }
MongoDB server version: 4.2.2
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
  http://docs.mongodb.org/
Questions? Try the support group
  http://groups.google.com/group/mongodb-user
Server has startup warnings:
2019-12-30T06:34:56.513+0000 I CONTROL [initandlisten]
2019-12-30T06:34:56.513+0000 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2019-12-30T06:34:56.513+0000 I CONTROL [initandlisten] **          Read and write access to data and configuration is unrestricted.
2019-12-30T06:34:56.513+0000 I CONTROL [initandlisten] ** WARNING: You are running this process as the root user, which is not recommended.
2019-12-30T06:34:56.513+0000 I CONTROL [initandlisten]
2019-12-30T06:34:56.514+0000 I CONTROL [initandlisten]
2019-12-30T06:34:56.514+0000 I CONTROL [initandlisten] ** WARNING: You are running on a NUMA machine.
2019-12-30T06:34:56.514+0000 I CONTROL [initandlisten] **          We suggest launching mongod like this to avoid performance problems:
2019-12-30T06:34:56.514+0000 I CONTROL [initandlisten] **          numactl --interleave=all mongod [other options]
2019-12-30T06:34:56.514+0000 I CONTROL [initandlisten]
2019-12-30T06:34:56.514+0000 I CONTROL [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.
2019-12-30T06:34:56.514+0000 I CONTROL [initandlisten] **          We suggest setting it to 'never'
2019-12-30T06:34:56.514+0000 I CONTROL [initandlisten]
2019-12-30T06:34:56.514+0000 I CONTROL [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/defrag is 'always'.
2019-12-30T06:34:56.514+0000 I CONTROL [initandlisten] **          We suggest setting it to 'never'
2019-12-30T06:34:56.514+0000 I CONTROL [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
> show databases;
admin    0.000GB
config  0.000GB
local   0.000GB
>

```

11.6.1.3 平台其它应用访问 MongoDB 服务

要让其他应用能够访问该 mongodb 服务，首先 mongodb 服务需要把自己的服务端口暴露出去，即添加一个访问入口（关于访问入口详见：[4.2.2](#)），如下：

