



# “星光” 用户使用手册

Ver 0.5

## 目录

|                            |    |
|----------------------------|----|
| 一、 星光简介 .....              | 4  |
| 二、 访问与注册 .....             | 5  |
| 1. 注册与登录 .....             | 5  |
| 2. 账号申请 .....              | 6  |
| 2.1 基本信息 .....             | 7  |
| 2.2 用户详情 .....             | 8  |
| 2.3 计算资源需求 .....           | 9  |
| 3. 账号绑定 .....              | 10 |
| 三、 首页总览 .....              | 12 |
| 四、 作业与应用 .....             | 13 |
| 1. 应用中心 .....              | 13 |
| 1.1 应用列表 .....             | 13 |
| 1.2 应用管理 .....             | 13 |
| 1.2.1 表单编辑 .....           | 14 |
| 1.2.2 运行编辑 .....           | 16 |
| 1.2.3 基本信息设置 .....         | 17 |
| 1.2.4 应用发布 .....           | 18 |
| 1.3 应用运行 .....             | 18 |
| 2. 作业详情 .....              | 19 |
| 3. 作业管理 .....              | 21 |
| 五、 存储管理 .....              | 22 |
| 六、 在线提问 .....              | 23 |
| 七、 用户中心 .....              | 24 |
| 八、 资源中心 .....              | 25 |
| 1. 申请记录 .....              | 25 |
| 2. 高性能计算 .....             | 25 |
| 3. 机时信息 .....              | 26 |
| 4. 资源调整 .....              | 26 |
| 九、 大数据、深度学习应用使用说明 .....    | 26 |
| 1. 应用通用说明 .....            | 27 |
| 1.1 大数据、深度学习应用简介 .....     | 27 |
| 1.2 ssh 工具远程登录作业 .....     | 29 |
| 1.3 module 加载设置 .....      | 32 |
| 1.4 网络代理设置 .....           | 34 |
| 1.5 数据上传下载 .....           | 34 |
| 1.6 自定义安装用户 python 包 ..... | 39 |
| 1.7 自定义源码编译安装软件 .....      | 39 |
| 2. Spark 使用说明 .....        | 40 |
| 3. Rstuiot 使用说明 .....      | 45 |

|                            |    |
|----------------------------|----|
| 4. Galaxy 使用说明 .....       | 50 |
| 5. Tenorflow 使用说明.....     | 54 |
| 6. Slurm 使用说明.....         | 58 |
| 7. Caffe 使用说明 .....        | 60 |
| 8. Mxnet 使用说明 .....        | 64 |
| 9. Cntk 使用说明.....          | 68 |
| 10. PaddlePaddle 使用说明..... | 71 |
| 11. Torch 使用说明 .....       | 76 |

## 一、星光简介

“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、大数据分析与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。

“星光”仍然在不断完善中，有任何平台使用问题和建议，请直接使用平台的[在线提问系统\(见本文 第六部分\)](#)，进行反馈。

## 二、访问与注册

访问地址：<https://starlight.nscc-gz.cn>

推荐使用 **Google Chrome** 浏览器访问星光系统。



图 1 星光登陆页面

### 1. 注册与登录

无论用户是否已经有“天河二号”超级计算机的账号，用户需要通过手机或者邮箱在“星光”上注册。注册成功后，使用“邮箱”或者“手机”登录平台。

如果是通过邮箱地址进行注册的用户，请及时点击邮箱里的激活邮件进行激活，否则无法进行正常的使用。手机/邮箱同时用于系统通知服务，请确保手机、邮箱的通畅。



图 2 星光 邮箱、手机 注册页面

## 2. 账号申请

如果您从来没有使用过“天河二号”，请点击“申请系统账号”，填写申请表进行系统账号的申请。申请通过之后，便可以直接登录使用“星光”，跳过下一小节的“账号绑定”过程。在申请的过程中，您可以登录平台，查看申请的状态。

\***用户需要申请系统账号，该账号可用在终端命令行方式下直接访问计算资源。账号名由平台管理员手动分配，格式一般是“单位缩写\_名字缩写\_数字索引”。手机或者邮箱并不能用作登录终端时的用户名。在手册、星光平台中出现的用户名、账户名都指的是由超算中心分配的系统账号。集群资源的直接访问基本通过该系统账号和星光登录密码来完成，除非本手册有特别的说明。**



图 3 账号申请、绑定页面

### 申请表填写说明:

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。

#### 2.1 基本信息

项目相关信息对于试用用户的配额分配有重要影响，对于国家重点项目以及其他有重要意义的项目，中心会从试用额度、机时优惠、技术服务等多方面进行支持。

基本信息

\* 联系人

\* 组织/公司

\* 联系邮箱

项目名称

项目信息对于试用用户的配额分配有重要影响，对于国家重点项目以及其他有重要意义的项目，中心会从试用额度、机时优惠、技术服务等多方面进行重点支持。

项目负责人

项目来源

项目简介

总经费（万元人民币）

图 4 申请表-基本信息页面

## 2.2 用户详情

中心根据应用所属的科学领域建立了六个方向的服务平台。请选择一个与您应用方向接近的服务平台，中心在后续的技术服务中会安排相关领域的技术支持人员与您对接。

用户详情

完善用户详情能够帮助加快用户申请表的处理以及提高用户的试用资源额度。

服务平台

生物医药健康

客户类型

个人

联系电话

通讯地址

推荐人信息

请填写推荐人信息

图 5 申请表-用户详情页面

### 2.3 计算资源需求

软件需求的填写有助于工作人员了解用户所需的大致的计算环境，以便开通相应的集群资源。最大计算规模需要用户根据项目情况评估可能用到的计算规模，工作人员基于此调整用户资源配置。如果有特殊的需求，也请注明。

计算资源需求

---

软件需求  
请输入软件需求

特殊需求说明  
请输入特殊需求说明

最大计算规模(核数)  
/

图 6申请表-计算资源需求页面

### 3. 账号绑定

如果您以前曾经开通过系统账号（使用过“天河二号”），您可以直接绑定该账号。



图 7账号申请、绑定页面

绑定账号通常只需要填写两个信息，分别是 **账户名** 和 **密钥文件**。这两个信息可以查询申请账户时收到的 **开户信息邮件**，邮件中包含用户的账户名（如 xyz\_npq\_1，一般格式是 单位缩写\_姓名缩写\_数字）以及密钥文件。密钥文件一般在 开户信息邮件 的附件中，格式一般是 账户名.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，请使用原始的文件。

A<sup>文</sup>

### 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**您注册天河星光所用的邮箱或手机。不清楚的用户请检查“客户服务 service.nscc@nscc-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 xyz\_npq\_1

请选择密钥文件  id\_rsa\_z...uaishuai

私钥口令

**绑定**

图 8 账号绑定详情页面

**私钥口令** 默认都是空值，除非您通过终端连接集群后重新生成了 SSH KEY 并进行了相关的设置。

### 三、 首页总览

首页集中展示用户相关信息，主要有：

1. 新闻、公告。
2. 用户文档。
3. 信息统计，包括机时使用、作业统计、存储使用等信息。
4. 资源及配额信息，包括可用的集群和分区、集群配额等。

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）。

The screenshot displays the Xingguang platform's main interface, divided into several sections:

- Top Left:** A banner for the "2018 China Supercomputer Conference (CNCC2018) 大数据推动数字经济" (Big Data Drives The Digital Economy). Below it, real-time resource usage statistics are shown: 机时 (Machine Hours) - 剩余无限机时 (Remaining unlimited machine hours), 已用 1.55 万机时 (Used 1.55 ten thousand machine hours); 作业 (Jobs) - 运行 0 (Running 0), 排队 0 (Queued 0).
- Top Right:** A sidebar with links to "指南" (Guidelines), "常用应用" (Common Applications) (listing dcv2017, ansysfluentgui, schrodinger-debug, workbenchgui), and "查看详情" (View Details).
- Middle Left:** A "存储配额" (Storage Quota) table showing storage partitions and their configurations.
- Middle Right:** A "可用集群" (Available Clusters) table listing clusters and their available partitions.
- Bottom:** Two tables under the heading "资源配额" (Resource Quota): one for cluster-level quotas and another for specific node-level quotas.

图 9 星光首页总览

## 四、作业与应用

### 1. 应用中心

#### 1.1 应用列表

点击导航栏“应用列表”将显示当前用户所有可用的应用。根据应用常用的领域，应用中心将应用进行了标签分类，如“工业仿真”、“生物医药”、“人工智能”等。用户可以根据分类查看相应的应用，也可以根据应用名称进行模糊搜索。

由于应用个数较多，应用列表提供收藏夹功能，用户可收藏常用应用。点击应用图标右上角的五角星可以将应用加入收藏夹，再次点击取消收藏。收藏的应用整理在“收藏夹”标签页中。

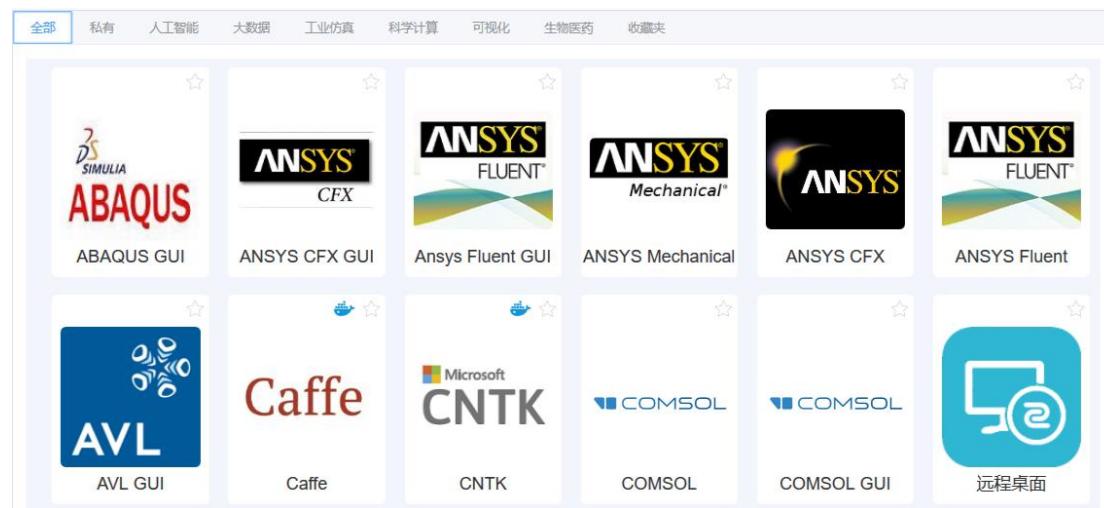


图 10 星光应用列表页面

#### 1.2 应用管理

应用列表中“私有”标签页用来放置用户自己创建的应用。用户可以在“应用管理”中添加自己的应用。



图 11 应用管理页面

为了方便用户创建应用，平台提供了基础应用模板，用户可以在这个模板基础上修改自己的应用，如图 11 所示。载入模板之后，进入应用编辑页面，以 lammmps 模板为例，界面如图 12 所示。

## 应用管理组件说明：

### 1.2.1 表单编辑

应用编辑页面提供对应用表单的编辑功能，通过“添加组件”添加 html 控件。常用的有 info, text, rfb, list, listdyn, group。在同一个应用表单页面中的控件都有一个唯一的 ID，后台能通过该 ID 来获取控件中设置的内容/值（value）。当添加一个组件之后，点击该组件，网页右边有相应的属性编辑框用来编辑该组件的属性，比如通过 Label 属性可以修改控件在网页表单上显示的文字。

Info 组件是个文本编辑工具，一般用于发布者编辑应用的说明，使用者看到的是不可编辑的文本信息。

Text 组件是文本输入框，用于应用使用者的文本参数输入。Default 属性用来设置控件的默认值。Rules 可设置正则表达式，用于制定文本的输入规则。请参考 lammmps 模板中的“作业名称”了解 Text 组件的具体运用。

Rfb 组件是文件选择器，用于应用使用者选择文件或文件夹，后台可以获取该文件/文件夹所在的路径。

List 组件是列表组件，用于预设选项供应用使用者选择。通过增加/删除 Label-value 对来添加/删除列表选项。Listdyn 是动态列表控件，其列表的 label-value 是通过另一个应用动态获取的。在 lammmps 模板中“软件版本”组

件使用了应用 module\_avail 来获取版本信息。module\_avail 使用 app=lammps&target=hpc 作为参数在后台运行了一个脚本查询了 lammps 的所有版本信息。

Group 组件是一个容器，用来盛放其它的组件，主要用于页面布局。

# 应用编辑

设置

提交

添加组件 ▾

↑ ↓ 删除

File - Edit - Insert - View - Format - Table -

B I U S E H M “ ” & < > I x x & < > { }

- H B H & G & A & A -

## Lammps Molecular Dynamics Simulator

[homepage](#)

5 WORDS POWERED BY TINYMCE

↑ ↓ 删除

作业名称 :

lammps

↑ ↓ 删除

Case Dir :

(点击选择文件夹 , 包含了需要的输入数据文件)

↑ ↓ 删除

输入文件 :

(点击选择输入文件)

↑ ↓ 删除

CPUs :

40

↑ ↓ 删除

Nodes :

2

↑ ↓ 删除

软件版本:

Submit

属性编辑栏

图 12 应用编辑页面

### 1.2.2 运行编辑

在“应用编辑”页面中点击最下方的“submit”，弹出“运行配置”对话框，在这里设定应用的运行脚本和作业脚本。当使用者使用应用时，在应用表单页点击提交，将触发后台运行“运行脚本”。在 lammps 模板中，运行脚本主要是设置了环境变量、通过 module 加载了指定 version 的 lammps 以及通过 sbatch/yhbatch 向资源管理系统提交了作业脚本（运行脚本里的\$1），最终计算资源运行作业脚本。在这些脚本里面，网页中控件的内容/值是通过 \$ + 控件 id 来获取的，比如 \$version 获取的是 lammps 版本号，\$nodes、\$cpus 分别是节点数和 cpu 数目。

作业脚本非常简单，通过 yhrun 运行了 lammps 的可执行文件 lmp\_intel\_cpu。当设置完脚本之后，点击确定回到应用编辑页面。



图 13 应用编辑-运行配置页面 1

Jobfile

```
#!/bin/sh  
yhrun -N $nodes -n $cpus lmp_intel_cpu < $infile
```

图 14 应用编辑-运行配置页面 2

### 1.2.3 基本信息设置

点击“设置”按钮，进入“基本信息”设置。在这里设置图标、名字、以及标题（在“应用列表”中查看应用时，图标下面显示的是标题）。注意，应用的名字不能重复，否则无法提交应用。另外可以通过打开访问权限控制，通过设置白名单或黑名单来让目标用户可用或者不可用。

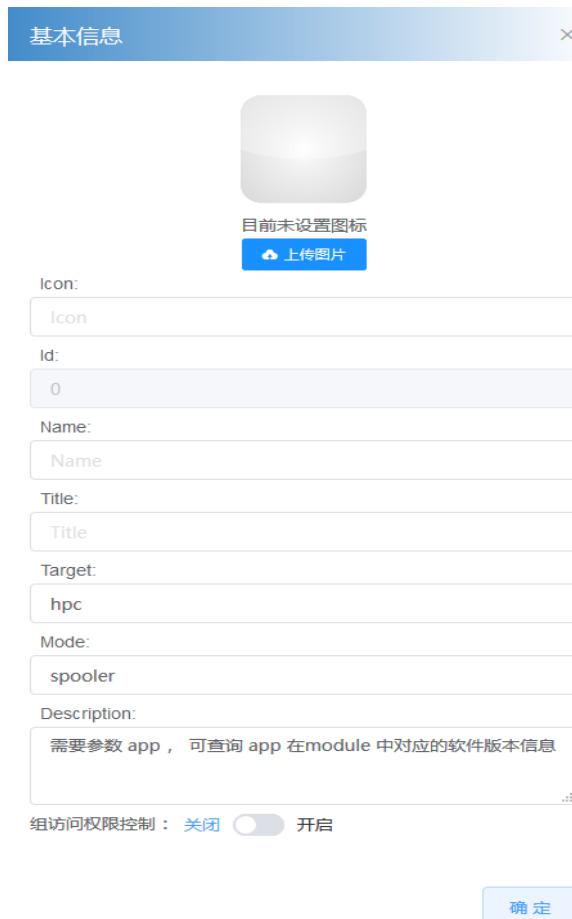


图 15 应用编辑-基本信息设置页面

#### 1.2.4 应用发布

完成表单编辑、脚本设置和基本信息设置后，点击“提交”便保存了该应用。这个时候，应用的发布者在“应用列表”的“私有”标签中可以看到并运行该应用。如果应用发布者想让整个“星光”平台的用户都能看到该应用（默认是开放给全部用户，除非在“基本信息”设置里面进行了黑名单或白名单设置），可以在“应用管理”中“发布”该应用，如图 16 所示。经过“星光”管理员审核通过之后，该应用便可供其他人使用。

| LOGO        | 名称                | 标题                | 目标           | 模式      | 说明                          | 状态  | 操作   |
|-------------|-------------------|-------------------|--------------|---------|-----------------------------|-----|--|
| SCHRODINGER | schrodinger-debug | Schrodinger-debug | hpc:tianhe2d | spooler | Schrodinger是分子模拟软件，广泛用于生... | 未处理 | <button>Enter</button> <button>更多</button><br><br><a href="#">发布</a> <a href="#">编辑</a> <a href="#">删除</a> |

共1条 10条/页 < 1 > 前往 1 页

图 16 应用管理列表

### 1.3 应用运行

点击应用图标，即可进入应用表单页面。根据应用运行的需要，填写参数表单，点击“提交”，页面自动跳转至作业详情页面。平台会根据当前用户的配额和机时判断用户是否可继续提交作业，如图 18 所示。

The screenshot shows the configuration form for the RStudio application. It includes fields for job name, cluster selection, job configuration, mirror selection, and a summary section.

作业配置类型: CPU(核) / 内存(GB)

作业名称: 唯一,不可重复

集群选择: 天河 CPU 集群

作业配置:

- 4核/12GB (selected)
- 6核/18GB
- 8核/24GB
- 12核/36GB
- 18核/54GB
- 22核/60GB

镜像选择: Rstudio Centos

费用: 4 核时/小时

应用教程:

- rstudio视频教程
- rstudio文档教程

您的机时不足,请及时充值

查看详情

提交

图 17 应用表单页面 1



图 18 应用表单页面 2

## 2. 作业详情

在作业详情页，用户可以看到作业运行的详细状态。详情页分成三部分：

1. 基本信息，展示作业运行所在的集群、运行时间，并可对作业进行关闭操作。
2. 文件信息，展示与作业相关的文件，包括日志文件以及应用的运行输出文件。注意，作业实际运行的工作路径一般并不同于输入文件所在的路径。为了避免输入文件夹中的其它文件可能对计算造成影响，系统会自动建立一个工作目录，路径显示在文件信息地址栏。



图 19 作业详情-文件信息页面

3. 访问入口，提供应用创建的集群/服务/资源的入口。

由于应用的不同，对应的作业详情页面会有所差别。深度学习/大数据应用目前的存储尚未接入，在详情页面看不到文件信息，这些应用的存储可以通过访问入口提供的 ssh 连接信息，通过 sftp 进行数据传输。具体请查看第九章。



图 20 作业详情-访问入口页面

应用的访问入口，用户可根据需要修改访问级别，控制访问入口的访问权限，如图 21 所示。应用的访问权限分三个层次：



图 21 访问入口权限修改页面

### 1. 无限制

无需验证，任何人都可直接访问。

### 2. 网站用户

登陆验证，“星光”平台的登录用户可访问。

### 3. 私有

只有作业的运行者可以访问。但可通过设置共享码的方式，提供给平台的其他登陆用户使用。点击“通过共享码共享此链接”，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陆入口。



图 22 访问共享-验证码输入页面

### 3. 作业管理

作业管理用于管理历史和正在运行的作业。用户可以通过作业状态、名称进行查询。查看详情可以进入作业详情界面，见上一节“作业详情”。历史作业是用户过去一个月的历史作业。

The screenshot shows a user interface for job management. On the left, there is a sidebar with several menu items: 应用列表, 应用管理, 作业管理 (highlighted with a red box), 存储管理, 在线提问, 用户中心, and 资源中心. The main area has two tabs at the top: 运行作业 (selected) and 历史作业. Below the tabs is a blue '刷新' (Refresh) button. A table lists jobs with columns: jobid/sid, 作业名称 (Job Name), 状态 (Status), 提交时间 (Submission Time), 开始时间 (Start Time), and 操作 (Operations). Two rows are visible:

| jobid/sid                         | 作业名称             | 状态  | 提交时间                | 开始时间                | 操作  |
|-----------------------------------|------------------|-----|---------------------|---------------------|---|
| v100-165<br>tvccZFkHjbQrpTgJut    | duzong-buyaoshan | 运行中 | 2018-11-15 09:36:08 | 2018-11-15 09:36:14 | <button>查看详情</button> <button>关闭作业</button> |
| tianhe-175<br>yxikkccIVPAvbjpolRU | rstudio-test     | 运行中 | 2018-11-19 11:32:22 | 2018-11-19 11:36:54 | <button>查看详情</button> <button>关闭作业</button> |

图 23 作业管理页面 1

用户提交的作业可能会提交到不同的集群，但都可以在“作业管理”中看到所有作业的状态。一般情况下，用户不需要去关心作业具体提交到了哪一个集群。但如果用户想要登录对应集群去进行终端操作，也可以在列表中看到作业实际运行的集群以及相应的作业 id。

This is a zoomed-in view of the job table from Figure 23. It highlights a specific row for a job named 'tianhe-2d-6901242'. The 'jobid/sid' column contains two parts: 'tianhe-2d-6901242' (in a blue box) and 'WsVGhoLoRuDgJyotjb' (in a red box). The '作业名称' column shows '集群-所在集群的作业id' above the red box and '星光平台的作业id' below it. The '状态' column shows a green '已结束' (Completed) button.

| jobid/sid                               | 作业名称                      | 状态  |
|---|---------------------------|-----|
| tianhe-2d-6901242<br>WsVGhoLoRuDgJyotjb | 集群-所在集群的作业id<br>星光平台的作业id | 已结束 |

图 24 作业管理页面 2

## 五、 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。

目前尚不支持对文件的批量上传下载，推荐使用 sftp 传输工具进行大文件传输。

大数据、深度学习应用的数据管理尚未接入，请查阅第九章。

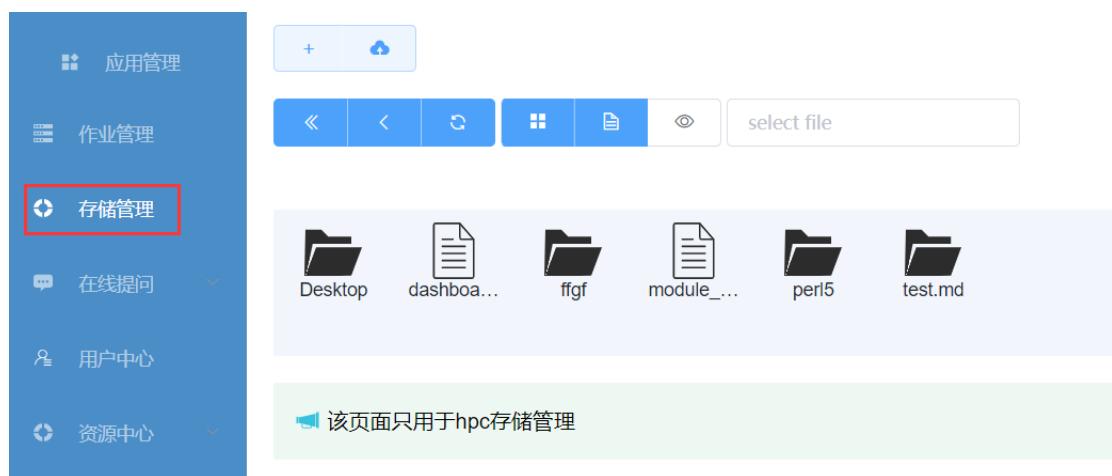


图 25 hpc 存储管理页面

## 六、在线提问

在反馈问题中，可以进行问题的提交。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，对现场截图贴图可以更好地辅助描述问题。

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

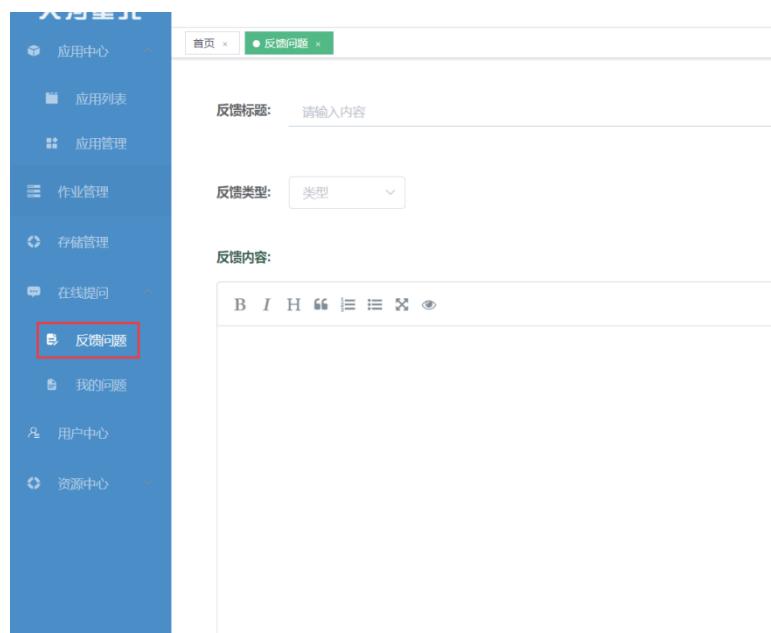


图 26 在线提问编辑页面

## **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、登陆密码的修改。

建议绑定邮箱和手机，以便获得更完整的平台功能。

## 八、 资源中心

### 1. 申请记录

记录用户分区开通、配额变更的申请。用户在状态栏中可以看到审批的进度。点击查看详情，查看详细的审批内容。

| ID | 序号  | 类别     | 申请信息                | 审核信息                | 状态  | 操作                   |
|----|-----|--------|---------------------|---------------------|-----|----------------------|
|    | 592 | 开通计算分区 | 2018-10-29 11:53:32 | 2018-10-29 16:12:48 | 已通过 | <a href="#">查看详情</a> |

图 27 申请记录页面

### 2. 高性能计算

展示用户可以连接的高性能计算集群。用户在具备使用权限的集群上可以看到可用的分区资源，点击“webSSH”将以终端的形式登录集群。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。

温馨提示：  
各集群下的分区资源配置信息仅供参考，具体配置以实际为准，申请开通分区请与市场部联系(service.nscc@nscc-gz.cn)。后续版本将会添加开通分区服务。

| 分区         | CPU核数/内存(GB) | 存储          | 操作系统    | 备注 | 状态  | 操作                   |
|------------|--------------|-------------|---------|----|-----|----------------------|
| work       | 64G / 24核    | HOME / WORK | rhel-65 |    | 未开通 | <a href="#">申请开通</a> |
| free       | 64G / 24核    | HOME / WORK | rhel-65 |    | 已开通 | <a href="#">申请开通</a> |
| MEM_128    | 128G / 24核   | HOME / WORK | rhel-65 |    | 未开通 | <a href="#">申请开通</a> |
| docker_128 | 128G / 24核   | HOME / WORK | docker  |    | 未开通 | <a href="#">申请开通</a> |

图 28 高性能计算页面

### 3. 机时信息

展示用户已用、可用的机时信息。机时状态为“超时”，则用户不能再提交新的作业。请及时联系中心工作人员 [service.nscc@nscc-gz.cn](mailto:service.nscc@nscc-gz.cn)。

| 机时信息       |          |          |          |      |
|------------|----------|----------|----------|------|
| 租户名        | 累计机时(核时) | 已用机时(核时) | 剩余机时(核时) | 机时状态 |
| [REDACTED] | 100000   | 100486   | -486     | 超时   |

图 29 机时信息页面

### 4. 资源调整

展示租户、用户可用集群的资源配置。用户可以根据项目需求，申请调整配额。

\* 租户（用户组）是一组用户的集合。每个用户账号都对应有一个租户，资源用量计费到租户上。比如，一个学校课题组会开一个租户，课题组的所有成员属于该租户。如果需要分别记账，可以再申请一个不同负责人的账号或者作特别说明（第二章第 2 节）。

\* 每个用户的配额、机时等资源，不仅受用户自身的资源限制，也不能超过租户的资源限制。

| 帐户名        |          | 用户类型: 合同账户 |    | 单位名称 |                     |  |  |
|------------|----------|------------|----|------|---------------------|--|--|
| ▲ 租户配额调整   | ▼ 用户配额调整 | 租户hpc资源限制  |    |      |                     |  |  |
| 租户名        |          |            |    |      |                     |  |  |
| [REDACTED] | banhe2d  | 32         | 32 | 64   | <button>申请</button> |  |  |
| [REDACTED] | banhe2g  | 5          | 5  | 5    | <button>申请</button> |  |  |
| [REDACTED] | banhe2h  | 32         | 32 | 64   | <button>申请</button> |  |  |

图 30 资源调整页面

作业提交数：用户最多能提交到作业队列里的作业数目，包含正在运行和排队等待资源的作业。作业提交数  $\geq$  作业运行数。

作业运行数：用户最多能同时运行的作业数目。

结点数：所有运行中的作业占据的结点总数的上限。

CPU、GPU 配额：创建自定义集群时，所有集群 CPU、GPU 数目总和的上限。

内存配额：创建自定义集群时，所有集群内存容量总和的上限。

# 九、 大数据、深度学习应用使用说明

## 1. 应用通用说明

通用说明部分主要为用户说明：用户所创建的大数据、深度学习应用如何进行远程登录、如何连接互联网、如何安装需要的软件、如何进行数据的上传下载等。

### 1.1 大数据、深度学习应用简介

大数据和深度学习的应用，会创建一个用户自定义的专属集群，供创建作业的用户使用。

目前上线的有 Spark、Rstudio 等大数据分析工具、Tensorflow、Caffe、CNTK、Torch、Mxnet、PaddlePaddle 等深度学习框架、Galaxy 生物分析平台以及 Slurm 高性能集群管理软件等，更多的应用也正在不断的发布和完善过程中。