添加访问入口

*名称: tcp-27017

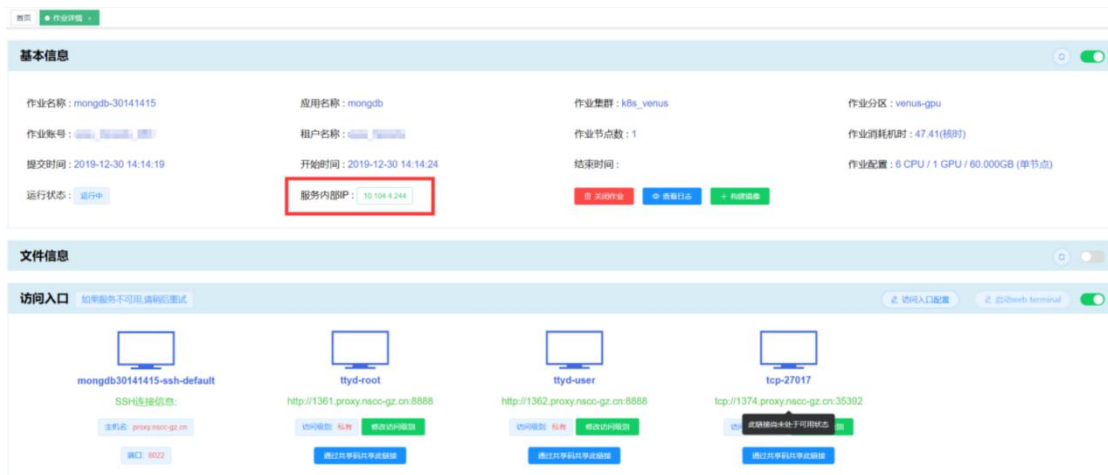
协议: tcp (将从32768-65535之间随机选择一个作为外部访问端口)

目的端口: 27017 (注: 容器内部服务所监听的端口)

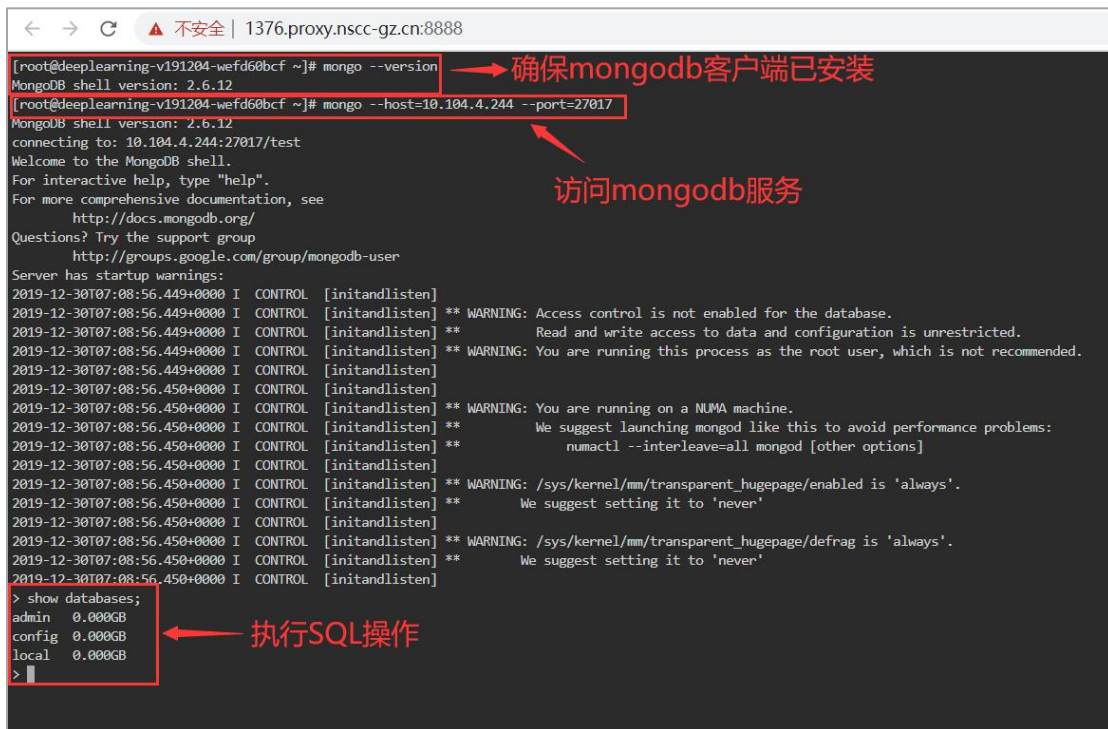
取消 确定

27017 为 mongodb 的默认服务端口。

其它应用通过 mongodb 服务的 IP 地址和端口信息即可访问 mongodb 服务，服务的 IP 地址可以在作业详情中查询：



另外，平台其它应用访问 mongodb 服务需要安装好 mongodb 客户端（可通过系统的软件包管理器如 yum 或通过源码等方式进行安装），通过 mongodb 客户端来访问 mongodb 服务。下图以深度学习应用“deeplearning”访问 mongodb 服务为例，演示其他应用访问 mongodb 的方法：



11.6.1.4 注意事项

在容器中运行数据库服务需要注意，容器随时可能发生重启、迁移，因此在配置数据库服务时有以下建议：

- 1) 数据目录应为用户共享目录等可持久化的目录。在星光平台中用户共享目录为 /GPUFS/\$USER_NAME
- 2) 为保证 mongodb 数据库中数据的安全性和防止程序或系统故障造成的数据丢失，应当设置数据库的定时备份

11.6.1.5 关键配置项参考

在星光平台上运行 mongodb 服务需要注意一下几个配置项：

```
1 # mongod.conf
2
3 # for documentation of all options, see:
4 # http://docs.mongodb.org/manual/reference/configuration-options/
5
6 # Where and how to store data.
7 storage:
8   dbPath: /GPUFS/${user_name}/mongodb # 此处需要修改为用户共享目录下路径
9   journal:
10     enabled: true
11 # engine:
12 # mmapv1:
13 # wiredTiger:
14
15 # where to write logging data.
16 systemLog:
17   destination: file
18   logAppend: true
19   path: /GPUFS/${user_name}/log/mongodb/mongod.log # 此处需要修改为用户共享目录下路径
20
21 # network interfaces
22 net:
23   port: 27017
24   bindIp: 127.0.0.1 # 若需暴露给外部应用或服务访问需要修改绑定IP
25
26
27 # how the process runs
28 processManagement:
29   timeZoneInfo: /usr/share/zoneinfo |
30
31 #security:
32
33 #operationProfiling:
34
35 #replication:
36
37 #sharding:
38
39 ## Enterprise-Only Options:
40
41 #auditLog:
42
43 #snmp:
```

MongoDB 配置项详见 MongoDB 官方文档：

<https://docs.mongodb.com/manual/reference/configuration-options/>

11.6.1.6 MongoDB 备份和状态监控

1) 备份

官方数据备份方案参考：<http://www.mongoinc.com/docs/core/backups.html>

Mongdb 数据备份通常有基于文件的数据备份、mongdbDump、mongdbExport 命令等方式进行备份，在星光平台上对 mongdb 数据进行备份时需要注意的是备份数据存储目录需要设置为用户共享目录，依据数据的重要性，建议设置备份周期为一天、一周或一个月等周期。

2) 状态监控