#### 应用环境说明：

| 应用名称         | 版本号     | 配套环境说明   | 其他(备注)                                |
|--------------|---------|--|---------------------------------------|
| Apache Spark | 2.3.2   | scala v2.10.7;<br>Jdk1.8.0_191;<br>anaconda2/5.3.0;anaconda3/5.3.0<br>anaconda3/4.2.0 –python3.5.2 | 支持单节点和集群模式，<br>集群模式中默认 master<br>数为 1 |
| Rstudio      | 1.1.456 | R 3.2.2  |                                       |
| Galaxy       | 18.05   |  |                                       |
| Slurm        | 17.02.9 | openmpi/1.10.7 openmpi/1.7.5   | 集群模式， 默认 master<br>数为 1               |
| TensorFlow   | 1.11.0  | anaconda2/5.3.0;anaconda3/4.2.0  | 默认单机模式；                               |
| Caffe        | 1.0.0   | anaconda2/5.3.0; anaconda3/4.2.0   | 默认单机模式                                |

|              |        |   |        |
|--------------|--------|---|--------|
| Mxnet        | 1.3.0  | anaconda2/5.3.0; anaconda3/4.2.0                  | 默认单机模式 |
| Cntk         | 2.6    | anaconda2/5.3.0; anaconda3/4.2.0                  | 默认单机模式 |
| PaddlePaddle | 0.15.0 | anaconda2/5.3.0; anaconda3/4.2.0                  | 默认单机模式 |
| Torch        | Torch7 | anaconda2/5.3.0; anaconda3/5.3.0; anaconda3/4.2.0 | 默认单机模式 |

## 平台已安装软件包：

当前平台已安装的软件包有如下几类，并且也在不断完善中。用户可使用 `module ava` 命令查看。(若采用的是 Ubuntu 基础镜像，则使用 `source /etc/profile.d/module.sh;` `module ava`)

Anaconda2/5.3.0; Anaconda3/5.3.0 Anaconda3/4.2.0 <https://www.anaconda.com/>

(Anaconda2 对应 python2.7; Anaconda3/4.2.0 对应 python3.5.2; Anaconda3/5.3.0 对应 python3.7)

Cuda/10.0; Cuda/9.0; Cuda/8.0; <https://developer.nvidia.com/cuda-zone>

Cudnn/10.0-v7.3; Cudnn/9.0-v7.1; Cudnn/8.0-v5.1; <https://developer.nvidia.com/cudnn>

Cudnn/8.0-v6.0; Cudnn/8.0-v7.1 <https://developer.nvidia.com/cudnn>

Ant/1.10.5 <https://ant.apache.org/>

Maven/3.6.0 <https://maven.apache.org/>

Apr/1.6.5 <http://apr.apache.org/>

Apr-utils/1.6.1 <https://apr.apache.org/>

Ffmpeg/4.0.2 <https://www.ffmpeg.org/>

Java/1.8.0\_191 <https://www.oracle.com/technetwork/java/javase/downloads/index.html>

Lua/5.3.5 <http://luabinaries.sourceforge.net/>

Openmpi/1.7.5; Openmpi/1.10.7; <https://www.open-mpi.org/>

R/3.2.2 <https://www.r-project.org/>

Scala/2.10.7 <https://www.scala-lang.org/>

Sox/14.4.2 <http://sox.sourceforge.net/>

Spark/2.3.2 <https://spark.apache.org/>

Sqlite/3.25.3 <https://www.sqlite.org/index.html>

Subversion/1.11.0 <https://subversion.apache.org/>

Torch/7 <http://torch.ch/>

## 1.2 ssh 工具远程登录作业

### Step1 打开常用的 xshell、putty 等远程连接工具

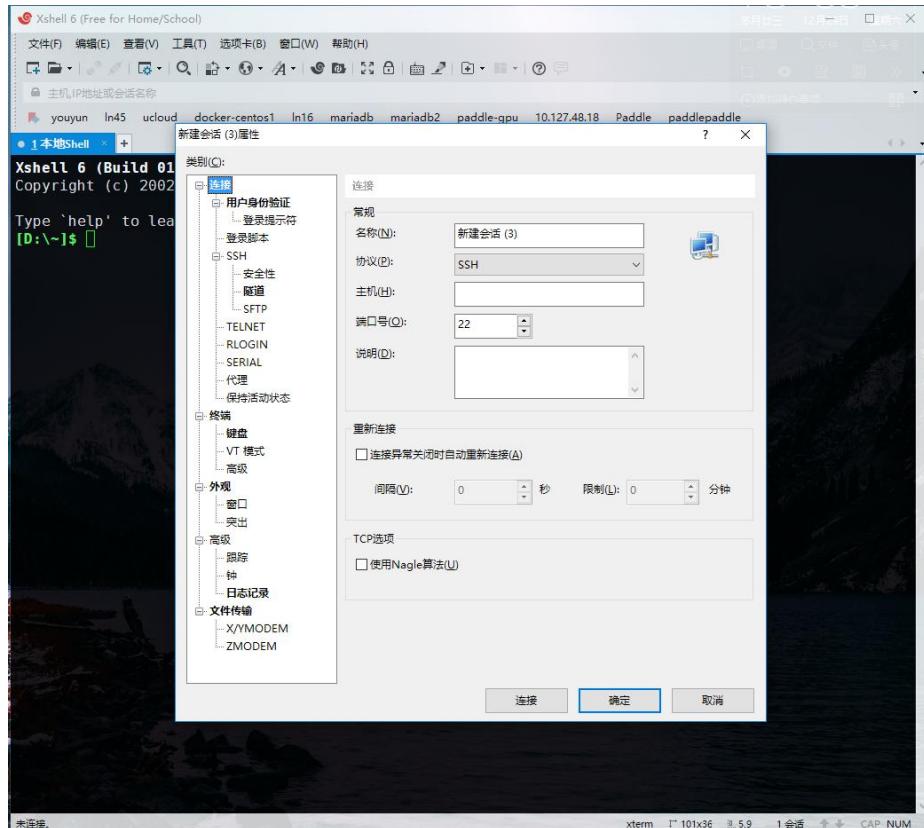


图 31 xshell 连接工具

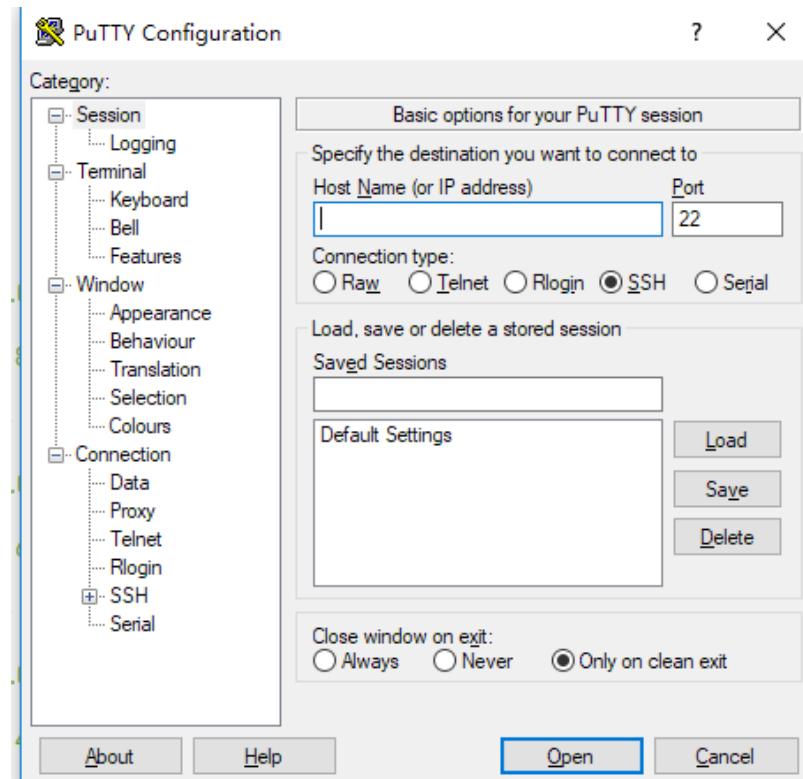


图 32 putty 连接工具

**Step2** 输入 ssh 登陆的域名和端口，域名为 **proxy.nscc-gz.cn**，端口为 **23**，域名和端口信息在“作业详情”页面每个作业的访问入口中记录，如图 33。

图 33 ssh 域名和端口信息

这里以 putty 工具为例，输入域名和端口，点击确定连接：

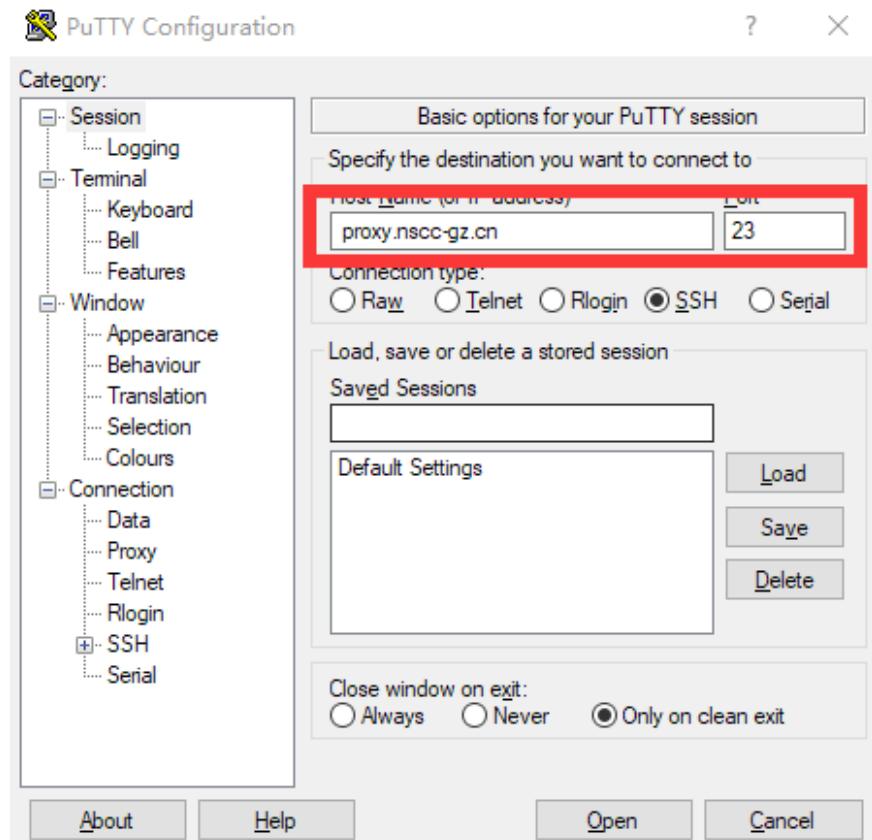


图 34 putty 远程登陆工具

**Step3 输入用户名和密码，用户名为作业详情-基本信息中的提交账号（如图35）。密码为星光平台的登陆密码。**



图 35 作业提交账号

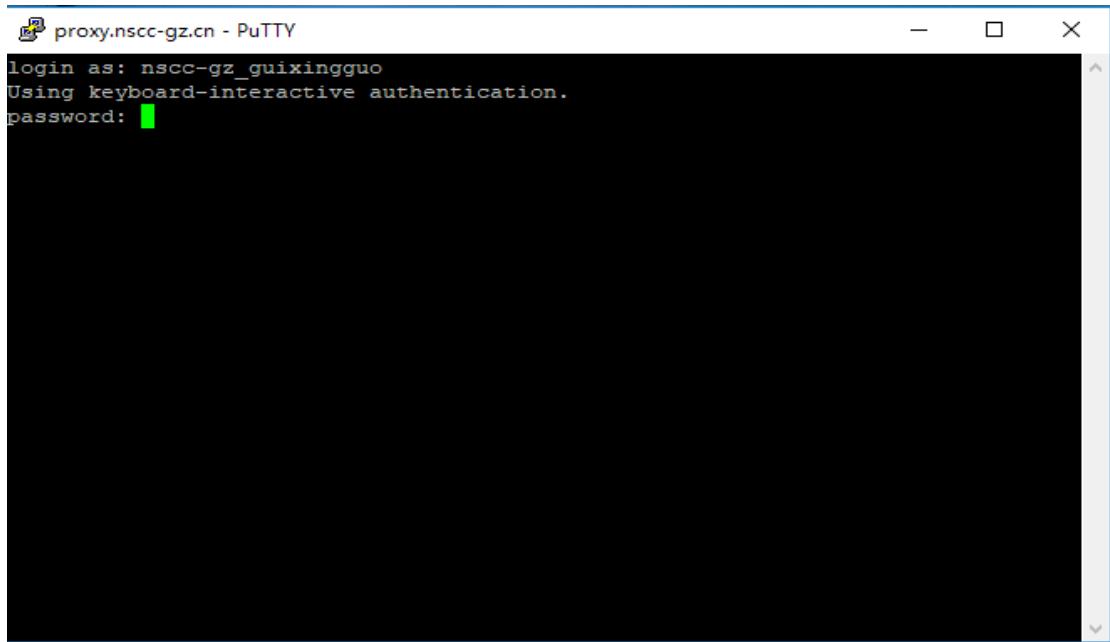


图 36 putty 输入用户名、密码

**Step4** 用户密码验证通过后,选择需要登录的作业名称([这里登录 test111 作业,输入编号 1, 回车后登录到作业](#))

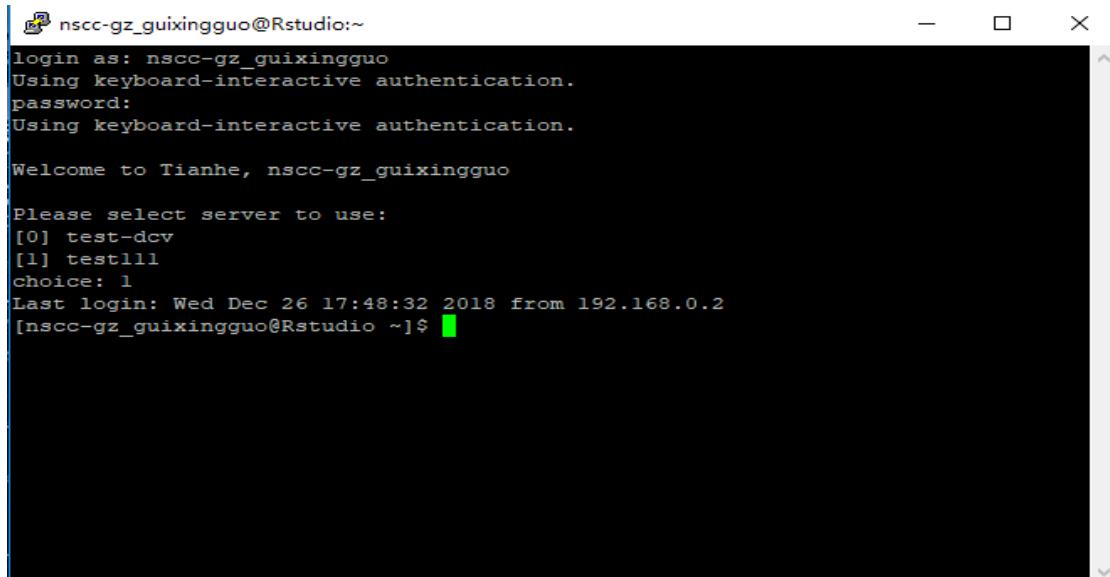


图 37 验证成功, 登录作业

### 1.3 module 加载设置

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。

a) 如果应用是以 centos 操作系统基础镜像创建集群,

执行 **module ava** 列出可用 module

选择要使用的 module 执行 **module load XXX**

例如: 执行 **module ava**

```
[nscc-gz_guixingguo@Rstudio ~]$ module ava
----- /app/centos/modulefiles -----
anaconda2/5.3.0  cuda/8.0      dcmtk/3.6.4    R/3.2.2
anaconda3/4.2.0  cuda/9.0      ffmpeg/4.0.2   scala/2.10.7
anaconda3/5.3.0  cuda/10.0     gradle/4.10.2  sox/14.4.2
ant/1.10.5       cudnn/7.5-v5.1 java/1.8.0_191 spark/2.3.2
apr-util/1.6.1   cudnn/7.5-v6.0 libconfuse/3.2.2 sqlite/3.25.3
apr/1.6.5        cudnn/8.0-v5.1  lua/5.3.5    subversion/1.11.0
cjson/1.7.9      cudnn/8.0-v6.0  maven/3.6.0   torch/7
cmake/3.5.2      cudnn/8.0-v7.1  openmpi/1.7.5  vfstools/1.0.0
cuda/7.0          cudnn/9.0-v7.1  openmpi/1.10.7
cuda/7.5          cudnn/10.0-v7.3 proxy/1.0.0
[nscc-gz_guixingguo@Rstudio ~]$
```