在容器环境下，容器是有可能发生销毁重建或容器迁移的，在此种环境下对运行的数据库类服务除数据的定时备份外，也需要对数据库服务的健康状态进行监控。若监控到服务异常，则通常需要人工干预。(该流程与在物理机环境下运维数据库服务相同)

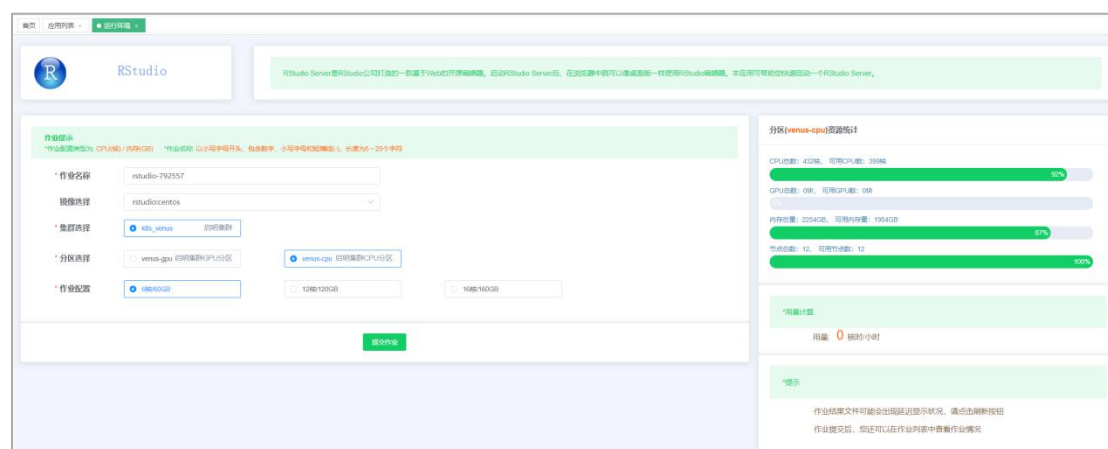
关于数据库服务的状态监控和数据库的定时备份，均可采用 python 程序或其他用户自定义程序的方式来进行。如可使用 python 中 sched、scheduler、timer 模块等进行数据库的定时备份；而对 mongodb 进程的监控可使用 python-mongodb 进行服务的状态获取和监控。

11.6.2 RStudio 使用说明

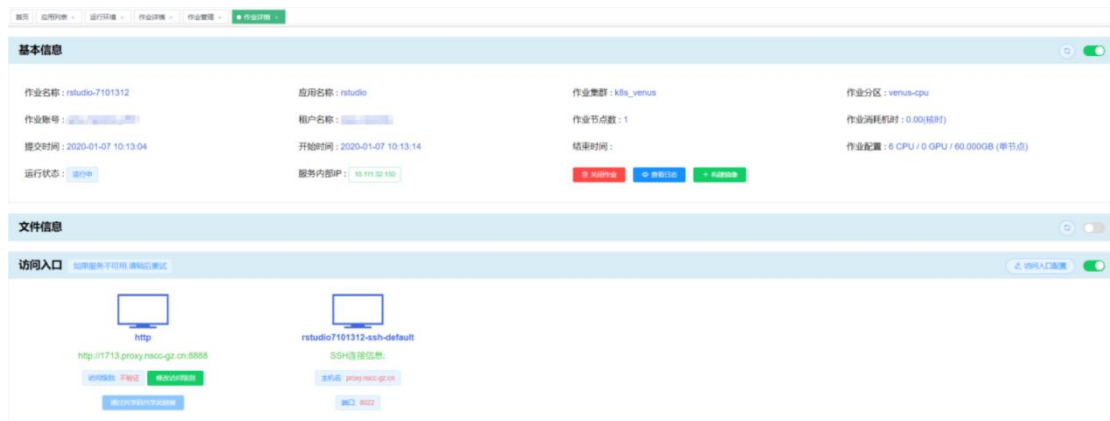
Rstudio 是针对 R 语言的一项开源的集成开发环境，其具有桌面版和服务器版。下面以 Rstudio 中 demo 函数为例，演示创建和使用 Rstudio 程序的步骤和相关流程，关于 Rstudio 的专业知识请参考 [rstudio 官网](#)。

11.6.2.1 RStudio 应用创建步骤

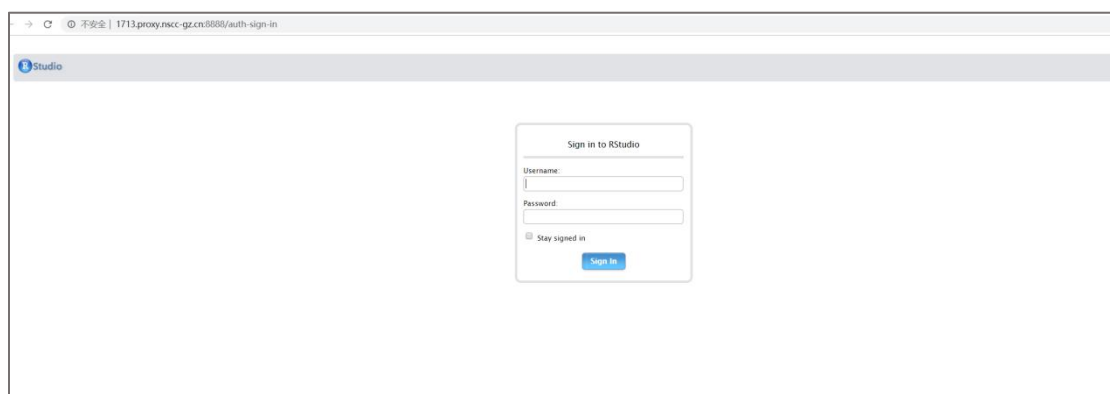
步骤 1：提交 RStudio 应用作业：



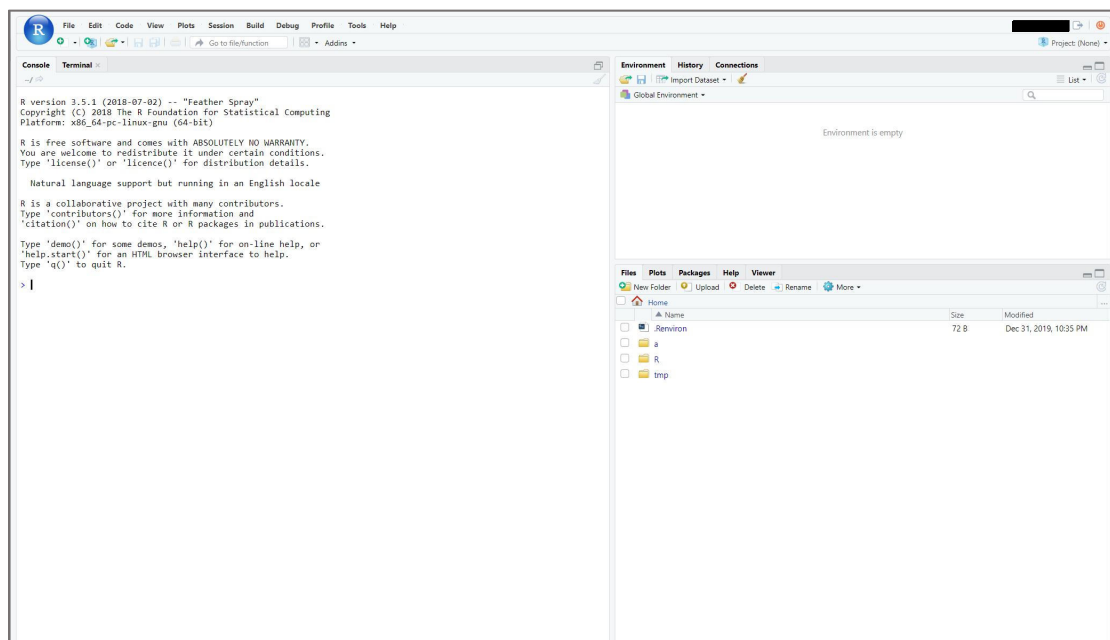
步骤 2：作业运行成功，自动进入作业详情界面：



步骤 3: 点击 http 协议的访问入口, 登录框的用户名、密码为用户的系统账号 (即“作业详情” - “基本信息”中的“提交账号”) 和“星光”登录密码:

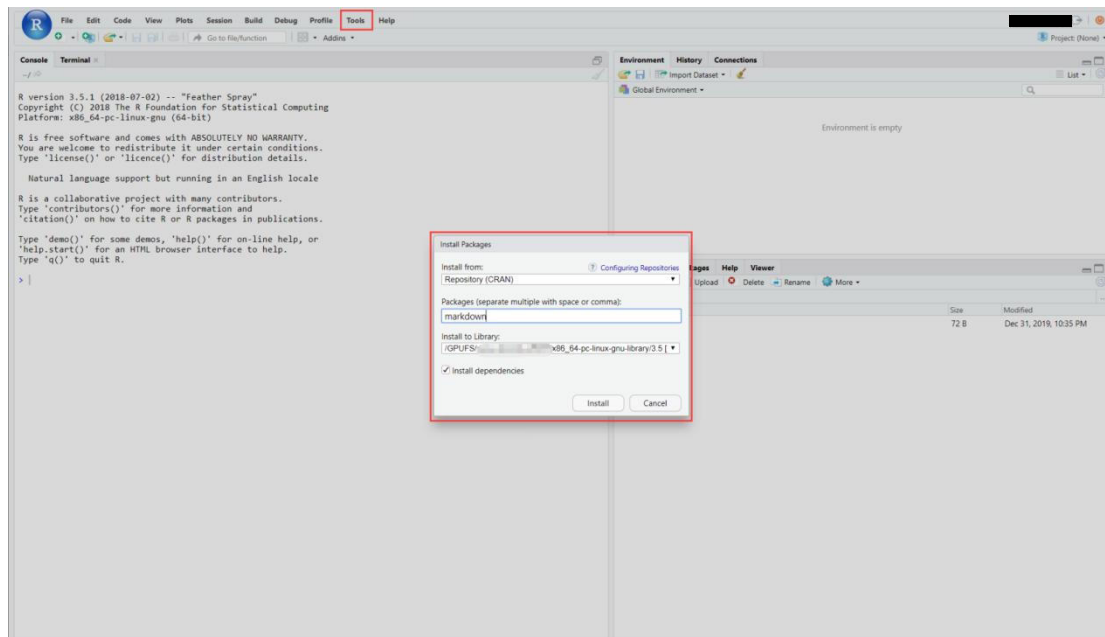


步骤 4: 输入用户名和密码信息后, 点击“Sign in”即可进入 RStudio 应用:

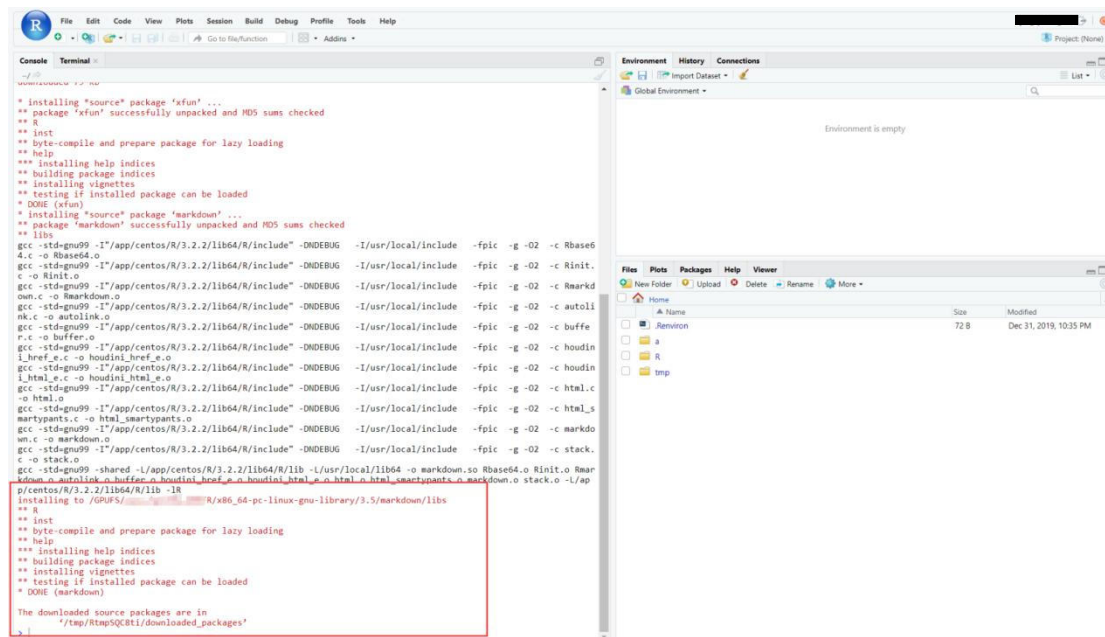


11.6.2.2 在线安装 R package

点击 RStudio 菜单栏的 tools, 在弹出框中输入要安装的 package:

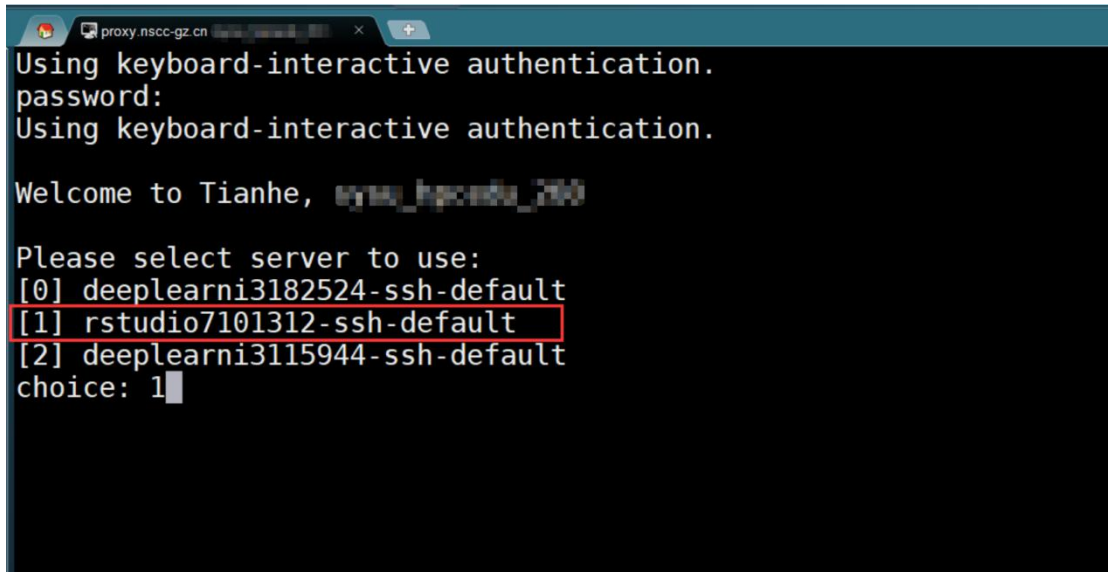


点击弹出框的“install”按钮即可完成安装：

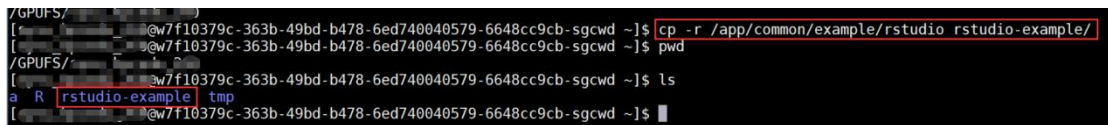


11.6.2.3 运行 RStudio 示例程序

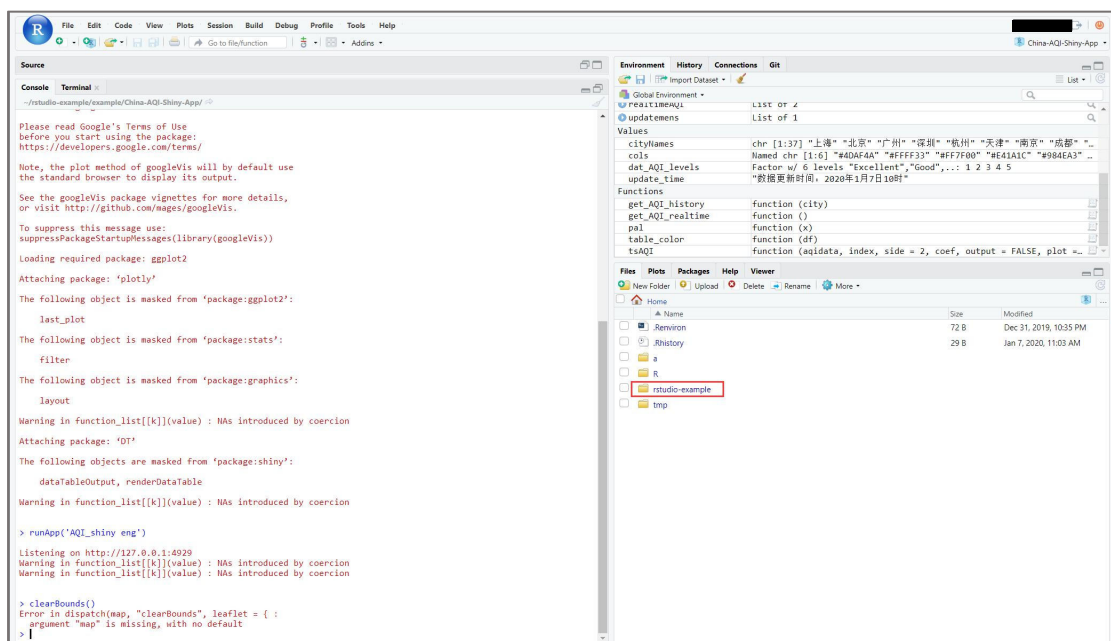
示例程序位于 `/app/common/example/rstudio` 目录下，需要将示例程序拷贝到工作目录下。首先通过 SSH 登录作业（详细步骤参考 [4.2.2.2](#)）：



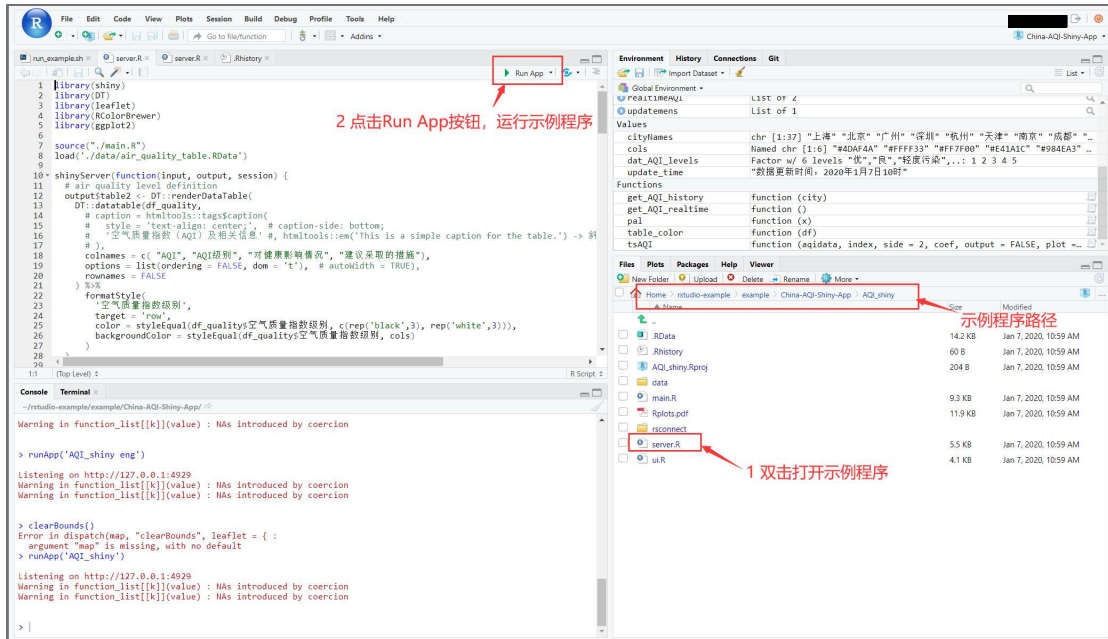
执行命令 `cp -r /app/common/example/rstudio rstudio-example/` 将示例程序拷到工作目录下:



此时, 进入 RStudio 应用便可以看到示例程序相关的目录:



找到示例程序, 点击“Run App”运行示例程序:



看到类似以下界面表明示例程序运行成功：



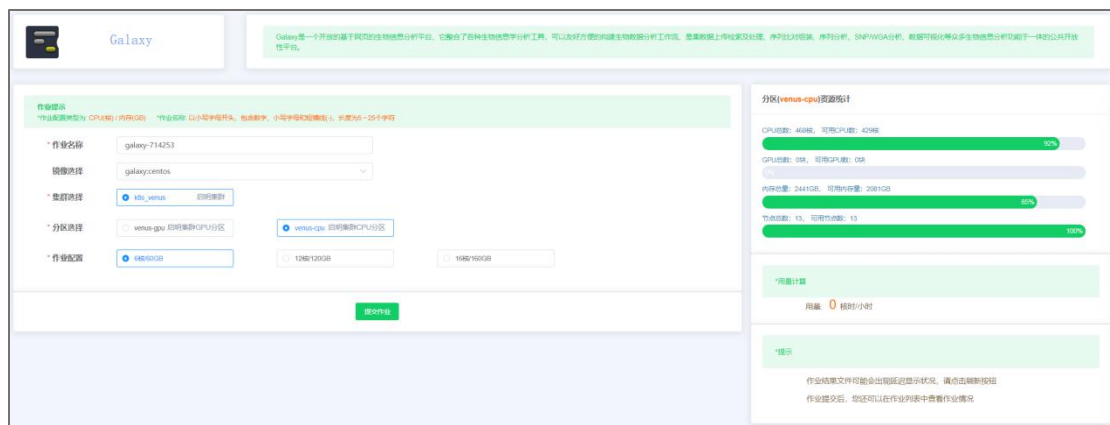
11.6.3 Galaxy 使用说明

Galaxy 是生物分析领域著名的数据分析平台，由美国宾夕法尼亚州立大学基因比较和生物计算中心和约翰·霍普金斯大学生物学院资助开发，在北美，欧洲和日本生命科学界有着广泛的应用。Galaxy 可以运用公用平台或自行安装，可以执行，复制和分享完整的分析流程。下面以 Galaxy 的常见应用为例，演示创建和使用 Galaxy 的相关步骤和流程。关于 galaxy 的专业知识可参考以下资料：

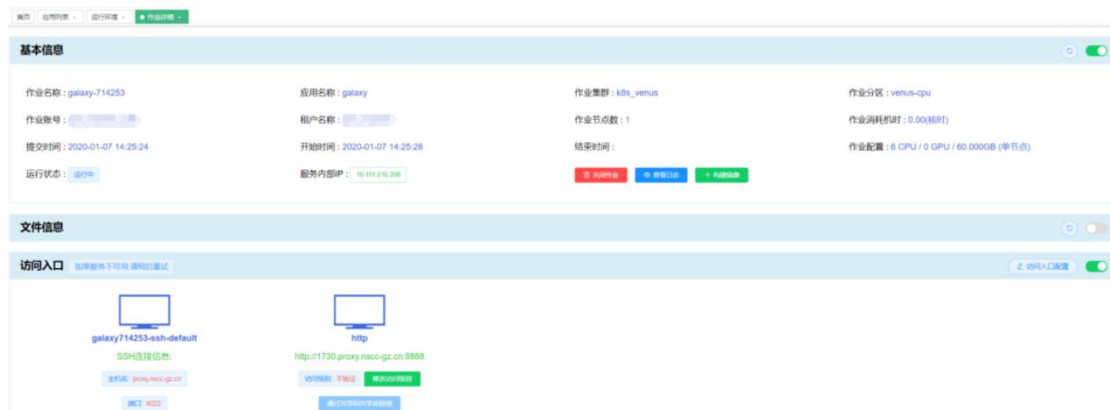
1. [Galaxy 官网](#)
2. [Galaxy 生物信息分析平台简明使用手册](#)
3. [Galaxy 数据处理平台简介](#)

11.6.3.1 Galaxy 应用创建步骤

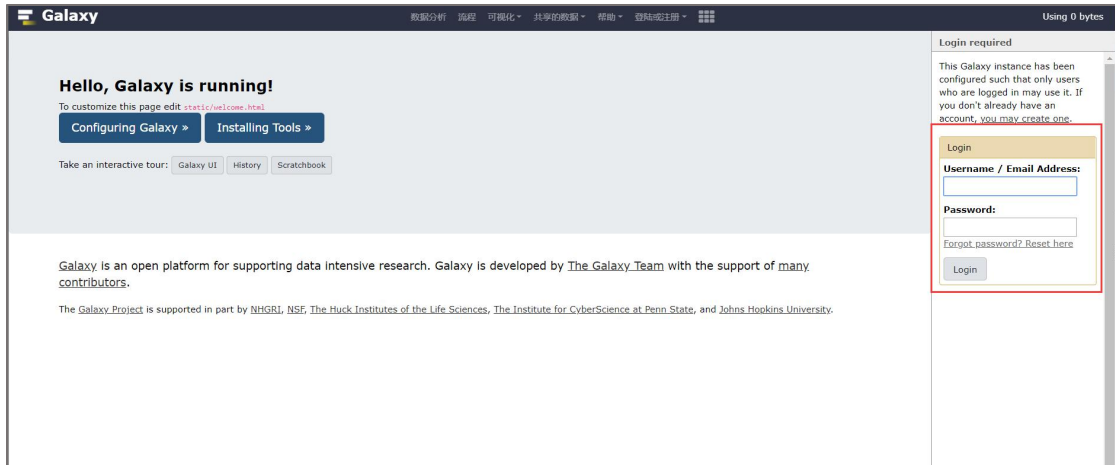
步骤 1：提交 Galaxy 作业



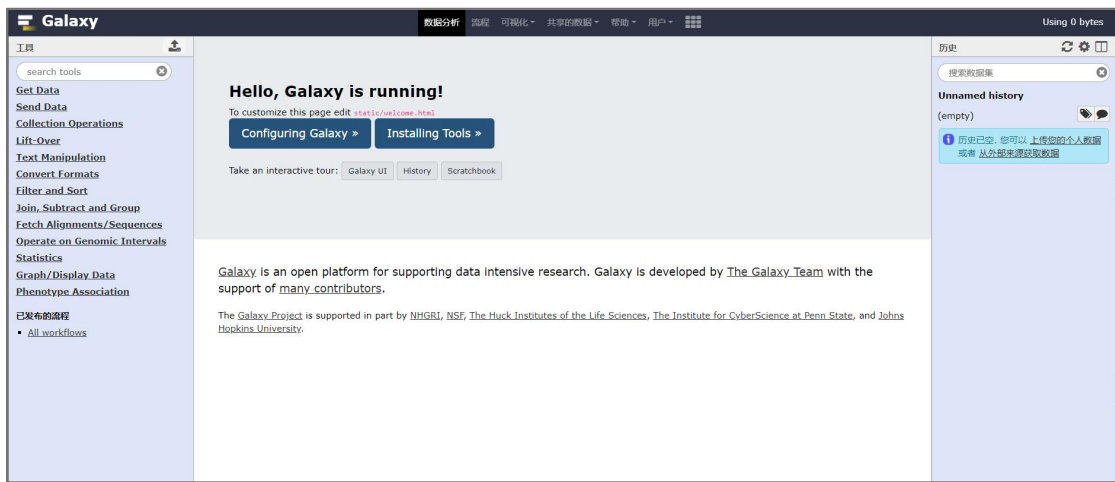
提交成功后自动进入作业详情界面：



步骤 2：点击 http 协议的访问入口访问 Galaxy 应用，Galaxy 需要使用 Galaxy 账号（首次使用需要创建账号，见 10.10.3.2）登录后才可正常使用，在登录框中输入 Galaxy 账号信息后点击“Login”按钮可登录：

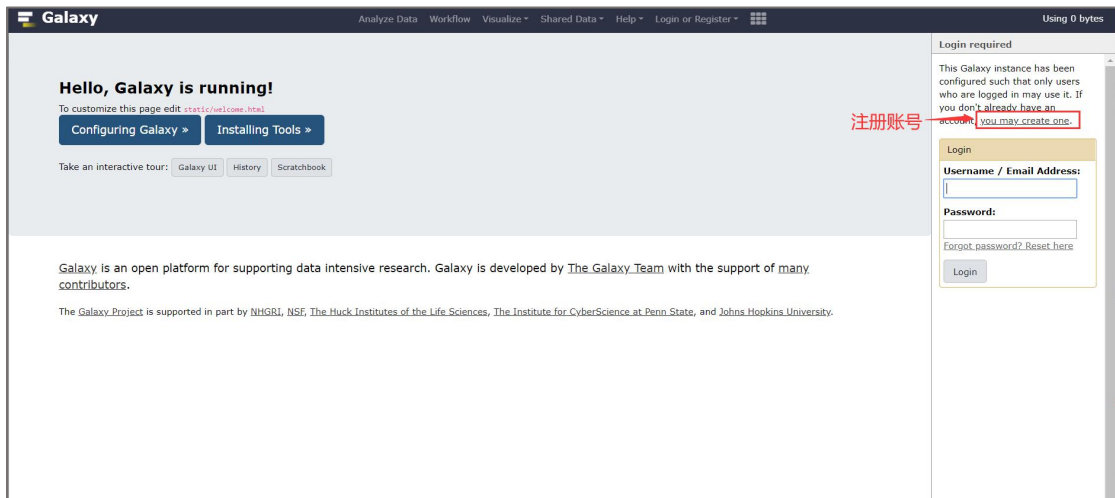


登录成功后进入类似以下界面：

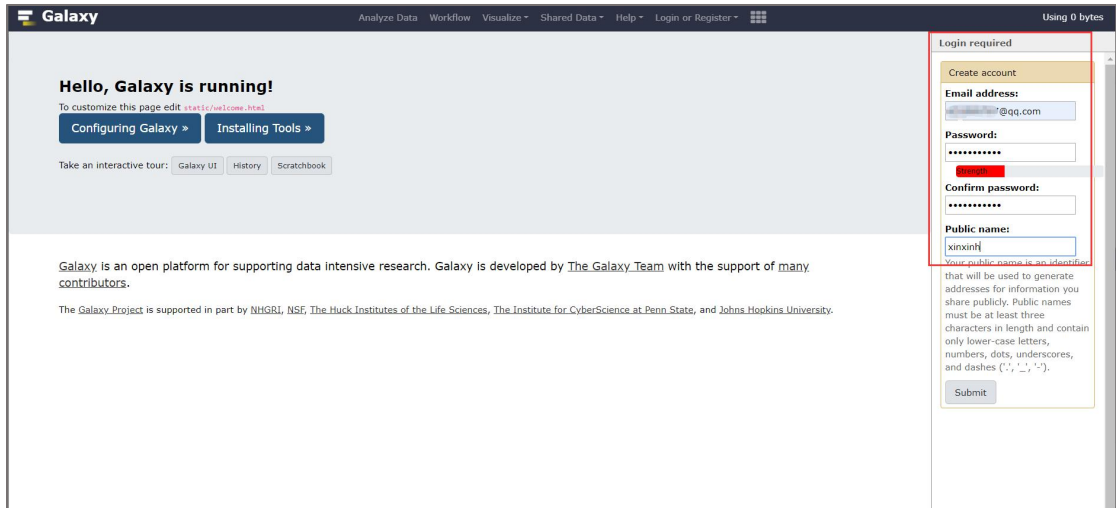


11.6.3.2 创建 Galaxy 账号

点击应用右侧的“you may create one”链接，进入创建账号界面：



输入邮箱、用户名和密码信息，点击“Submit”按钮即可完成账号创建，创建完后应用会自动使用刚创建的账号登录系统：

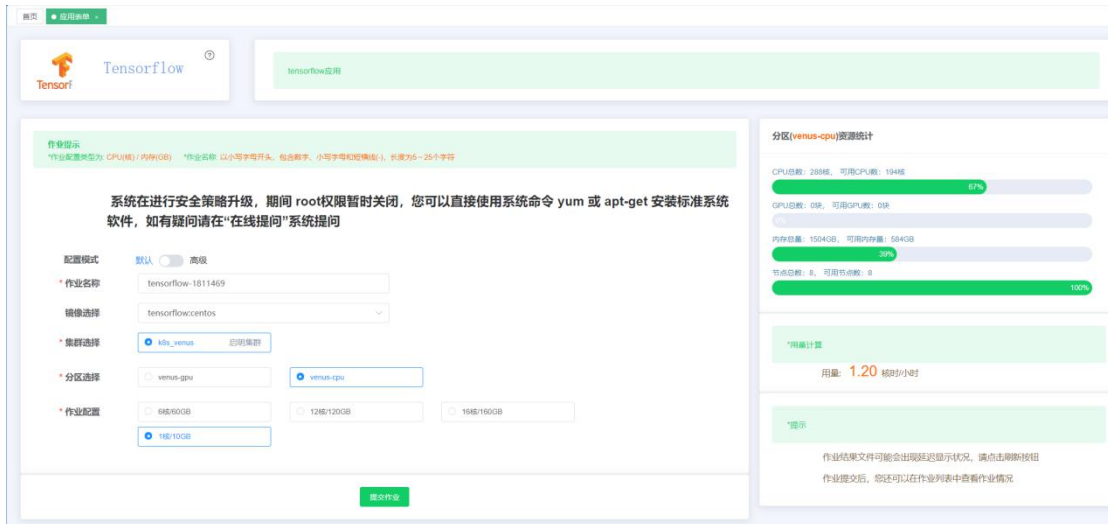


11.6.4 Tensorflow 使用说明

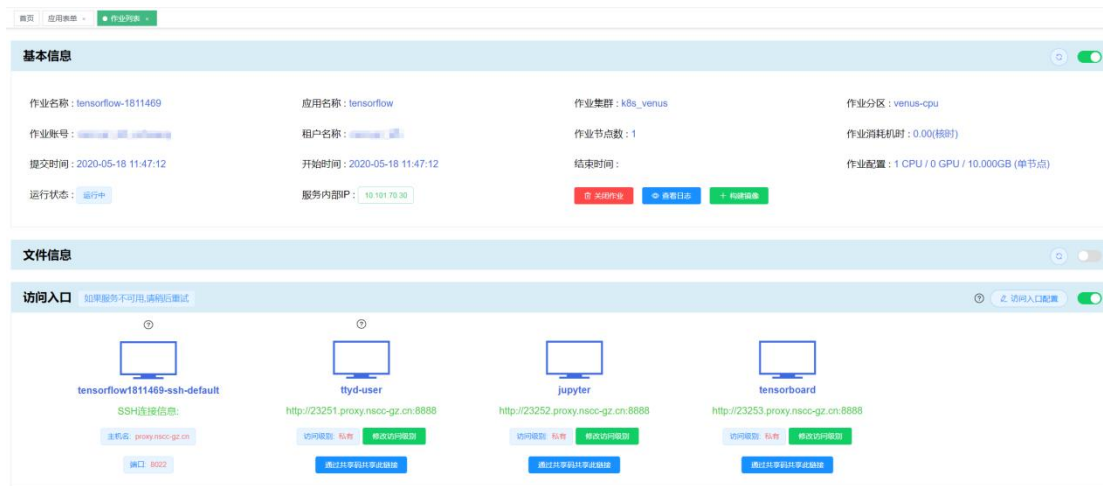
本节演示了 Tensorflow 应用的创建以及使用流程，关于 Tensorflow 的专业知识请参考 [Tensorflow 官网](#)。

11.6.4.1 Tensorflow 应用创建步骤

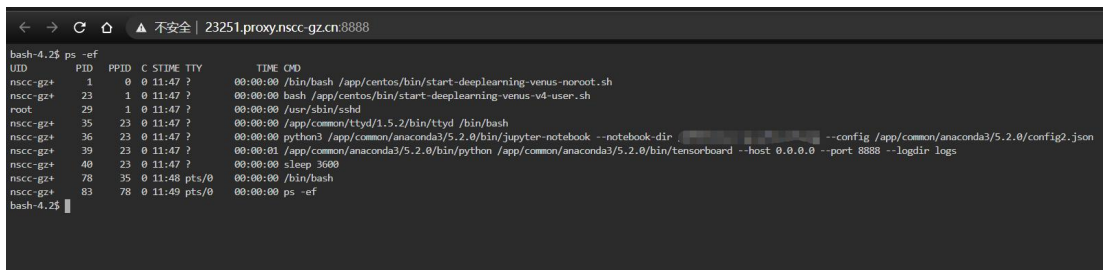
和其他应用类似，首先先提交 Tensorflow 作业：



提交成功后自动进入作业详情界面：

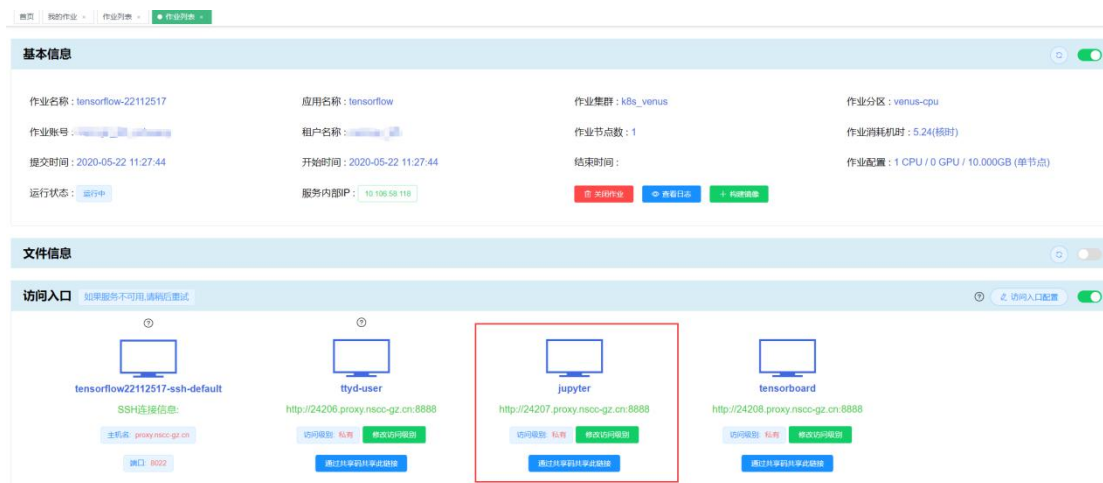


通过“ttyd-user”访问入口访问应用：

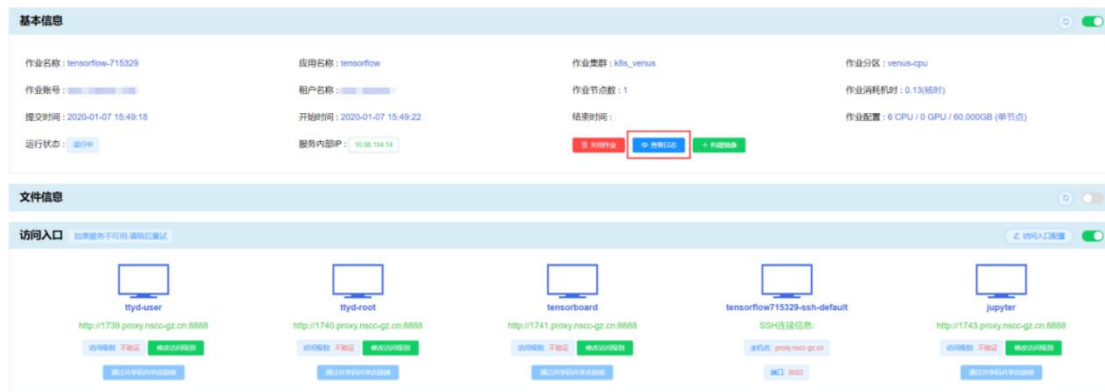


11.6.4.2 通过 Jupyter 访问

点击“jupyter”访问入口：



使用 jupyter 登录需要 token 信息，token 信息需要到“基本信息”的“查看日志”中查看：



点击“查看日志”按钮，把日志拉到最后，可看见类似下面的 token 信息：

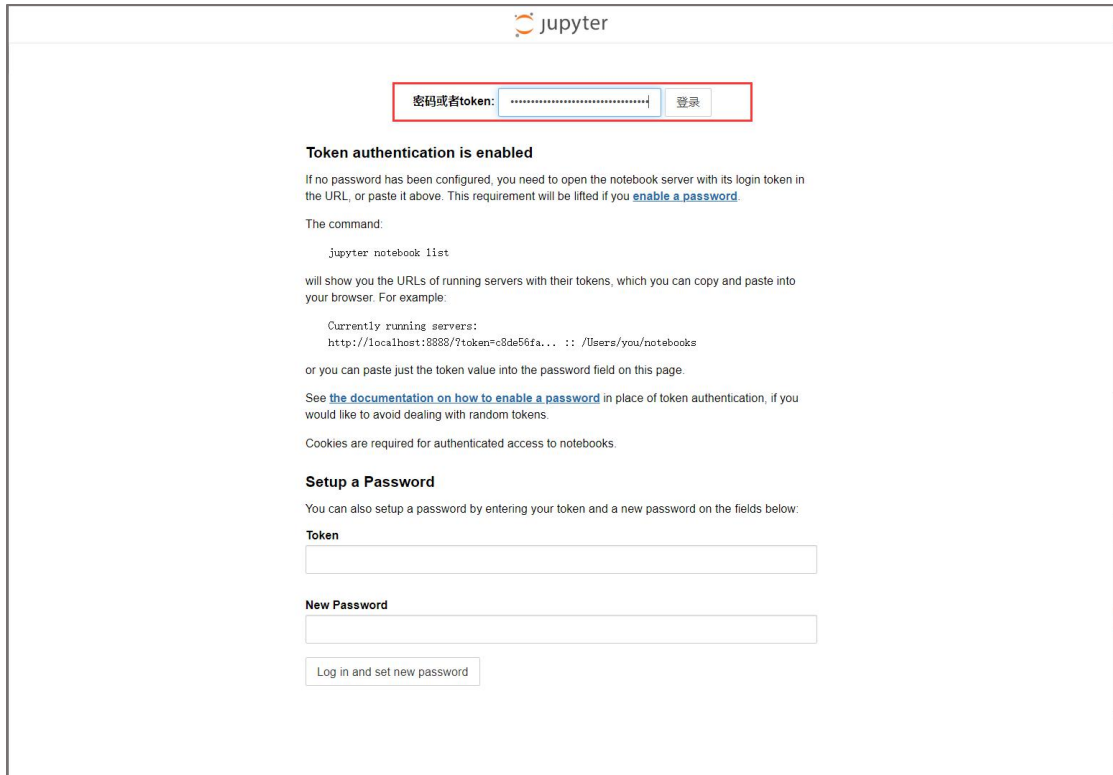
```

[2020/01/07 15:49:32:3516] NOTICE: libev support not compiled in
[2020/01/07 15:49:32:3516] NOTICE: libuv support not compiled in
[2020/01/07 15:49:32:3517] NOTICE: Threads: 1 each 1048576 fds
[2020/01/07 15:49:32:3565] NOTICE: mem: platform fd map: 8388608 bytes
[2020/01/07 15:49:32:3565] NOTICE: Compiled with OpenSSL support
[2020/01/07 15:49:32:3565] NOTICE: SSL disabled: no LWS_SERVER_OPTION_DO_SSL_GLOBAL_INIT
[2020/01/07 15:49:32:3565] NOTICE: Creating Vhost 'default' port 7681, 2 protocols, IPv6 off
[2020/01/07 15:49:32:3566] NOTICE: Listening on port 7681
[2020/01/07 15:49:32:3566] NOTICE: mem: per-conn: 568 bytes + protocol rx buf
[2020/01/07 15:49:32:3566] NOTICE: canonical_hostname = tensorflow-wb21c2ad0
[W 15:49:36.310 NotebookApp] WARNING: The notebook server is listening on all IP addresses and not using encryption. This is not recommended.