图 38 module ava 打印软件列表

加载需要的 anaconda3/5.3.0 cuda/10.0 ,执行:

**module load anaconda3/5.3.0 cuda/10.0**

```
[nscc-gz_guixingguo@Rstudio ~]$ module load anaconda3/5.3.0 cuda/10.0
[nscc-gz_guixingguo@Rstudio ~]$
```

图 39 module load 加载软件

b) 如果应用是以 Ubuntu 操作系统基础镜像创建集群

执行 **source /etc/profile.d/modules.sh**

**module ava** 列出可用 module

要使用的 module 执行 **module load XXX**

例如: 执行 **module ava**

```
[nscc-gz_guixingga@Rstudio ~]$ module ava  
----- /app/centos/modulefiles -----  
anaconda2/5.3.0  cuda/8.0          dcmtk/3.6.4      R/3.2.2  
anaconda3/4.2.0  cuda/9.0          ffmpeg/4.0.2     scala/2.10.7  
anaconda3/5.3.0  cuda/10.0         gradle/4.10.2    sox/14.4.2  
ant/1.10.5       cudnn/7.5-v5.1    java/1.8.0_191   spark/2.3.2  
apr-util/1.6.1   cudnn/7.5-v6.0    libconfuse/3.2.2  sqlite/3.25.3  
apr/1.6.5        cudnn/8.0-v5.1    lua/5.3.5       subversion/1.11.0  
cjson/1.7.9      cudnn/8.0-v6.0    maven/3.6.0     torch/7  
cmake/3.5.2      cudnn/8.0-v7.1    openmpi/1.7.5   vfstools/1.0.0  
cuda/7.0         cudnn/9.0-v7.1    openmpi/1.10.7  
cuda/7.5         cudnn/10.0-v7.3   proxy/1.0.0  
[nscc-gz_guixingga@Rstudio ~]$
```

图 40 module ava 打印软件列表

**module load anaconda3/5.3.0 cuda/10.0**

```
[nscc-gz_guixingga@Rstudio ~]$ module load anaconda3/5.3.0 cuda/10.0  
[nscc-gz_guixingga@Rstudio ~]$
```

图 41 module load 加载软件

## 1.4 网络代理设置

登录应用创建的自有集群之后，如果需从集群中访问互联网，需设置访问代理。

代理设置方式为：

**module load proxy/1.0.0**

取消网络代理为：

**module unload proxy/1.0.0**

用户仅在需要设置代理时进行上述代理设置操作，并且**在需要使用代理步骤完成后，建议立即取消代理**，否则会造成 tensorflow 等应用无法正常运行。

## 1.5 数据上传下载

数据上传和下载可使用 SSH 工具进行上传下载，推荐使用 xftp 工具。若数据量巨大，则可使用硬盘挂载的方式进行数据上传下载[需联系技术支持人员进行硬盘挂载]。下面是使用 Xftp 工具进行数据上传和下载示例。

**Step1** 进入作业详情页，获取远程应用入口，即域名 `proxy.nscc-gz.cn`，端口 23。

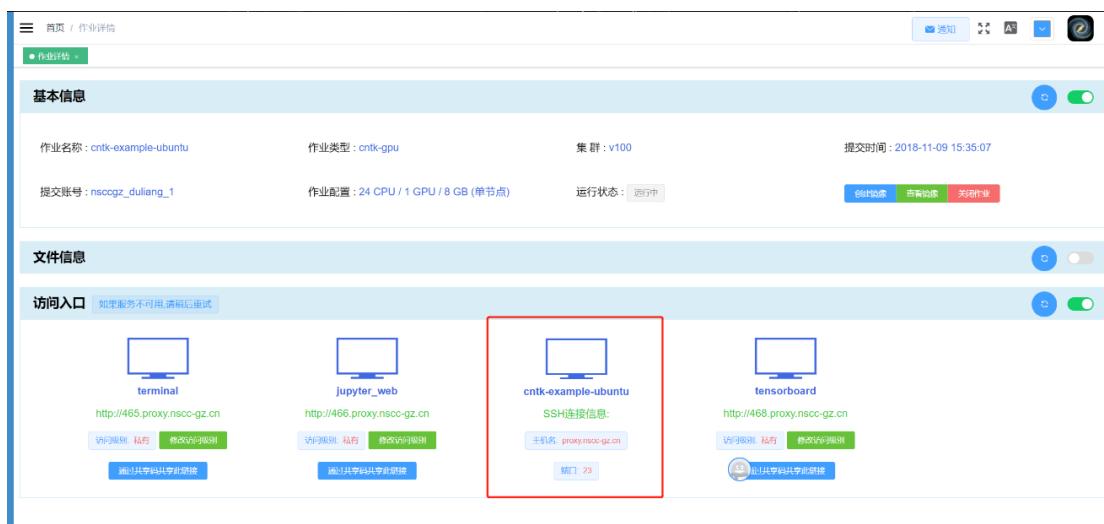


图 42 作业详情-访问入口

**Step2** 采用 xftp 工具进行远程登录

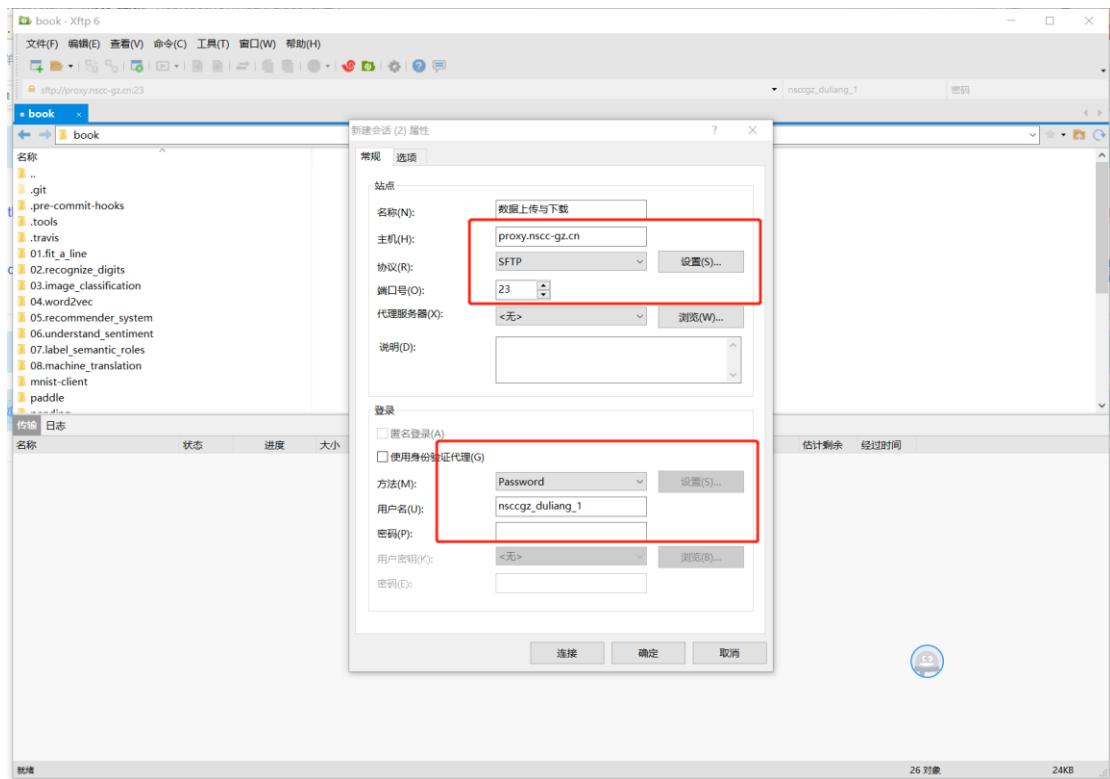


图 43 xftp 远程连接工具

**Step3 输入账户密码，用户名为作业详情-基本信息中的提交账号（如图 38）。密码为星光平台的登陆密码。**



图 44 提交账号信息

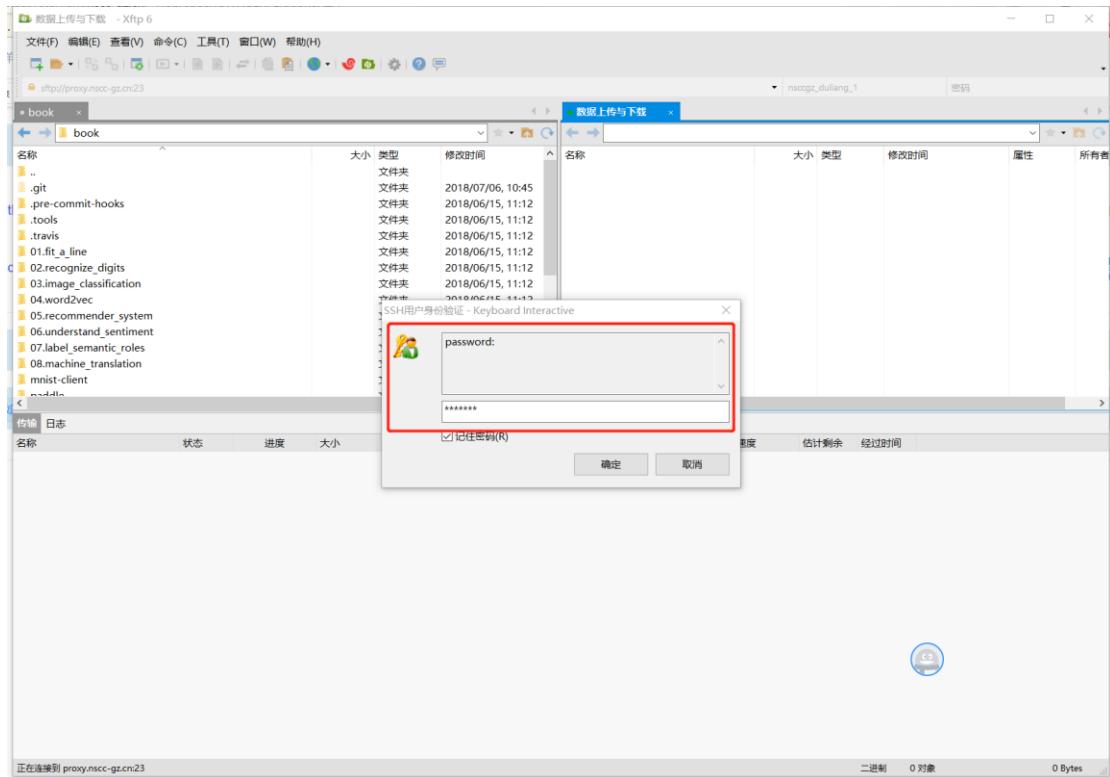


图 45 xftp 密码输入

**Step4 用户密码验证通过后，选择需要登录的作业名称(这里登录 centos-spark 作业，输入编号 3，回车后登录到作业)**

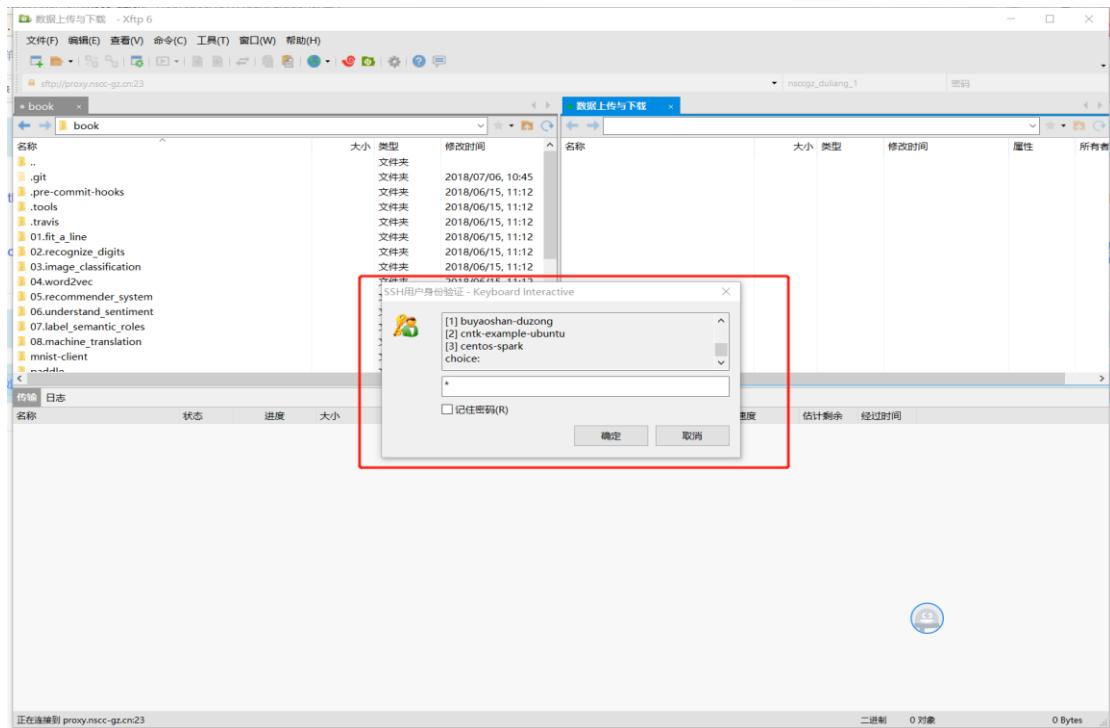


图 46 xftp 目标应用选择

## Step5 选择上传或者下载文件进行文件上传和下载

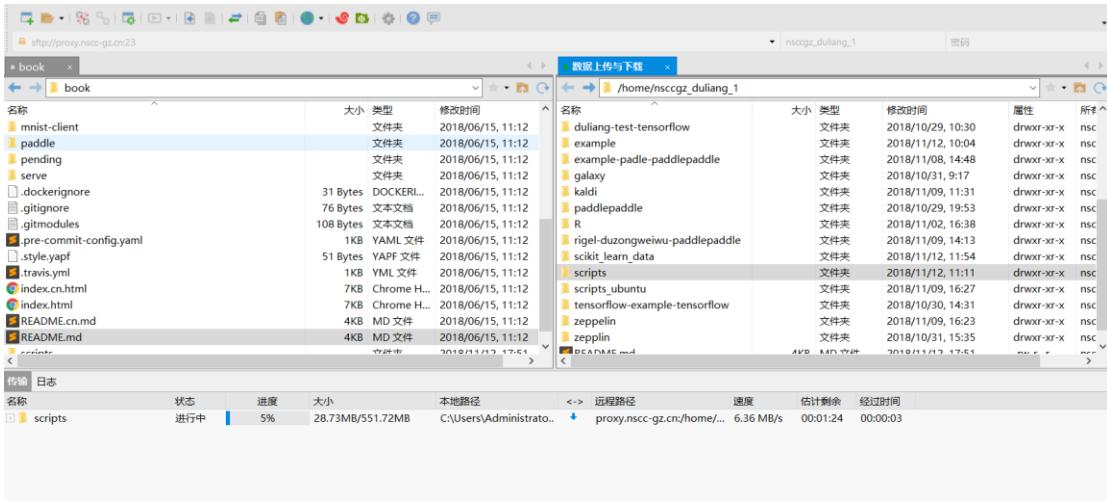


图 47 sftp 登陆成功-文件上传下载

## Step6 文件上传与下载示意图

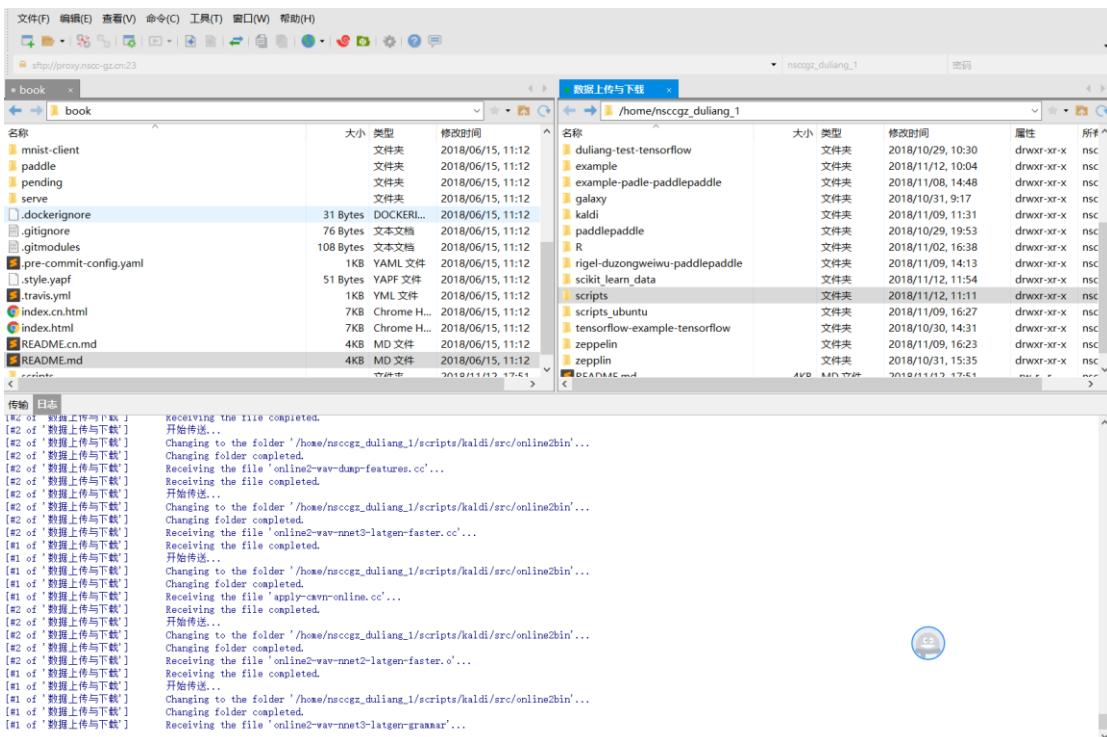


图 48 上传下载示例

## 1.6 自定义安装用户 python 包

用户在本自有集群中若直接使用 pip 命令进行 python 包的安装，会出现权限不足(**permission denied**)错误。因此要在本平台进行用户自定义 python 包安装时可利用 pip 命令制定安装参数进行 python 包的安装。Pip 使用手册链接为: [https://pip.pypa.io/en/stable/user\\_guide/](https://pip.pypa.io/en/stable/user_guide/)

通过命令

**pip install -- user xxx**

可以将 xxx 软件包安装于

**~/.local/lib/pythonversion/site-packages/xxx**

示例如下：

```
[nsccgz_duliang_1@PaddlePaddle ~]$ pip install --user pp
Collecting pp
  Downloading https://files.pythonhosted.org/packages/14/e9/f69030681985226849be0cd36b04e2c0cb99babff23c8342bc4e30ded06b2/pp-1.6.5.tar.gz
Building wheels for collected packages: pp
  Running setup.py bdist_wheel for pp ... done
    Stored in directory: /home/nsccgz_duliang_1/.cache/pip/wheels/c8/a4/b1/34348a50f905b7031fb3621b302d82a269abb9446d20f9cf59
Successfully built pp
Installing collected packages: pp
Successfully installed pp-1.6.5
[nsccgz_duliang_1@PaddlePaddle ~]$ cd /home/nsccgz_duliang_1

[nsccgz_duliang_1@PaddlePaddle a4]$ python
Python 2.7.15 |Anaconda, Inc.| (default, May 1 2018, 23:32:55)
[GCC 7.2.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import pp
>>> exit
Use exit() or Ctrl-D (i.e. EOF) to exit
>>> exit()
[nsccgz_duliang_1@PaddlePaddle a4]$ cd ~/.local/
[nsccgz_duliang_1@PaddlePaddle .local]$ ls
bin lib share
[nsccgz_duliang_1@PaddlePaddle .local]$ cd lib/
[nsccgz_duliang_1@PaddlePaddle lib]$ ls
python2.7
[nsccgz_duliang_1@PaddlePaddle lib]$ cd python2.7/
[nsccgz_duliang_1@PaddlePaddle python2.7]$ ls
site-packages
[nsccgz_duliang_1@PaddlePaddle python2.7]$ cd site-packages/
[nsccgz_duliang_1@PaddlePaddle site-packages]$ ls
pp-1.6.5.dist-info pauto.py ppauto.pyc ppcommon.py ppcommon.pyc pp.py pp.pyc pptransport.py pptransport.pyc ppworker.py ppworker.pyc
[nsccgz_duliang_1@PaddlePaddle site-packages]$ pwd
/home/nsccgz_duliang_1/.local/lib/python2.7/site-packages
[nsccgz_duliang_1@PaddlePaddle site-packages]$
```

图 49 python 包安装示例

## 1.7 自定义源码编译安装软件

用户在本自有集群中若需编译安装其他第三方软件，若直接安装时出现权限不足情况时，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，而不需要 root 权限。

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。

## 2. Apache Spark 使用说明

Apache Spark 是专为大规模数据处理而设计的快速通用的计算引擎。

Spark 是 UC Berkeley AMP lab (加州大学伯克利分校的 AMP 实验室)所开源的类 Hadoop MapReduce 的通用并行框架，拥有 Hadoop MapReduce 所具有的优点；下面以 spark 中经典的 wordcount 程序为例，演示创建和使用 Spark 程序的步骤和相关流程。

Spark 应用的创建与使用示例视频(zepplin 启动较为耗时，初次使用时需等待 10 分钟左右)：

### Step1 选择创建参数，点击创建 Spark 应用



图 50 红框内为 spark 视频、文档教程

## Step2 创建成功后，跳转至作业详情页

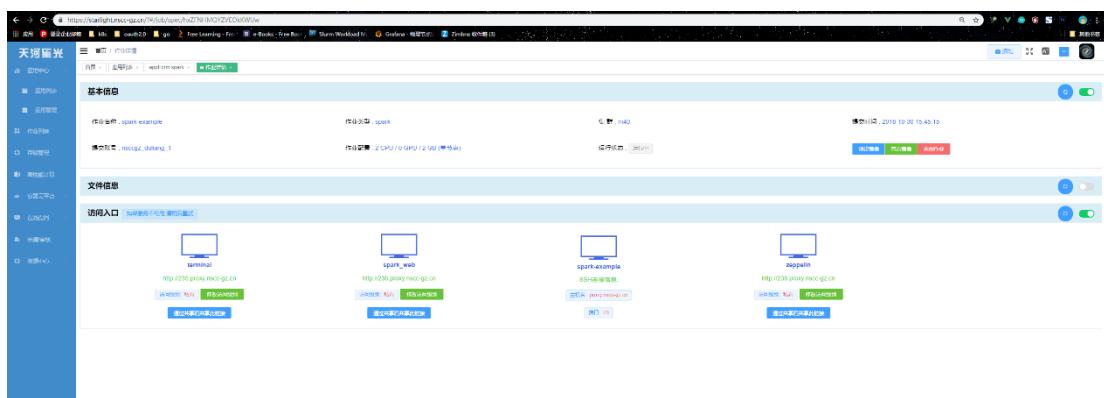


图 51 作业详情页面

在访问入口处，依据自身需求点击相应链接进入到应用内部

## Step3 Spark-Terminal 示例

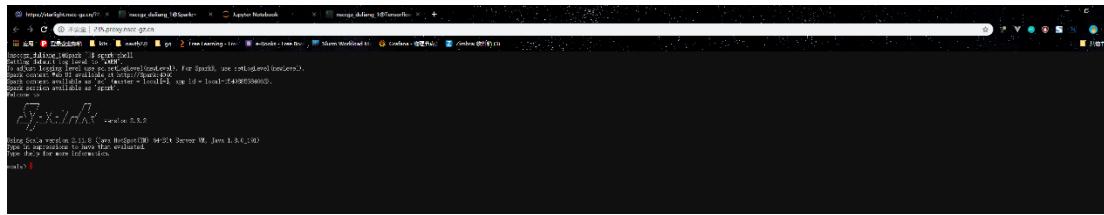


图 52 Spark-Terminal

## Step4 Spark-WebUI 示例

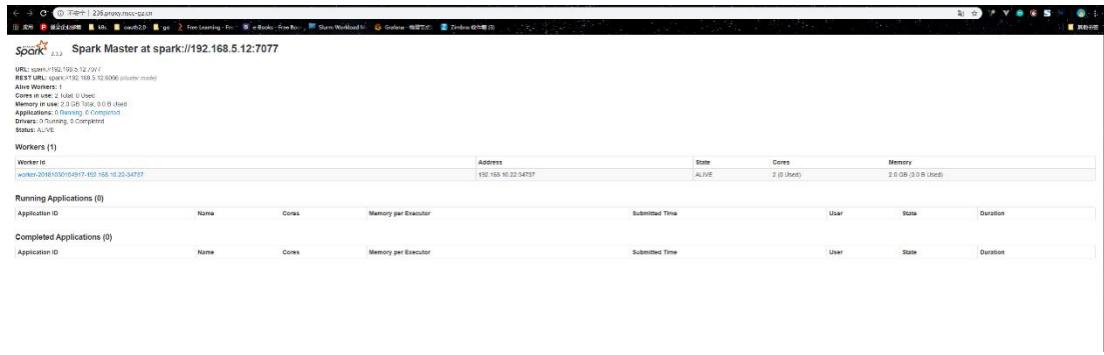


图 53 Spark-WebUI

## Step5 Zeppelin 使用示例(账号密码为： admin/admin)

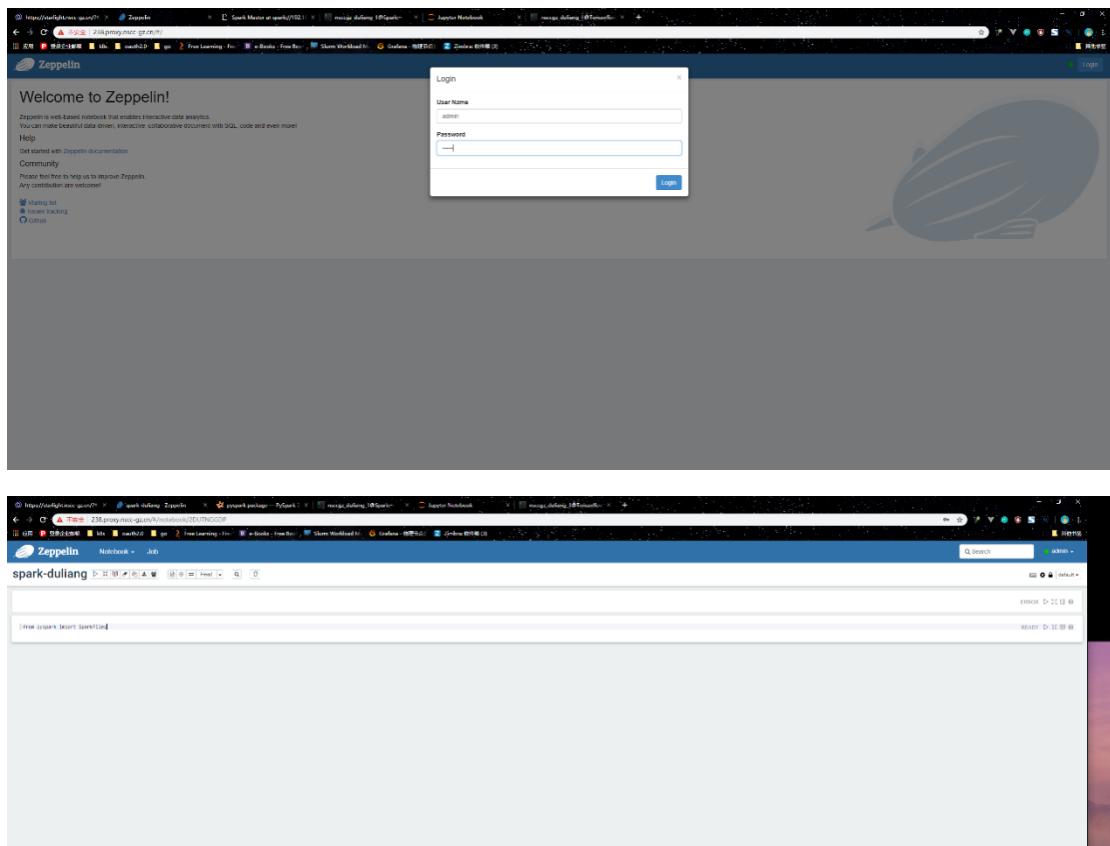
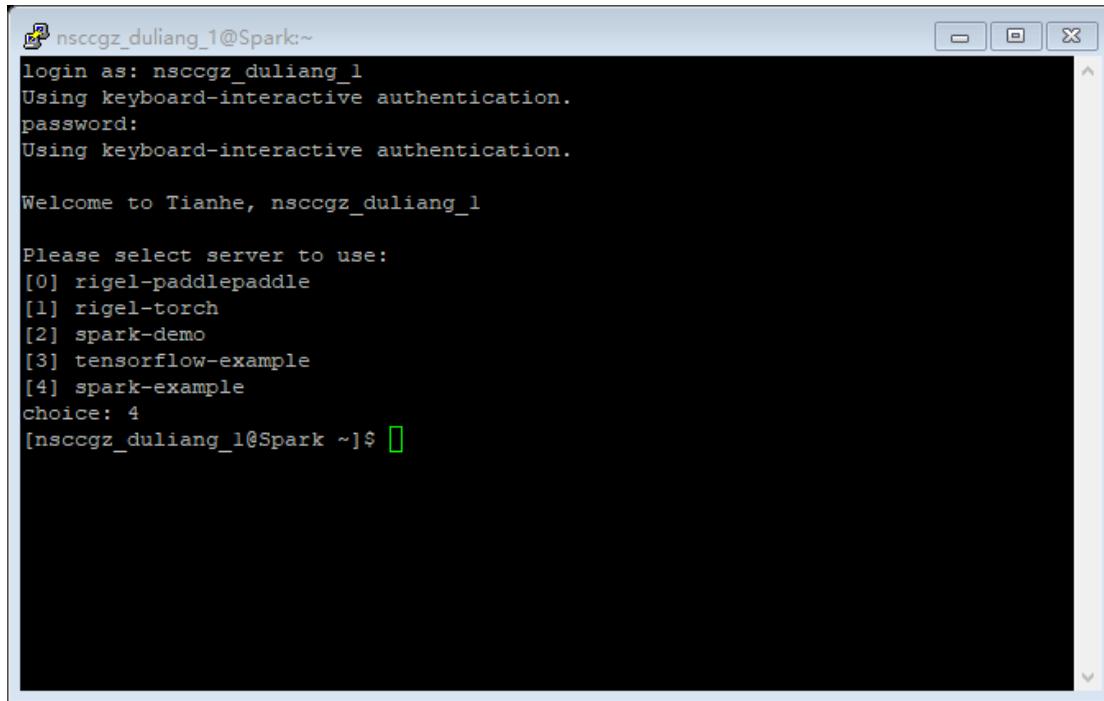