[I 15:49:36.397 NotebookApp] JupyterLab beta preview extension loaded from /app/common/anaconda3/5.2.0/lib/python3.6/site-packages/jupyterlab
[I 15:49:36.397 NotebookApp] JupyterLab application directory is /app/common/anaconda3/5.2.0/share/jupyter/lab
[I 15:49:36.403 NotebookApp] 启动notebooks 在本地路径: /GPUS/
[I 15:49:36.403 NotebookApp] 本程序运行在: http://tensorflow-wb21c2ad0:8080/?token=fd0994836053aeb256fe7f87f0a2af0516579af22560ed4c
[I 15:49:36.403 NotebookApp] or http://127.0.0.1:8080/?token=fd0994836053aeb256fe7f87f0a2af0516579af22560ed4c
[I 15:49:36.403 NotebookApp] 使用control-c停止此服务器并关闭所有内核(两次跳过确认)。
[C 15:49:36.412 NotebookApp]

To access the notebook, open this file in a browser:
file:///GPUS/.../local/share/jupyter/runtime/nbserver-58-open.html
Or copy and paste one of these URLs:
http://tensorflow-wb21c2ad0:8080/?token=fd0994836053aeb256fe7f87f0a2af0516579af22560ed4c
or http://127.0.0.1:8080/?token=fd0994836053aeb256fe7f87f0a2af0516579af22560ed4c
/app/common/anaconda3/5.2.0/lib/python3.6/site-packages/h5py/_init_.py:36: FutureWarning: Conversion of the second argument of isubdtype from 'float' to
from _conv import register_converters as _register_converters
TensorBoard 1.10.0 at http://tensorflow-wb21c2ad0:8888 (Press CTRL+C to quit)
[2020/01/07 15:52:08:8221] NOTICE: HTTP / - 192.168.11.192 (192.168.11.192)
[2020/01/07 15:52:08:8657] NOTICE: HTTP /auth_token.js - 192.168.11.192 (192.168.11.192)
[2020/01/07 15:52:08:8937] NOTICE: WS /ws - 192.168.11.192 (192.168.11.192), clients: 1
[2020/01/07 15:52:09:2065] NOTICE: started process, pid: 71
[2020/01/07 15:57:22:4173] NOTICE: sending SIGHUP (1) to process (group) -71
[2020/01/07 15:57:22:4173] NOTICE: WS closed from 192.168.11.192 (192.168.11.192), clients: 0
[2020/01/07 15:57:22:4177] NOTICE: process exited with code 1, pid: 71
[I 15:57:47.928 NotebookApp] 302 GET / (192.168.11.192) 1.05ms
[I 15:57:47.933 NotebookApp] 302 GET /tree? (192.168.11.192) 0.56ms
[W 16:00:00.260 NotebookApp] 401 POST /login?next=%2Ftree%3F (192.168.11.192) 1.90ms referer=http://1743.proxy.nssc-gz.cn:8888/login?next=%2Ftree%3F
[I 16:05:55.416 NotebookApp] 302 GET / (192.168.11.192) 0.53ms
[I 16:05:55.424 NotebookApp] 302 GET /tree? (192.168.11.192) 0.53ms
[I 16:05:58.000 NotebookApp] 302 POST /login?next=%2Ftree%3F (192.168.11.192) 0.98ms
[I 16:06:02.772 NotebookApp] 302 GET /login?next=%2Ftree%3F (192.168.11.192) 0.75ms

```

将 token 输入后点击“登录”按钮：



出现类似以下界面则表明登录成功：

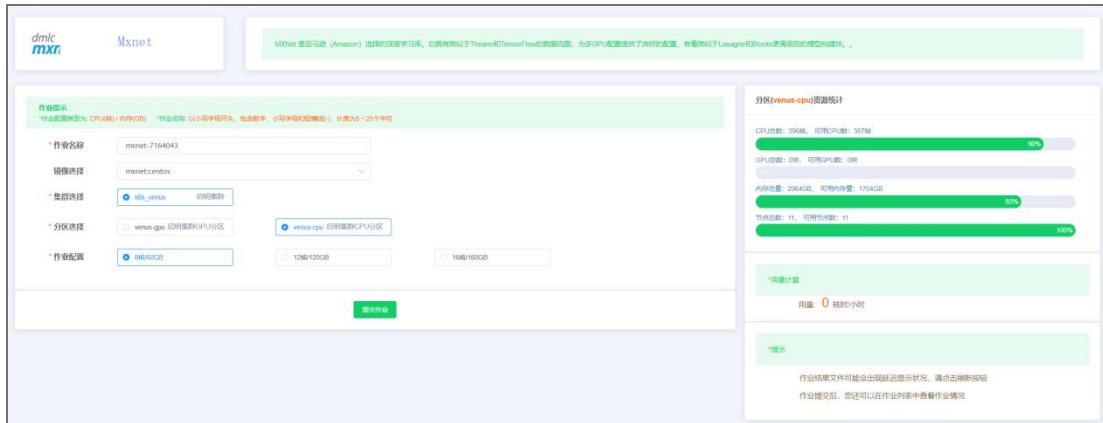


11.6.5 Mxnet 使用说明

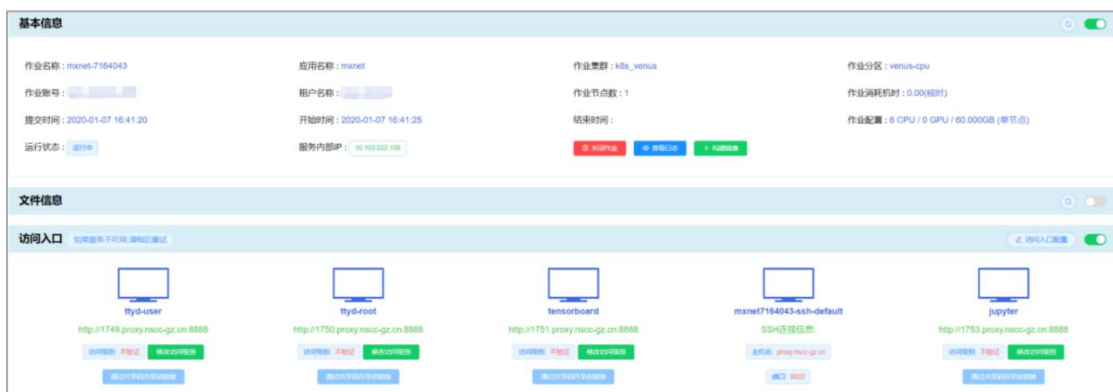
MXNet 是亚马逊 (Amazon) 选择的深度学习库。它拥有类似于 Theano 和 TensorFlow 的数据流图，为多 GPU 配置提供了良好的配置，有着类似于 Lasagne 和 Blocks 更高级别的模型构建块，并且可以在你可以想象的任何硬件上运行（包括手机）。对 Python 的支持只是其冰山一角—MXNet 同样提供了对 R、Julia、C++、Scala、Matlab 和 Javascript 的接口。下面以官方示例程序为例，演示使用 mxnet 的流程和相关步骤，其中程序和数据来源源于 [mxnet 官方网站](#)。

11.6.5.1 Mxnet 应用创建步骤

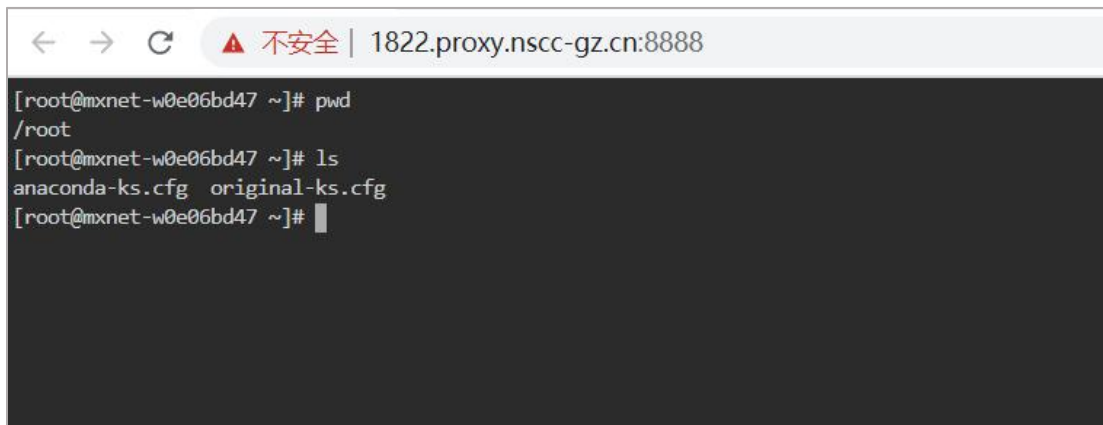
和其他应用类似，首先先提交作业：



提交成功后自动进入作业详情界面：



点击“ttyd-root”访问入口即可进入该应用：



11.6.5.2 运行 Mxnet 示例程序

示例程序位于 `/app/common/example/mxnet` 目录下，程序脚本为 `run_example.sh`。首先需要将示例程序拷贝到用户目录下，可以通过 SSH 工具远程登录作业（详见 4.2.2.2）或者直接在 web 终端上执行命令：`cp -r /app/common/example/mxnet/ ~`，将示例程序拷贝到用户目录：

```
[root@mxnet-w0e06bd47 ~]# cp -r /app/common/example/mxnet/ ~/mxnet
[root@mxnet-w0e06bd47 ~]# ls
anaconda-ks.cfg  mxnet  original-ks.cfg
[root@mxnet-w0e06bd47 ~]#
```

进入 mxnet 目录，执行./run_example.sh 命令即可运行示例程序：

```
[root@mxnet-w0e06bd47 ~]# cd ~/mxnet/example/
[root@mxnet-w0e06bd47 example]# ./run_example.sh
---run MXNET examples---
you can choose example to run
for example TextClassification,Phist
---step1 set http proxy---
step2 load modules
----begin to load modules for centos---
enter example name TextClassification
run LanguageUnderstanding
python example/cn_chinese_text_classification/text_cnn.py
020-01-09 17:38:31,943:filename text_cnn.py: lineno 88:INFO:loading data...
wget: /app/common/anaconda2/5.3.0/lib/libcud.so.1: no version information available (required by wget)
pchive: chinese_text.zip
020-01-09 17:39:04,929:filename text_cnn.py: lineno 112:INFO:Train/Valid split: 62855/1000
020-01-09 17:39:04,929:filename text_cnn.py: lineno 113:INFO:train shape: (62855, 99)
020-01-09 17:39:04,929:filename text_cnn.py: lineno 114:INFO:valid shape: (1000, 99)
020-01-09 17:39:04,929:filename text_cnn.py: lineno 115:INFO:sentence max words: 99
020-01-09 17:39:04,929:filename text_cnn.py: lineno 116:INFO:embedding size: 300
020-01-09 17:39:04,929:filename text_cnn.py: lineno 117:INFO:vocab size: 4859
17:39:05] src/operator/tensor/./matrix_op-inl.h:189: Using target_shape will be deprecated.
17:39:05] src/operator/tensor/./matrix_op-inl.h:189: Using target_shape will be deprecated.
020-01-09 17:39:05,033:filename module.py: lineno 396:WARNING:Already bound, ignoring bind()
/app/common/anaconda2/5.3.0/lib/python2.7/site-packages/monet/module/base_module.py:502: UserWarning: Parameters already initialized and force_init=False. init_params call ignored.
allow_missing_allow_missing_force_init_force_init)
020-01-09 17:39:06,034:filename module.py: lineno 494:WARNING:optimizer already initialized, ignoring...