图 54 Zeppelin

**Step6 采用 SSH 工具登陆应用(SSH 账户密码为用户的提交账号和密码)。参考  
第九章 1.2 节 ssh 工具远程登录作业。**



```
nsccgz_duliang_1@Spark:~  
login as: nsccgz_duliang_1  
Using keyboard-interactive authentication.  
password:  
Using keyboard-interactive authentication.  
  
Welcome to Tianhe, nsccgz_duliang_1  
  
Please select server to use:  
[0] rigel-paddlepaddle  
[1] rigel-torch  
[2] spark-demo  
[3] tensorflow-example  
[4] spark-example  
choice: 4  
[nsccgz_duliang_1@Spark ~]$
```

图 55 远程连接工具

**Step7 运行 Spark 示例程序**

示例程序位于/app/common/example/spark 目录下, 程序脚本为 run\_example.sh

执行 cp -r /app/common/example/spark /home/your\_home\_dir/targetdir/

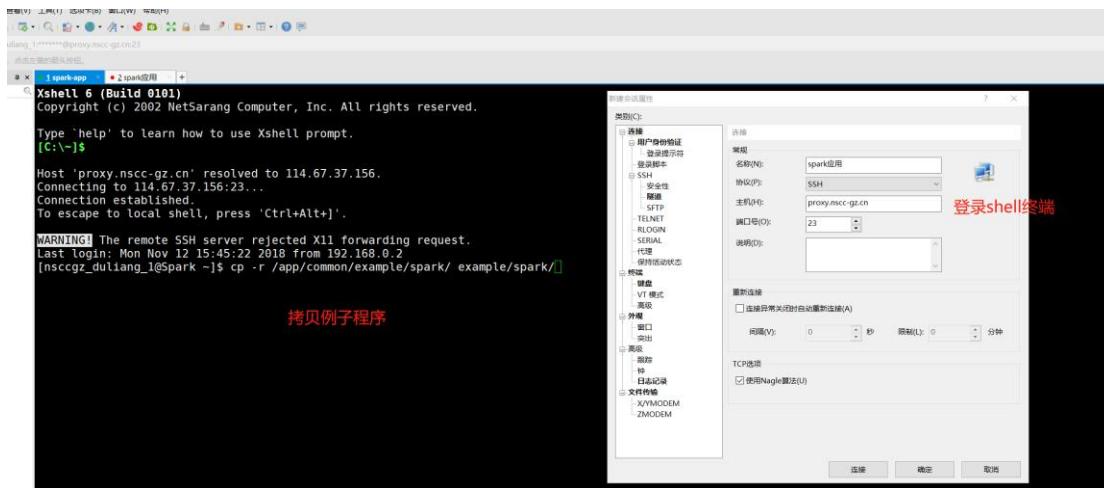
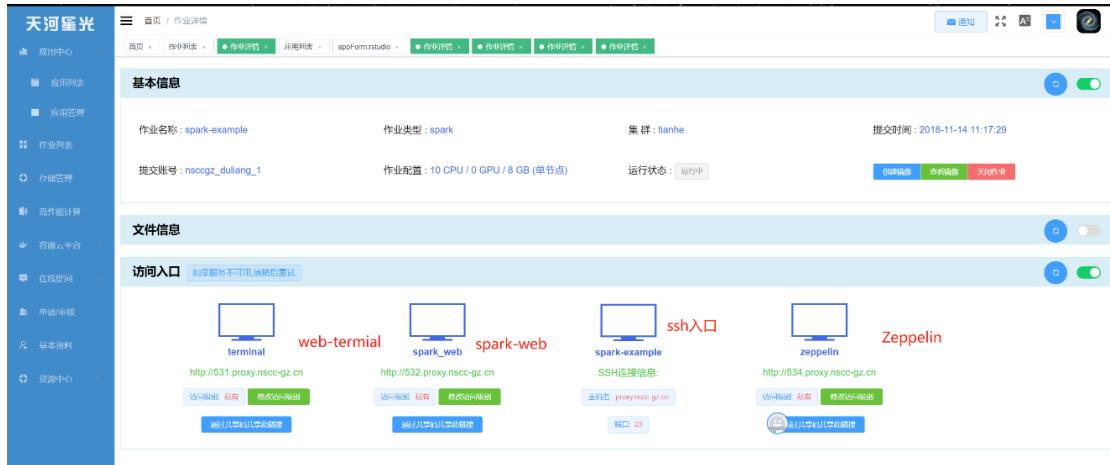


图 56 ssh 工具连接执行例程



```

https://starlight.nscc-gz.cn/?t= nsccgz_duiling_1@PaddlePad TensorBoard Home Untitled1 nsccgz_duiling_1@PaddlePad
<-- C A 不安全 461.proxy.nscc-gz.cn
应用 登录企业邮箱 Kbs oauth2.0 go Free Learning - Fre B e-Books - Free Books Slurm Workload M Grafana - 物理节点 Zimbra - 收件箱 (3)

for example Java-SparkPi,Java-WordCount,Scala-SparkPi,Python-WordCount,Python-SparkPi
enter example name*: [Java-WordCount]
exampleName can be 'Java-SparkPi', 'Java-WordCount', 'Scala-SparkPi', 'Python-WordCount', 'Python-SparkPi'
(nsccgz_duiling_1@PaddlePad) run.example.sh
--run spark python scala examples...
you can choose Java-SparkPi, Java-WordCount, Scala-SparkPi, Python-WordCount, Python-SparkPi
enter example name:Java-SparkPi
run java pi examples[calculate pi value by spark]
spark-submit --class nscc.JavaSparkPi --master local[*] java/SparkPi.jar
java HOME is not set
(nsccgz_duiling_1@PaddlePad) spark$ module ava 列出module
awsconsolid/5.3.0 ant/1.10.5 apache-maven/3.6.3 cuda/8.0 cuda/10.0 cudnn/7.5-cudnn/8.0 cudnn/8.0-v6.0 cudnn/9.0-v7.1 ffmpeg/4.0.2 lua/5.3.5 openmpi/1.10.7 scalr/2.3.2 subversion/1.11.0
awsconsolid/5.3.0 ant/1.10.5 apache-maven/3.6.3 cuda/8.0 cuda/10.0 cudnn/7.5-cudnn/8.0 cudnn/8.0-v5.2 cudnn/9.0 cudnn/9.0-v7.1 cudnn/10.0-v7.3 java/1.8.0_191 maven/3.6.0 openmpi/1.10.7 scalr/2.3.2 subversion/1.11.0
(nsccgz_duiling_1@PaddlePad) spark$ module load nscc.example.sh
--run spark python scala examples...
you can choose example to run
for example Java-SparkPi,Java-WordCount,Scala-SparkPi,Python-SparkPi,Python-WordCount
run java pi examples[calculate pi value by spark]
spark-submit --class nscc.JavaSparkPi --master local[*] java/SparkPi.jar
2018-11-14 10:50:05 WARN NativeCodeLoader:62 - Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
2018-11-14 10:50:05 INFO SparkContext:23 - Registering Spark version 2.3.1
2018-11-14 10:50:05 INFO SecurityManager:54 - Changing view acls to: nsccgz_duiling_1
2018-11-14 10:50:05 INFO SecurityManager:54 - Changing modify acls to: nsccgz_duiling_1
2018-11-14 10:50:05 INFO SecurityManager:54 - Changing view acls groups to:
2018-11-14 10:50:05 INFO SecurityManager:54 - Changing modify acls groups to:
2018-11-14 10:50:05 INFO Utils:54 - Successfully started service 'sparkDriver' on port 3629.
2018-11-14 10:50:05 INFO Utils:54 - Successfully started service 'sparkUI' on port 4040.
2018-11-14 10:50:05 INFO SparkEnv:54 - Registering BlockManagerMaster
2018-11-14 10:50:05 INFO BlockManagerMasterEndpoint:54 - Using org.apache.spark.storage.DefaultTopologyMapper for getting topology information
2018-11-14 10:50:05 INFO BlockManagerMasterEndpoint:54 - Registering master endpoint at /tmp/blockmgr-22d519ac-8600-46fb-ac20-f93695bfed01
2018-11-14 10:50:05 INFO MemoryStore:54 - MemoryStore started with capacity 366.3 MB
2018-11-14 10:50:05 INFO SparkEnv:54 - Registering OutputCommitCoordinator
2018-11-14 10:50:05 INFO Utils:54 - Logging initialized @80ms.
2018-11-14 10:50:05 INFO Utils:54 - Current hostname: 3.3.2-0~rc1+9504, build timestamp: unknown, git hash: unknown
2018-11-14 10:50:05 INFO Server:419 - Started 4606ms
2018-11-14 10:50:05 INFO AbstractConnector:278 - Started ServerConnector@61060803[HTTP/1.1][{http://1.1.1.1:4040}][{0.0.0.0:4040}]
2018-11-14 10:50:05 INFO Utils:54 - Successfully started service 'sparkUI' port 4040.
2018-11-14 10:50:05 INFO ContextHandler:781 - Started o.s.j.s.ServletContextHandler@e722f/{/jobs,null,AVAILABLE}@spark
2018-11-14 10:50:05 INFO ContextHandler:781 - Started o.s.j.s.ServletContextHandler@4789a/{/jobs/json,null,AVAILABLE}@spark
2018-11-14 10:50:05 INFO ContextHandler:781 - Started o.s.j.s.ServletContextHandler@b8665/{/jobs/job/json,null,AVAILABLE}@spark
2018-11-14 10:50:05 INFO ContextHandler:781 - Started o.s.j.s.ServletContextHandler@2eabb0/{/tags/null,AVAILABLE}@spark
2018-11-14 10:50:05 INFO ContextHandler:781 - Started o.s.j.s.ServletContextHandler@5282b/{/tags/json/null,AVAILABLE}@spark
2018-11-14 10:50:05 INFO ContextHandler:781 - Started o.s.j.s.ServletContextHandler@7a653a/{/tags/tag/json/null,AVAILABLE}@spark
2018-11-14 10:50:05 INFO ContextHandler:781 - Started o.s.j.s.ServletContextHandler@6af524/{/stages/stage/json/null,AVAILABLE}@spark
2018-11-14 10:50:05 INFO ContextHandler:781 - Started o.s.j.s.ServletContextHandler@c62ab3/{/stages/stage/json/null,AVAILABLE}@spark
2018-11-14 10:50:05 INFO ContextHandler:781 - Started o.s.j.s.ServletContextHandler@c6cfef/{/tags/pool/json,null,AVAILABLE}@spark
2018-11-14 10:50:05 INFO ContextHandler:781 - Started o.s.j.s.ServletContextHandler@9539e3a/{/tag/json/null,AVAILABLE}@spark
2018-11-14 10:50:05 INFO ContextHandler:781 - Started o.s.j.s.ServletContextHandler@957a13a/{/storage/json/null,AVAILABLE}@spark
2018-11-14 10:50:05 INFO ContextHandler:781 - Started o.s.j.s.ServletContextHandler@95c72ca/{/storage/json/json,null,AVAILABLE}@spark
2018-11-14 10:50:05 INFO ContextHandler:781 - Started o.s.j.s.ServletContextHandler@95d9111/{/storage/json/json/json,null,AVAILABLE}@spark

```

图 57 spark web-terminal 连接执行例程

若选择 Ubuntu 镜像，则需先执行 `source /etc/profile.d/module.sh` 然后执行 `module load java`

**module load java**

## 在 spark-web 中查看结果

The screenshot shows the Spark Master web interface at <https://Starlight.nccc-gz.cn/>. The main page displays cluster statistics: URL: spark://192.168.18.2:7077, Web UI: spark://192.168.18.2:8086 (cluster mode), Active Workers: 20, Cores In use: 20 Total, 0 Used, Memory In use: 16.0 GB Total, 0.0 B Used, Applications: 0 Running, 0 Completed, Drivers: 0 Running, 0 Completed, Status: ALIVE. Below this, there are sections for Workers (2) and Applications (0). The Applications section is currently empty.

图 58 spark web

## 3. Rstuiio 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。

类似的，使用 Rstuiio 应用分为填写应用参数，创建 Rstuiio 应用和通过应用访问入口访问应用等步骤。

### Step1 创建 Rstuiio 应用

The screenshot shows the RStudio application creation form. It includes fields for Job Type (作业配置类型), Job Name (作业名称), Cluster Selection (集群选择), Job Configuration (作业配置), Image Selection (镜像选择), Application Programs (应用教程), and a Note (提示) section. The job type is set to CPU(核)/内存(GB). The job name is "唯一不可重复". The cluster selection is "天河CPU集群". The job configuration options are 6核/6GB, 12核/15GB, 24核/30GB, and 48核/60GB. The image selection is "RStudio Centos". The application programs listed are "rstudio视频教程" and "rstudio文档教程". A note at the bottom states: "kubernetes应用不使用存储管理页面上传数据".

图 59 Rstuiio 应用表单

## Step2 应用创建成功

The screenshot shows the 'Application Details' page of the Tianhe Starlight system. On the left is a sidebar with various navigation options: Application Center, Application Management, Job List, Storage Management, Heterogeneous Computing, Cloud Platform, Online Monitoring, Application Submission, Basic Configuration, and Resource Center. The main content area has tabs for 'Basic Information' and 'File Information'. Under 'Basic Information', it displays the job name as 'rstudio-example', job type as 'rstudio', cluster as 'tianhe', submission time as '2018-11-13 10:29:34', and configuration as '10 CPU / 0 GPU / 12 GB (Single Node)'. It also shows the running status as 'Running'. Below this, there's a section for 'File Information' and 'Access入口' (Access入口). Under 'Access入口', there are two entries: 'rstudio\_web' with URL 'http://499.proxy.nscc-gz.cn' and 'SSH连接信息' (SSH Connection Information) with host 'proxy.nscc.gz.cn' and port '22'.