020-01-09 17:39:06,276:filename monitor.py: lineno 143:INFO:Batch: 1 vocab_embed_0 0.057770625
020-01-09 17:39:06,276:filename monitor.py: lineno 143:INFO:Batch: 1 reshape0_0 0.057770625
020-01-09 17:39:06,276:filename monitor.py: lineno 143:INFO:Batch: 1 convolusion0_output 0.09582065
020-01-09 17:39:06,276:filename monitor.py: lineno 143:INFO:Batch: 1 activation0_output 0.061694324
020-01-09 17:39:06,276:filename monitor.py: lineno 143:INFO:Batch: 1 pooling0_output 0.18489733
020-01-09 17:39:06,276:filename monitor.py: lineno 143:INFO:Batch: 1 convolusion1_output 0.11160671
020-01-09 17:39:06,276:filename monitor.py: lineno 143:INFO:Batch: 1 activation1_output 0.08223873
020-01-09 17:39:06,276:filename monitor.py: lineno 143:INFO:Batch: 1 pooling1_output 0.21653892
020-01-09 17:39:06,276:filename monitor.py: lineno 143:INFO:Batch: 1 convolusion2_output 0.13446827
020-01-09 17:39:06,276:filename monitor.py: lineno 143:INFO:Batch: 1 activation2_output 0.10783863
020-01-09 17:39:06,276:filename monitor.py: lineno 143:INFO:Batch: 1 pooling2_output 0.24486141
020-01-09 17:39:06,276:filename monitor.py: lineno 143:INFO:Batch: 1 concat0_output 0.21682024
020-01-09 17:39:06,276:filename monitor.py: lineno 143:INFO:Batch: 1 reshape1_0 0.21682024
020-01-09 17:39:06,276:filename monitor.py: lineno 143:INFO:Batch: 1 high_fc_output 0.22311683
020-01-09 17:39:06,276:filename monitor.py: lineno 143:INFO:Batch: 1 activation3_output 0.18597256
020-01-09 17:39:06,276:filename monitor.py: lineno 143:INFO:Batch: 1 high_trans_sigmoid_output 2.0153286
020-01-09 17:39:06,276:filename monitor.py: lineno 143:INFO:Batch: 1 activation4_output 0.12366362
020-01-09 17:39:06,276:filename monitor.py: lineno 143:INFO:Batch: 1 mul0_0 0.020050405
020-01-09 17:39:06,276:filename monitor.py: lineno 143:INFO:Batch: 1 r_minusscalar0_0 0.8794962
020-01-09 17:39:06,276:filename monitor.py: lineno 143:INFO:Batch: 1 mul1_0 0.19051225
020-01-09 17:39:06,276:filename monitor.py: lineno 143:INFO:Batch: 1 plus0_0 0.2015362
020-01-09 17:39:06,276:filename monitor.py: lineno 143:INFO:Batch: 1 fullyconnected0_output 0.08949399
020-01-09 17:39:06,276:filename monitor.py: lineno 143:INFO:Batch: 1 softmax_output 0.5804934
020-01-09 17:39:06,276:filename monitor.py: lineno 143:INFO:Batch: 1 softmax_backward_data 0.4956643
020-01-09 17:39:06,276:filename monitor.py: lineno 143:INFO:Batch: 1 softmax_backward_label 72.40877
```

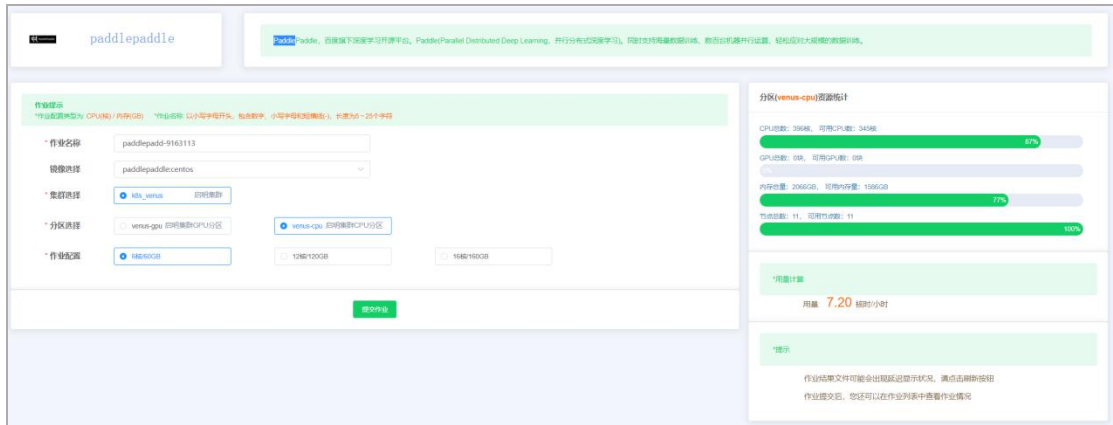
出现以上界面表示示例程序运行成功。

11.6.6 PaddlePaddle 使用说明

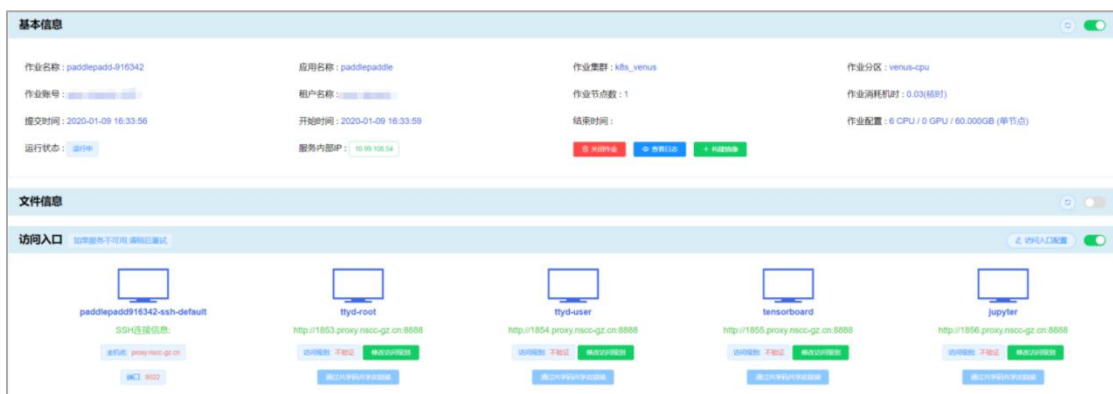
PaddlePaddle 是百度公司开源的一项易学易用的分布式深度学习平台，它提供了丰富的算法服务，支持诸如机器视觉、自然语言理解、推荐系统等领域的算法服务。并且具有易用、高效、灵活和扩展性良好的特性。下面以线性回归模型为例，演示使用 PaddlePaddle 进行深度学习的步骤和方法。数据来源和示例程序来自 [PaddlePaddle 官方网站](#)。

11.6.6.1 PaddlePaddle 应用创建步骤

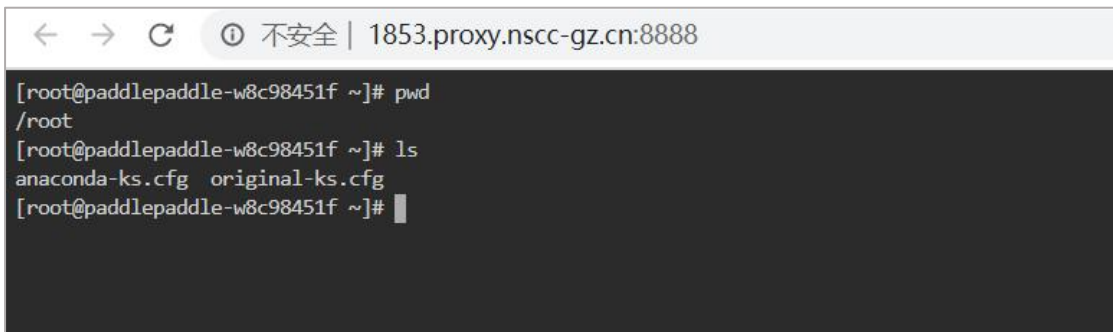
和其他应用类似，首先先提交作业：



提交成功后自动进入作业详情界面：



通过“ttyd-root”等访问入口访问应用：



能够进入类似以上界面表示应用已创建成功。

11.6.6.2 运行 PaddlePaddle 示例程序

示例程序位于 `/app/common/example/paddlepaddle` 目录下，程序脚本为 `run_example.sh`。首先需要将示例程序拷贝到用户目录下，可以通过 SSH 工具远程登录作业（详见 4.2.2.2）或者直接在 web 终端上执行命令：`cp -r /app/common/example/paddlepaddle ~`，将示例程序拷贝到用户目录：

```
← → ↻ ⚠ 不安全 | 1853.proxy.nscg-gz.cn:8888
[root@paddlepaddle-w8c98451f ~]#
[root@paddlepaddle-w8c98451f ~]#
[root@paddlepaddle-w8c98451f ~]# ls
anaconda-ks.cfg  original-ks.cfg
[root@paddlepaddle-w8c98451f ~]# cp -r /app/common/example/paddlepaddle ~
[root@paddlepaddle-w8c98451f ~]# ls
anaconda-ks.cfg  original-ks.cfg  paddlepaddle
[root@paddlepaddle-w8c98451f ~]#
```

进入 paddlepaddle 目录，执行 ./run_example.sh 命令即可运行示例程序：

```
[root@paddlepaddle-w8c98451f ~]# ls
anaconda-ks.cfg  original-ks.cfg  paddlepaddle
[root@paddlepaddle-w8c98451f ~]# cd paddlepaddle/
[root@paddlepaddle-w8c98451f paddlepaddle]# ls
examples  run_example.sh
[root@paddlepaddle-w8c98451f paddlepaddle]# ./run_example.sh
--run Paddlepaddle examples--
you can choose example to run
for example fit_a_line,recognize_digits
---step1 set http_proxy---
step2 load modules
---begin to load modules for centos---
enter example name:fit_a_line
run fit_a_line example
python examples/01_fit_a_line/train.py
In the fluid 1.0, the trainer and inferencer are moving to paddle-fluid.contrib
Cache file /root/.cache/paddle/dataset/uci_housing/housing.data not found, downloading https://archive.ics.uci.edu/ml/machine-learning-databases/housing/housing.data
=====]Train cost, Step 0, Cost 779.886824
test cost, Step 0, Cost 302.089734
rain cost, Step 10, Cost 584.032104
rain cost, Step 20, Cost 829.325562
rain cost, Step 30, Cost 645.146362
rain cost, Step 40, Cost 422.302551
rain cost, Step 50, Cost 422.713562
rain cost, Step 60, Cost 446.533783
rain cost, Step 70, Cost 542.198364
rain cost, Step 80, Cost 522.212891
rain cost, Step 90, Cost 348.454987
rain cost, Step 100, Cost 325.884875
```

出现以上界面表示示例程序运行成功。

11.6.7 Spark 使用说明

“天河二号”星光平台支持用户运行自制容器镜像，且赋予了用户容器内 root 权限。用户可通过如下方式来使用 Spark 应用：

- 1、自行编辑和发布大数据应用或者采用官方如 databricks、Cloudera 等公司提供的大数据应用来使用其开源产品(该方案需要用户使用自定义应用镜像并且针对该镜像制作相应的平台应用，难度较大，建议高级用户使用)
- 2、直接使用平台提供的官方 Apache-Spark (包括 Spark-Master、Spark-Worker 两个应用)和微软的 MMLSpark。MMLSpark 为单机版，官方 Apache-Spark 支持集群模式。