图 60 作业详情

## Step3 Web 方式访问 Rstudio, 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

The screenshot shows a web browser window with the URL '499.proxy.nscc-gz.cn/auth-sign-in'. The page title is 'Sign in to RStudio'. It contains fields for 'Username' (set to 'nsccg2\_dullang\_1') and 'Password', a 'Stay signed in' checkbox, and a 'Sign In' button.

图 61 Rstudio 登陆

## Step4 进入 Rstudio 应用

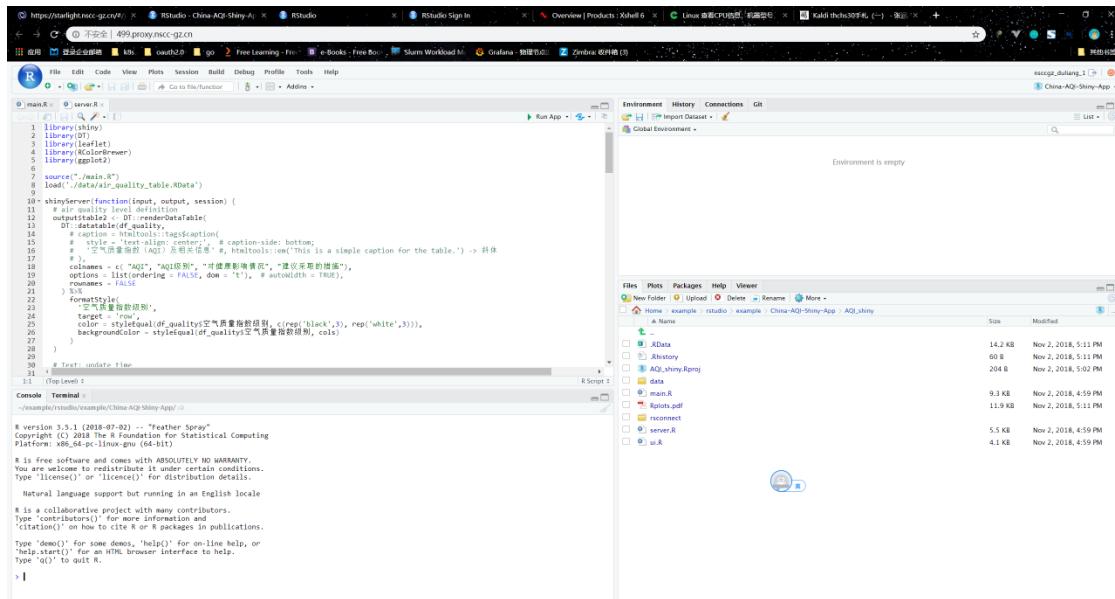
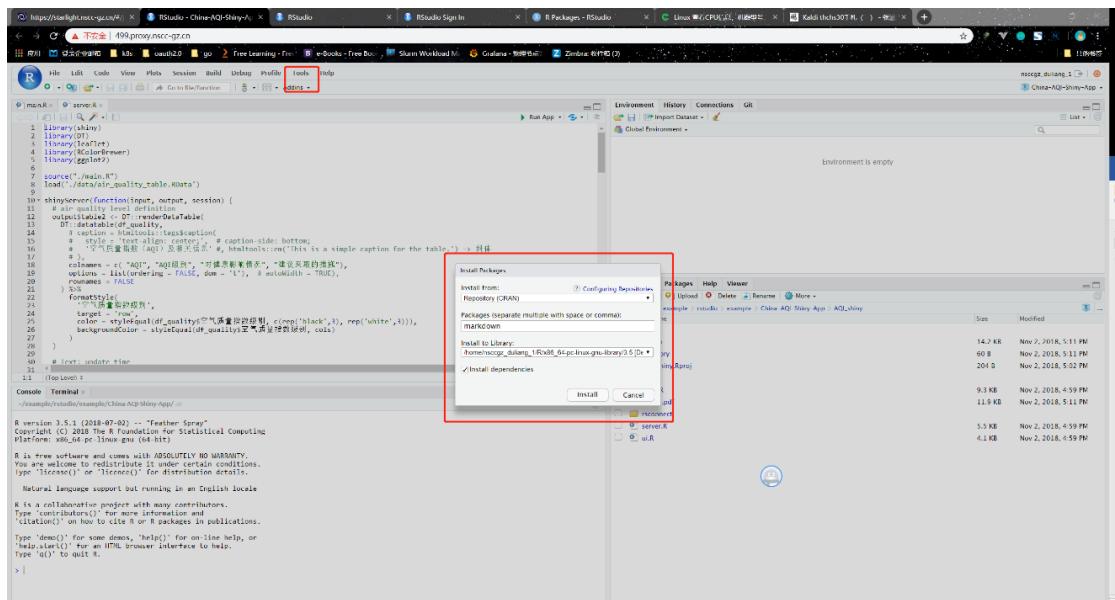


图 62 Rstudio 应用

## Step5 在线安装 R package



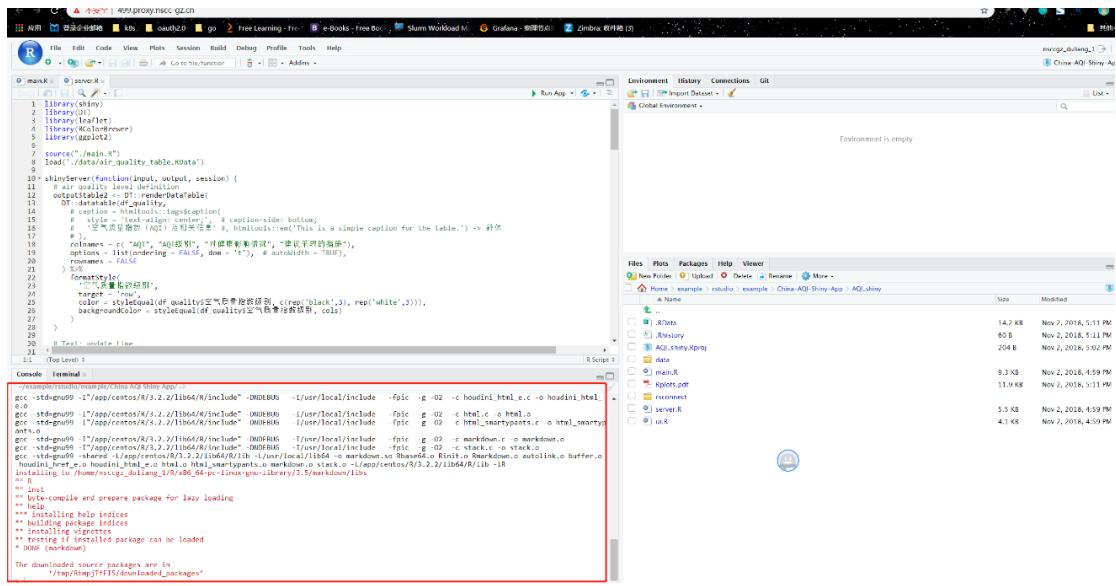


图 63 R package 安装

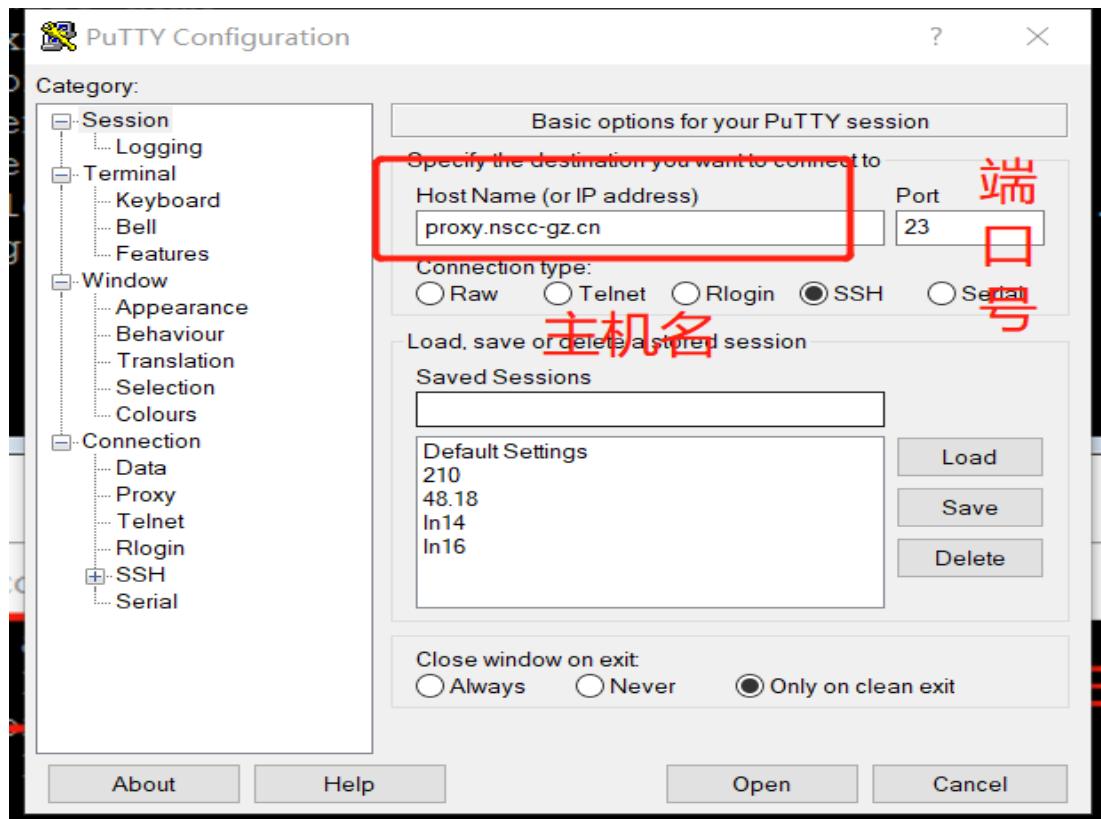
## Step6 运行 Rstuido 示例程序

示例程序位于 `/app/common/example/rstudio` 目录下

需要首先执行：

```
cp -r /app/common/example/rstudio /home/your_home_dir/targetdir/
```

远程登录参考第九章 1.2 节 ssh 工具远程登录作业



The screenshot shows a terminal window titled 'nsccgz\_duliang\_1@Rstudio:~'. The text in the window is as follows:

```
login as: nsccgz_duliang_1
Using keyboard-interactive authentication.
password: 用户登录
Using keyboard-interactive authentication.

Welcome to Tianhe, nsccgz_duliang_1

Please select server to use:
[0] buyaoshan-duzong
[1] cntk-example-ubuntu
[2] centos-spark
[3] rstudio-example
[4] galaxy-example
[5] caffe-demo
[6] mxnet-example
[7] torch-example
[8] tensorflow-example 选择应用
choice: 3
Last login: Wed Nov 14 09:38:31 2018 from 192.168.1.2
[nscggz_duliang_1@Rstudio ~]$ cp -r /app/common/example/rstudio/ rstudio-example/ 拷贝例子程序
```

图 64 ssh 远程登录执行例程

## Jupyter web 编写代码运行：

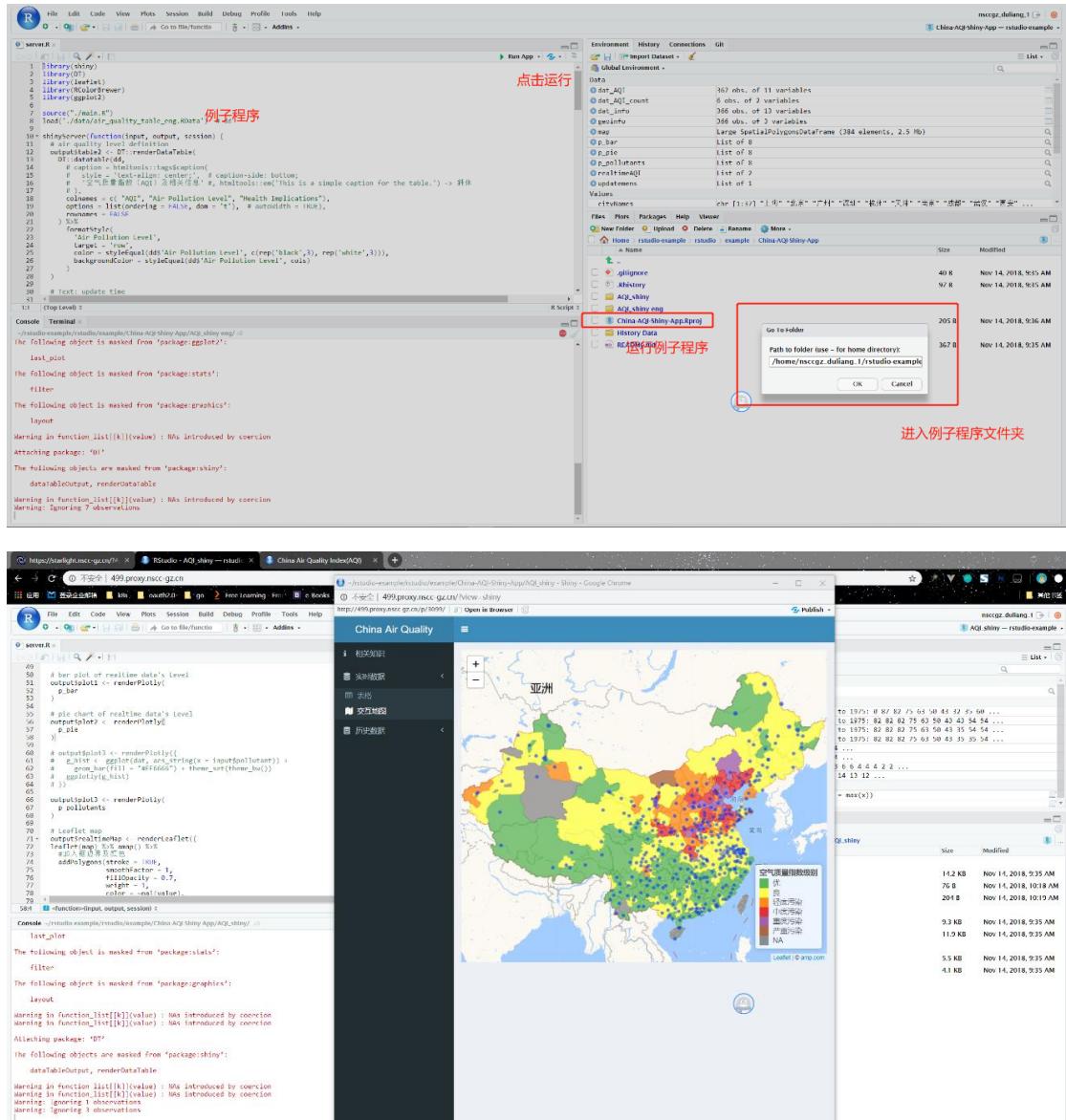


图 65 jupyter-web 编写代码并执行

以上演示了使用 RStudio 的使用流程和相关必要步骤，关于 RStudio 的相关知识请移步至 [rstudio 官网](#)

## 4. 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，演示创建创建和使用 galaxy 的相关步骤和流程。

## Step1 创建 Galaxy 应用

The screenshot shows the Galaxy application configuration interface. It includes fields for job name, cluster selection, job configuration, image selection, and a summary section.