使用 MMLSpark 和官方 Apache-Spark 应用(以下简称 Apache-Spark)都需要对 Spark 应用本身要有一定了解。由于在星光平台上运行的大数据应用都是在容器中运行的，与在物理机上运行是有所不同的，如容器可能会迁移、重启，因此用户程序和用户数据需要配置到用户的共享目录。另外，由于平台的特殊性，运行 Spark 应用的底层文件系统不是 HDFS，而是 Lustre 文件系统。

注：MMLSpark 是微软提供针对机器学习环境的 Spark 版本 (<https://github.com/Azure/mmlspark>),关于 mmlspark 可参考论文：<https://arxiv.org/pdf/1810.08744.pdf>

在星光平台上的 Spark 应用配置信息如下：

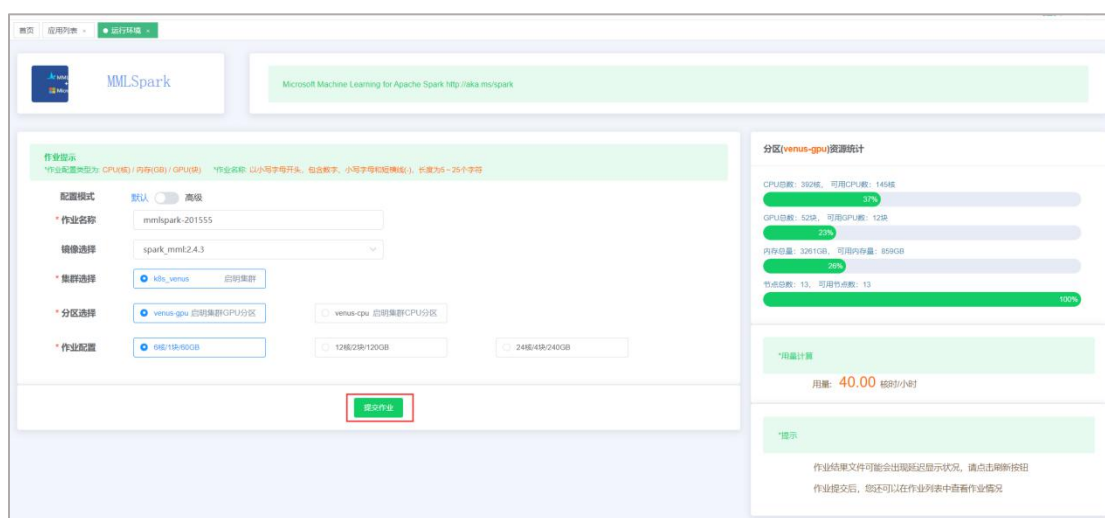
	Spark	Python	Java	OS
Apache-Spark	2.4.3	3.7	1.8.0	Alphine-Linux
MMLSpark	2.4.3	3.7	1.8.0	Ubuntu 16.04

以下分别介绍 MMLSpark 和 Apache-Spark 的使用。

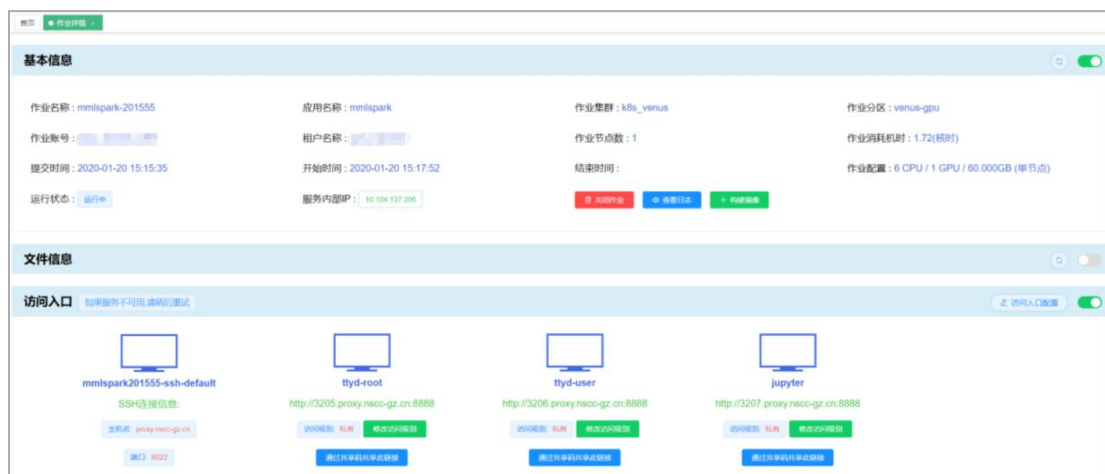
11.6.7.1 MMLSpark 使用说明

使用步骤：

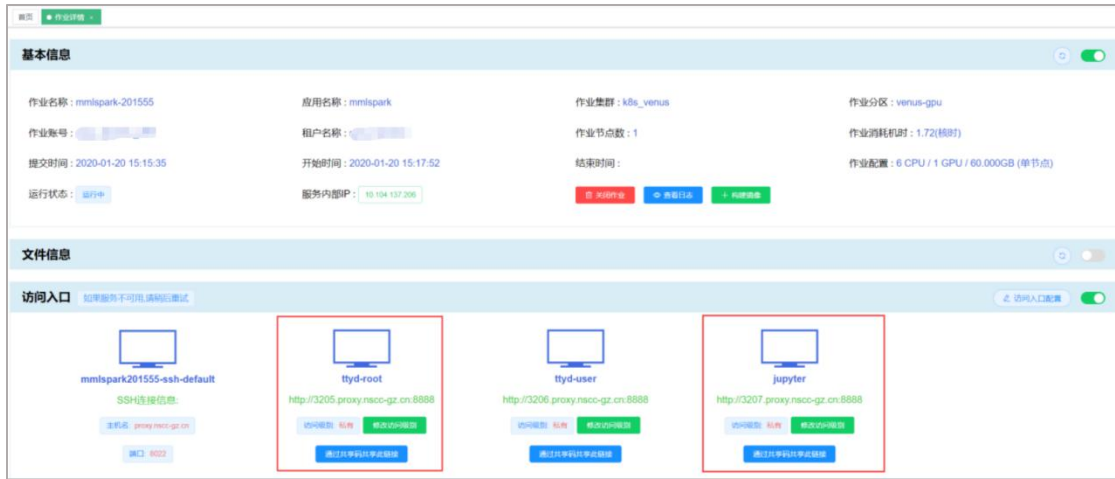
步骤 1：和其他应用类似，首先提交作业：



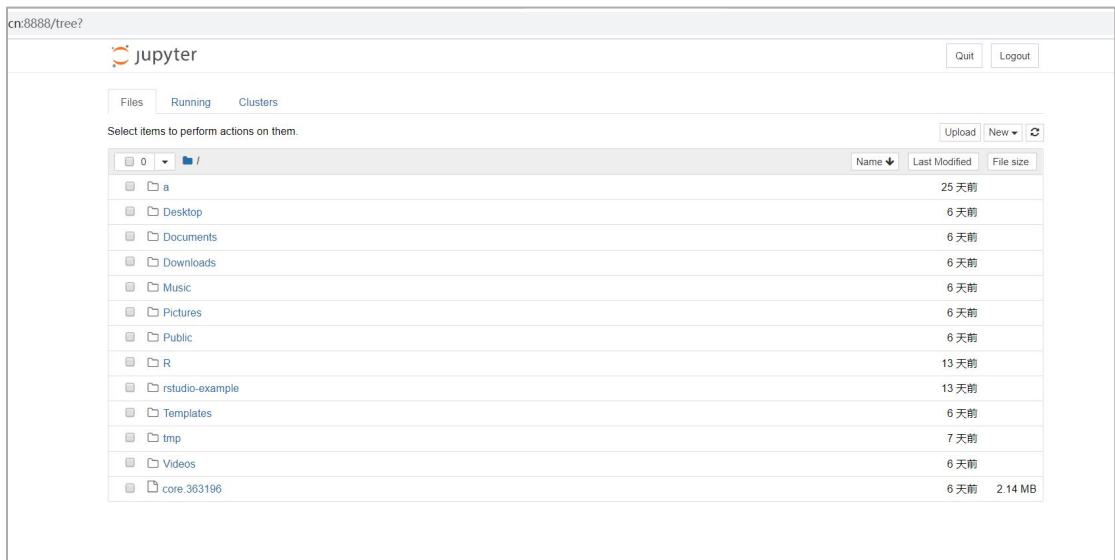
提交成功后自动进入作业详情界面：



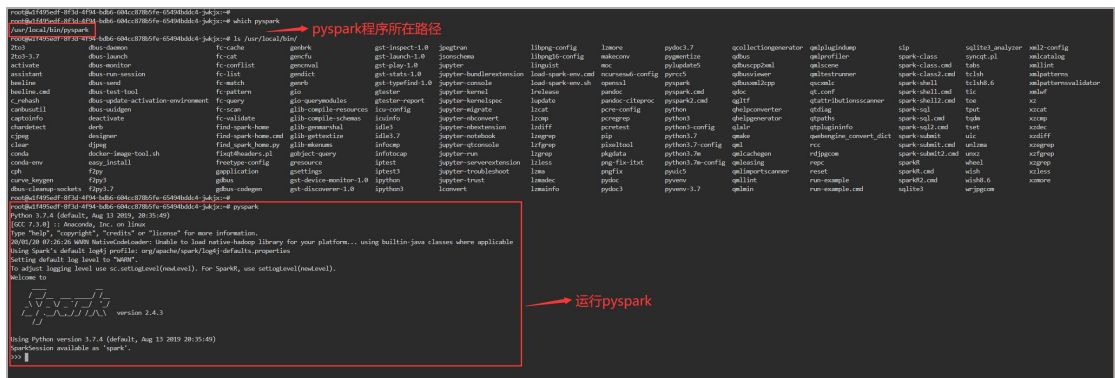
步骤 2：通过访问入口访问应用，以 jupyter 和 ttyd-root 访问入口为例：



使用 jupyter 访问入口访问应用（密码为 jupyter）：



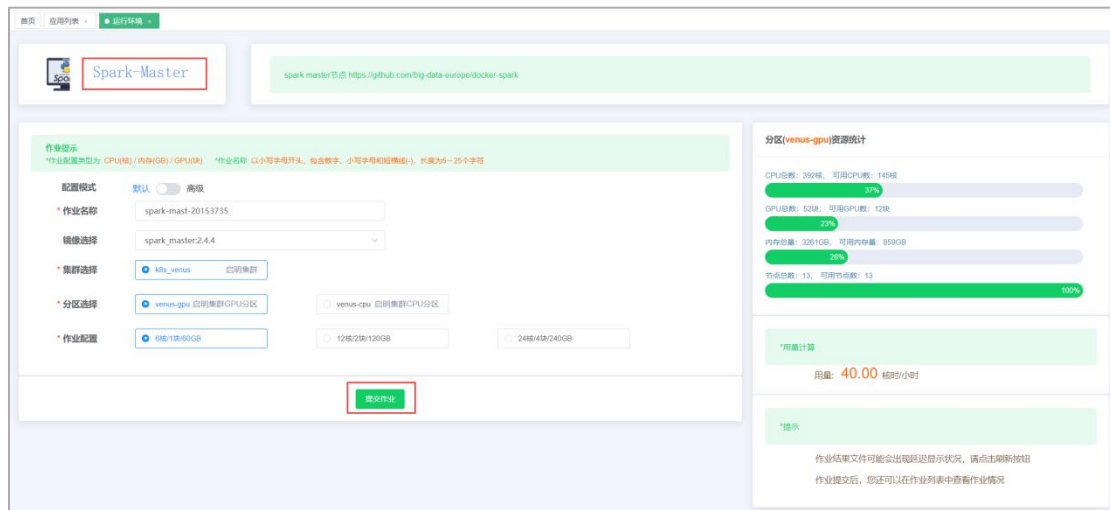
使用 ttyd-root 访问入口访问应用：



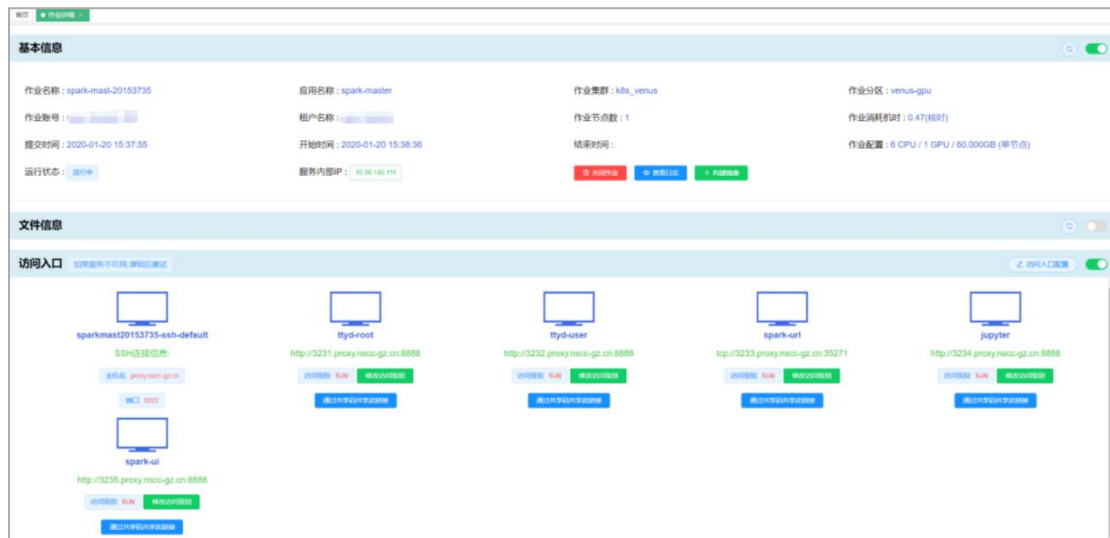
11.6.7.2 Apache-Spark 使用说明

Apache-Spark 支持集群模式，以下演示一个 Spark 集群的创建步骤。

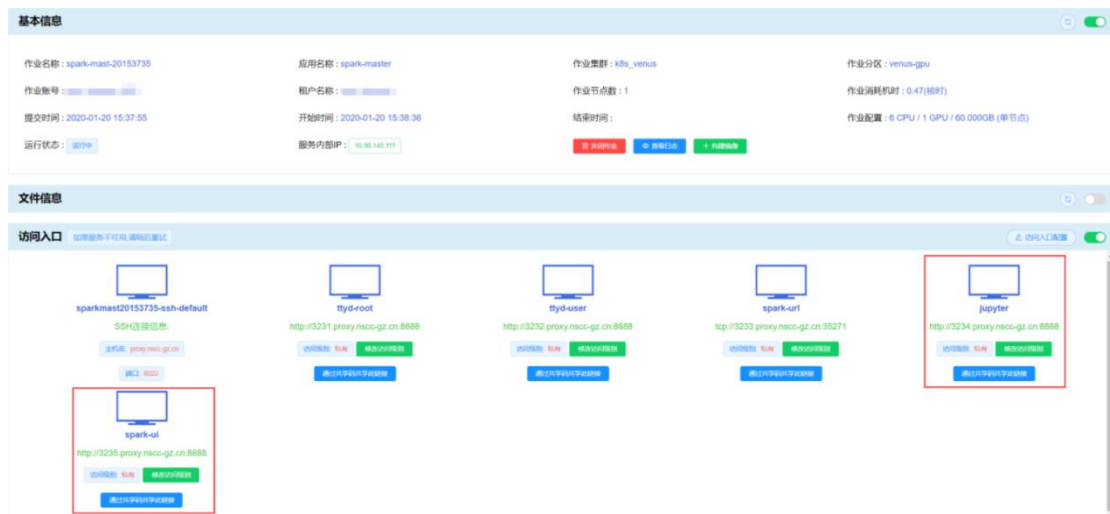
步骤 1: 提交 Spark-Master 应用作业:



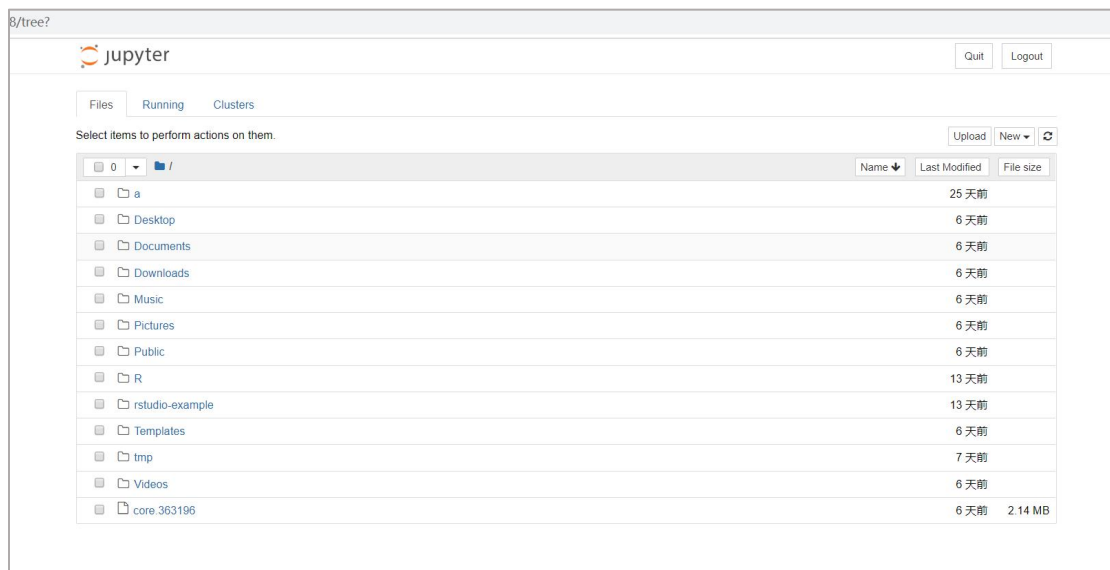
提交成功后自动进入作业详情界面:



步骤 2: 通过访问入口访问 Apach-Spark 应用, 以 jupyter 和 sprak-ui 访问入口为例:



使用 jupyter 访问入口访问应用（密码为 jupyter）：

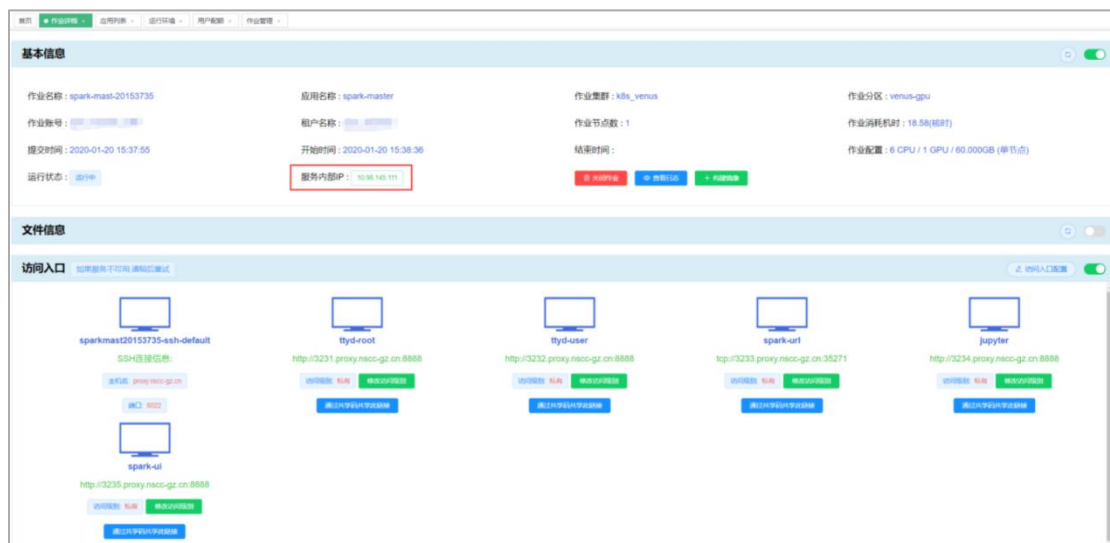


使用 sprak-ui 访问入口访问应用：

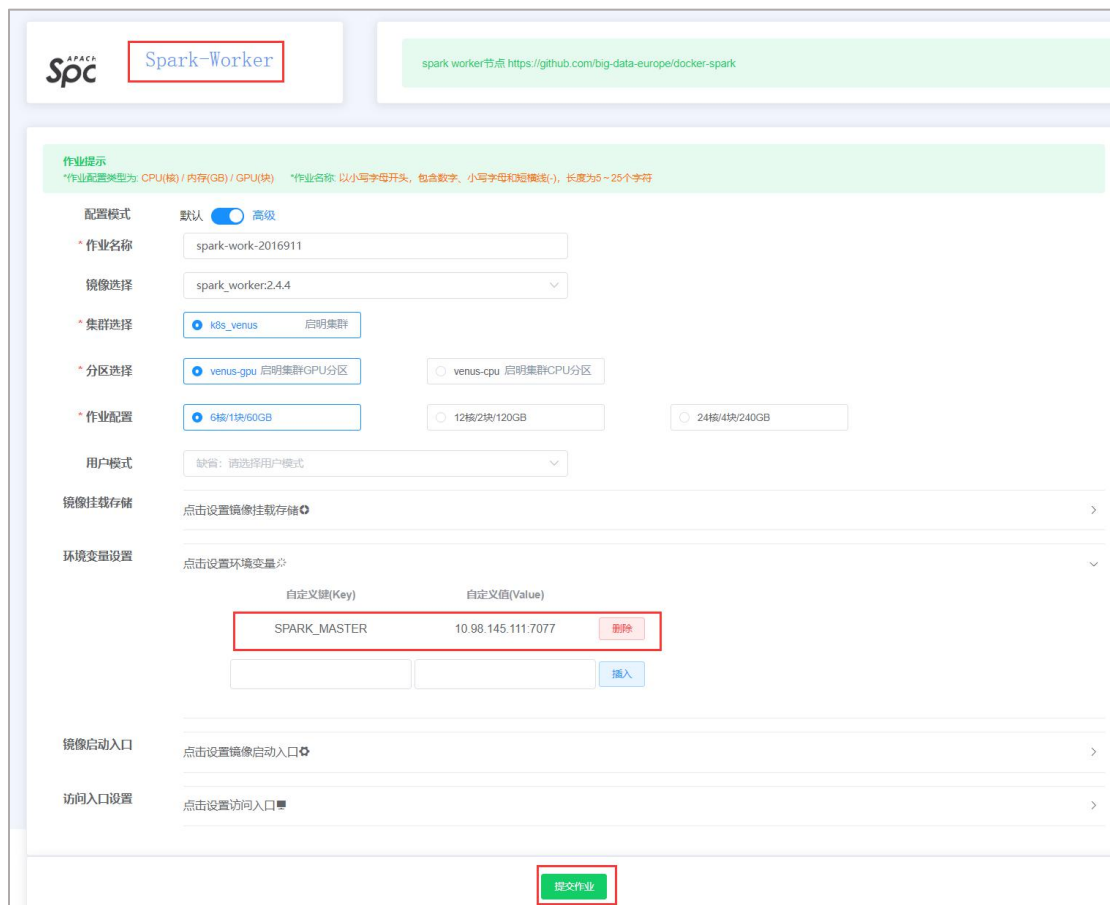


步骤 3：提交 Spark-Worker 应用作业

Spark-Worker 节点依赖于 Spark-Master 节点，创建 Spark-Worker 节点时需要使用高级配置模式配置环境变量 SPARK_MASTER，SPARK_MASTER 的值为 Spark-Master 的内部服务 IP:端口号（一般为 7077），Spark-Master 的内部服务 IP 可以在 Spark-Master 的作业详情界面查看：

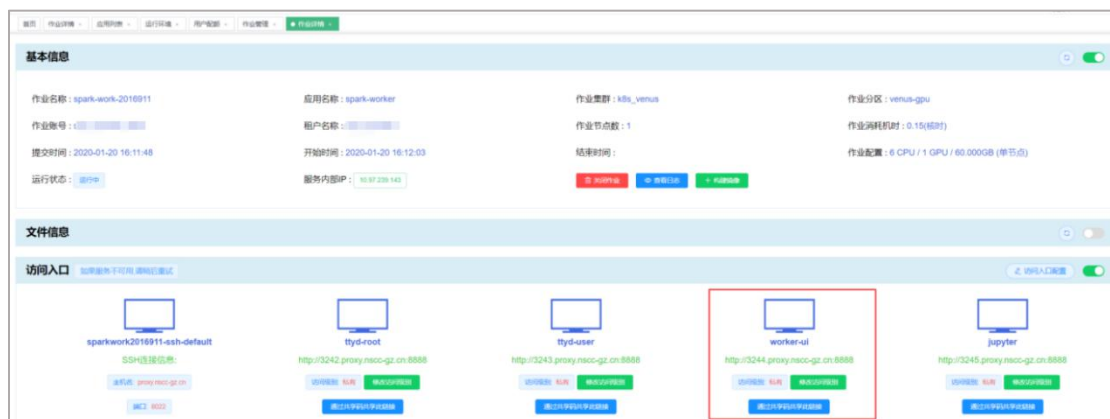


Spark-Worker 的作业配置类似下图：

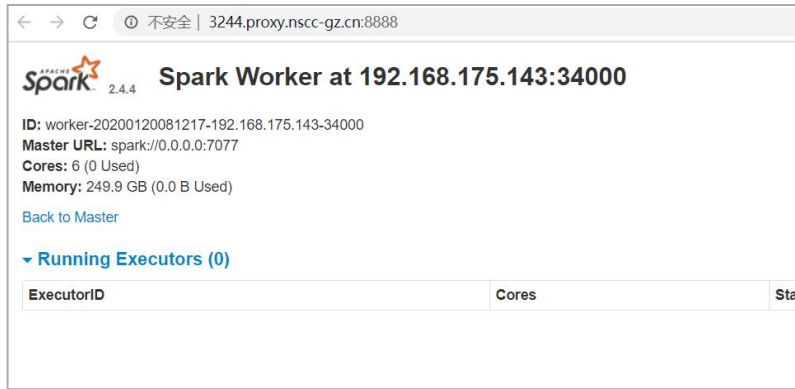


点击“提交作业”完成 Spark-Worker 的创建。

步骤 4：通过访问入口访问 Spark-Worker 应用，以 worker-ui 访问入口为例：



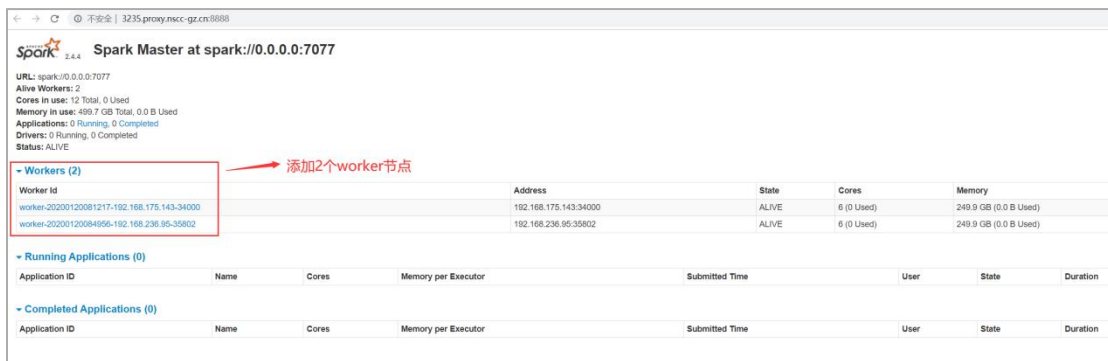
出现类似以下界面表示 Spark-Worker 创建成功：



步骤 5: 在 Spark-Master 节点查看集群信息



出现类似以上界面，表明 worker 节点已成功加入集群，Spark 集群创建成功。用户可以参考步骤 3~5 继续向集群中添加 worker 节点，如下为添加了 2 个 worker 节点的 Spark 集群信息：



11.6.7.3 注意事项

用户在使用 Spark 应用时，有以下几点需要注意：

1. Spark 集群版应用创建后 spark 相关 URI-IP 为集群内部 IP，不可外部访问(如 worker url 等链接不支持浏览器中直接打开)
2. 底层文件系统不是 HDFS 文件系统，而是 Lustre 文件系统

3. Apache-Spark 应用默认安装路径为: /spark, 在 Spark-Worker 应用中没有将该路径加载到环境变量, 若用户有需要, 可使用命令自行加载: `export PATH=/spark/bin:$PATH`
4. 若用户想自定义配置 Apache-Spark, 可自行修改和配置 Spark-Master 应用中 /master.sh 文件和 Spark-Worker 应用中 /worker.sh 文件 (需要从 ttyd-root 登录, 以 root 身份修改, 且需要关闭现有的配置)
5. Apache-Spark 的操作系统为 alpine-linux, 且已安装基本的程序开发、编译工具, 如 gcc、python 等。在 alpine linux 中软件的安装、卸载操作使用的是 apk 命令, 关于 alpine linux 的更多信息可参考 <https://alpinelinux.org/>。