作业名称: 唯一,不可重复

集群选择: 天河 CPU 集群

作业配置: 6核/6GB, 12核/15GB, 24核/30GB, 48核/60GB

镜像选择: Galaxy CentOS

费用: 6 核时/小时

应用教程:

- 暂无galaxy视频教程
- galaxy文档教程

提示: kubernetes应用不使用存储管理页面上传数据

图 66 galaxy 应用表单

## Step2 web 方式访问 galaxy 应用(首次使用用户需注册)

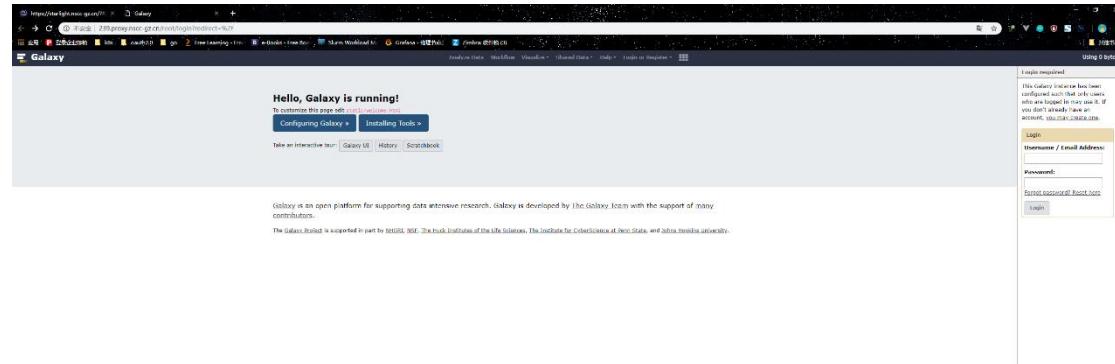


图 67 galaxy web 访问

## Step3 注册 galaxy 用户(email 为:admin@nscc-gz.cn)

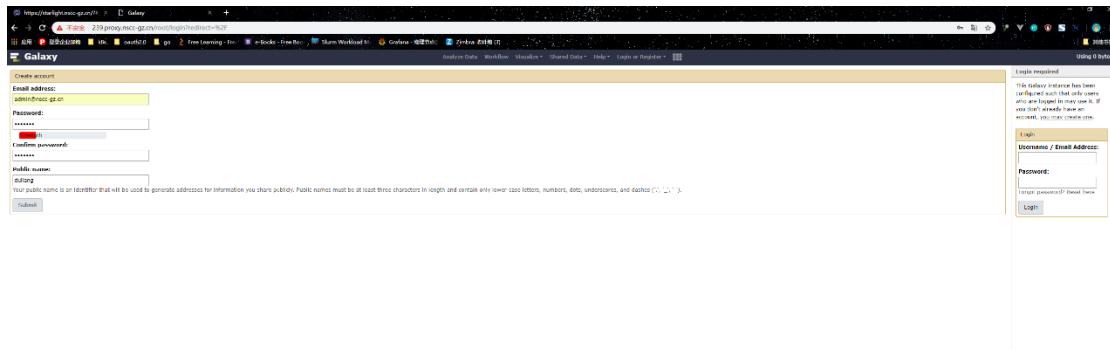


图 68 galaxy 用户注册

#### Step4 注册 galaxy 用户成功后，登陆 Galaxy

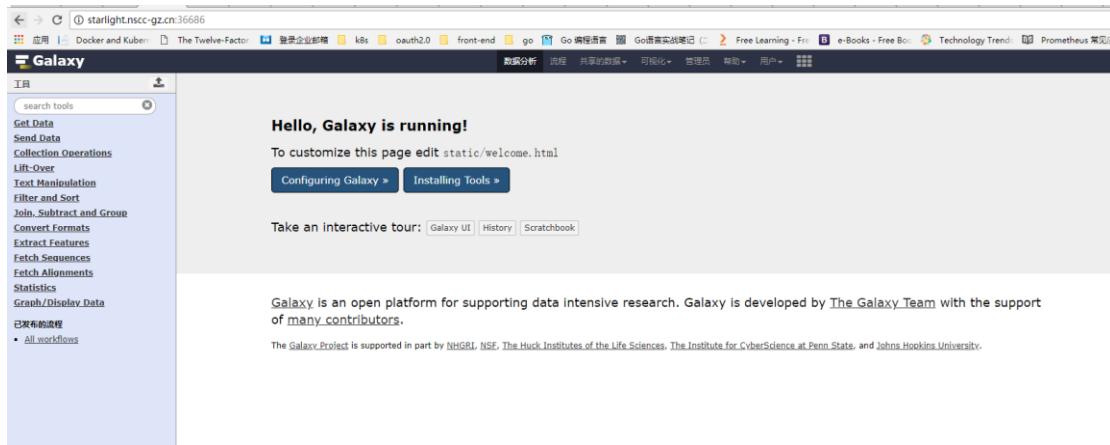
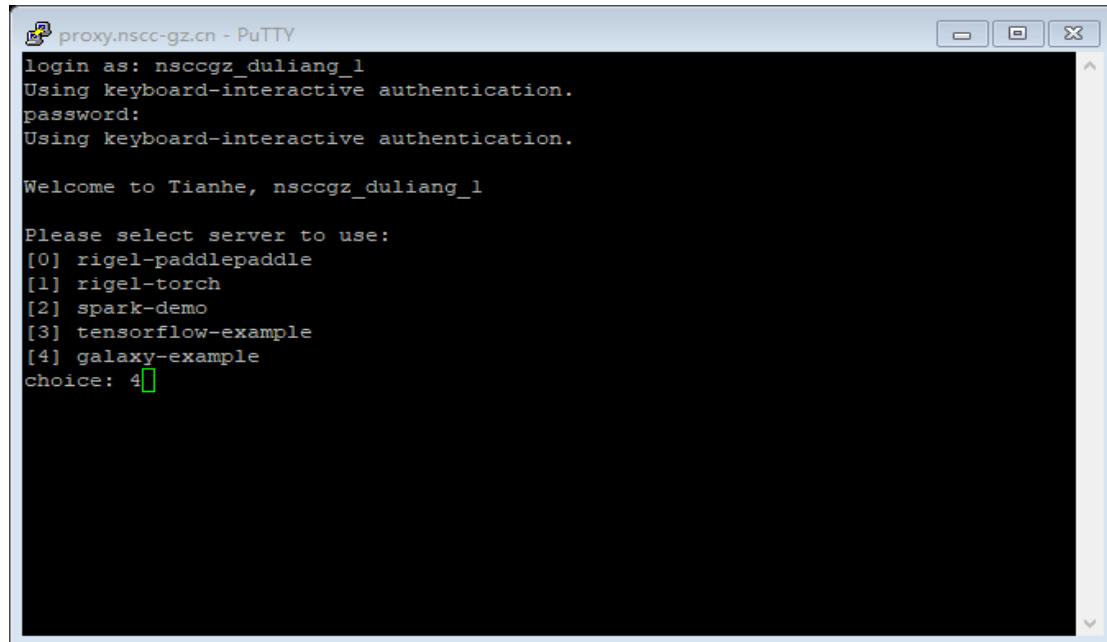


图 69 galaxy 登陆

**Step5 使用 SSH 工具登陆 galaxy 应用, 参考第九章 1.2 节 ssh 工具远程登录作业。**



```
proxy.nscc-gz.cn - PuTTY
login as: nsccgz_duliang_1
Using keyboard-interactive authentication.
password:
Using keyboard-interactive authentication.

Welcome to Tianhe, nsccgz_duliang_1

Please select server to use:
[0] rigel-paddlepaddle
[1] rigel-torch
[2] spark-demo
[3] tensorflow-example
[4] galaxy-example
choice: 4
```

图 70 ssh 工具登陆 galaxy

**Step6 运行 galaxy 示例程序**

点击 Galaxy UI 进入 galaxy 交互式引导环境

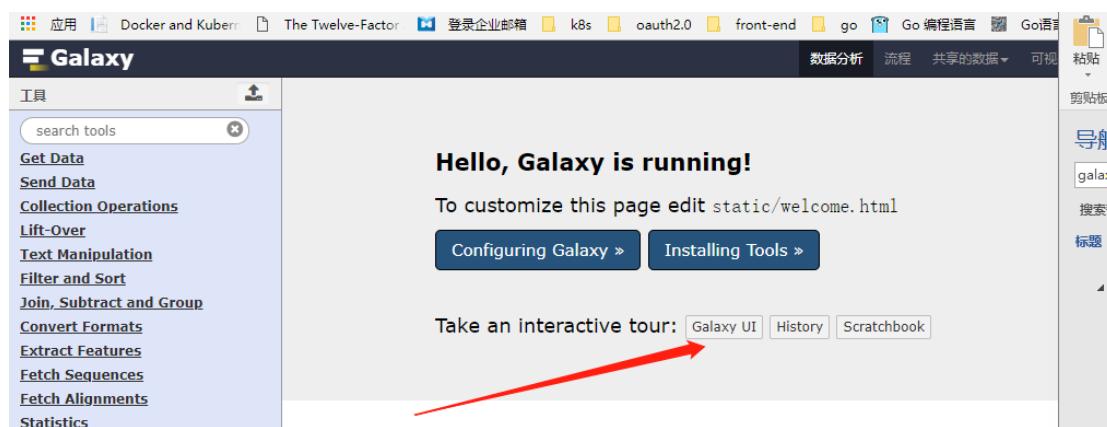


图 71 galaxy web ui

## 依据需要安装所需软件

The figure consists of two screenshots of the Galaxy web interface. The top screenshot shows the 'Categories of Valid Repositories by Category' page, which lists various software categories with their descriptions and the number of valid repositories. A red arrow points to the 'Assembly' category, and a red '3' is overlaid on the arrow. The bottom screenshot shows the details of a specific tool repository, 'bamtools\_wrappers'. It includes sections for repository information, dependencies, contents, and tool metadata. A red arrow points to the 'Install to Galaxy' button.

| Name                          | Description  | Valid repositories |
|-------------------------------|--|--------------------|
| Assembly                      | Tools for working with assemblies  | 97                 |
| ChIP-seq                      | Tools for analyzing and manipulating ChIP-seq data.                            | 56                 |
| Combinatorial Selections      | Tools for combinatorial selection  | 7                  |
| Computational chemistry       | Tools for use in computational chemistry                                       | 54                 |
| Constructive Solid Geometry   | Tools for constructing and analyzing 3-dimensional shapes and their properties | 11                 |
| Convert Formats               | Tools for converting data formats  | 98                 |
| Data Export                   | Tools for exporting data to various destinations                               | 1                  |
| Data Managers                 | Utilities for Managing Galaxy's built-in data cache                            | 46                 |
| Data Source                   | Tools for retrieving data from external data sources                           | 68                 |
| Entomology                    | Tools that involve insect studies  | 1                  |
| Epigenetics                   | Tools for analyzing Epigenetic/Epigenomic datasets                             | 16                 |
| Fasta Manipulation            | Tools for manipulating fasta data  | 100                |
| Fasta Manipulation            | Tools for manipulating fastq data  | 80                 |
| Flow Cytometry Analysis       | Tools for manipulating and analyzing FCS files                                 | 24                 |
| Genome annotation             | Tools for annotating genomic information                                       | 2                  |
| Genome editing                | Tools for analyzing genome editing data  | 6                  |
| Genome-Wide Association Study | Utilities to support Genome-wide association studies                           | 17                 |
| Genomic Interval Operations   | Tools for operating on genomic intervals                                       | 52                 |

图 72 galaxy web 安装软件

以上演示了使用 Galaxy 生物信息分析平台的使用流程和相关必要步骤，关于 galaxy 的相关知识请移步至 [galaxy 官网](#) 或参考以下文档 [galaxy 生物信息分析平台简明使用手册](#)；[Galaxy 数据处理平台简介](#)

## 5. Tensorflow 使用说明

与以上创建 Spark、Rstudio 等应用相似，在创建 Tensorflow 等深度学习类应用时，创建流程和使用方法基本相同，在此不做过多赘述。

## Step1 创建 Tenorflow 应用

The screenshot shows the Tensorflow application submission interface. It includes fields for job name, cluster selection (M40 GPU cluster selected), job configuration (6核/30GB/1块 selected), image selection (Tensorflow Centos selected), and a submit button. To the right, there's a summary section showing resource usage (1 CPU \* 1 + GPU \* 32 + 内存 \* 0.01), cost (38 核时/小时), and application logs (tensorflow-single-gpu教程, 皆无tensorflow-single-gpu文档教程). A note at the bottom states: "kubernetes应用不使用存储管理页面上传数据".

图 73 Tensorflow 应用表单

## Step2 创建应用成功，获取应用详情页

The screenshot shows the Tensorflow job details page. It displays basic information like job name (tensorflow-example), type (tensorflow-single-gpu), cluster (m40), submission time (2018-10-30 15:29:15), and status (正在运行). Below this is a file information section and a access入口 (Access入口) section. The access入口 section lists four services: terminal, jupyter\_web, tensorflow-example, and tensorboard, each with its corresponding URL (e.g., http://231.proxy.nscc-gz.cn for terminal).

图 74 Tensorflow 作业详情

### Step3 访问应用

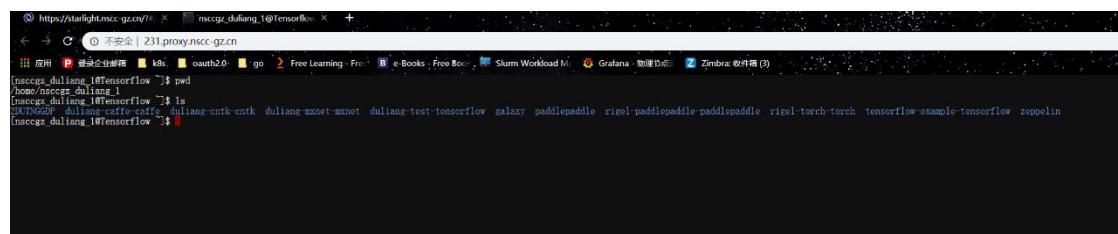


图 75 访问应用

**Step4** 利用 Jupyter 工具访问应用(密码为:jupyter)

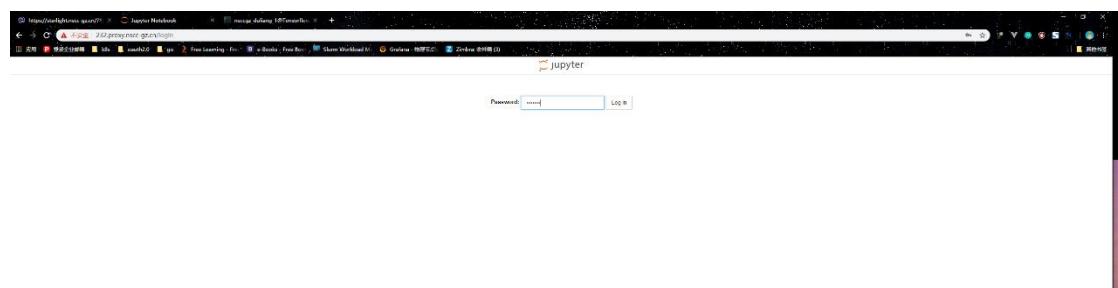


图 76 Jupyter 访问

## Step5 利用 Jupyter 工具访问应用



图 77 Jupyter 访问

**Step6 利用 SSH 工具访问应用。参考第九章 1.2 节 ssh 工具远程登陆作业。**

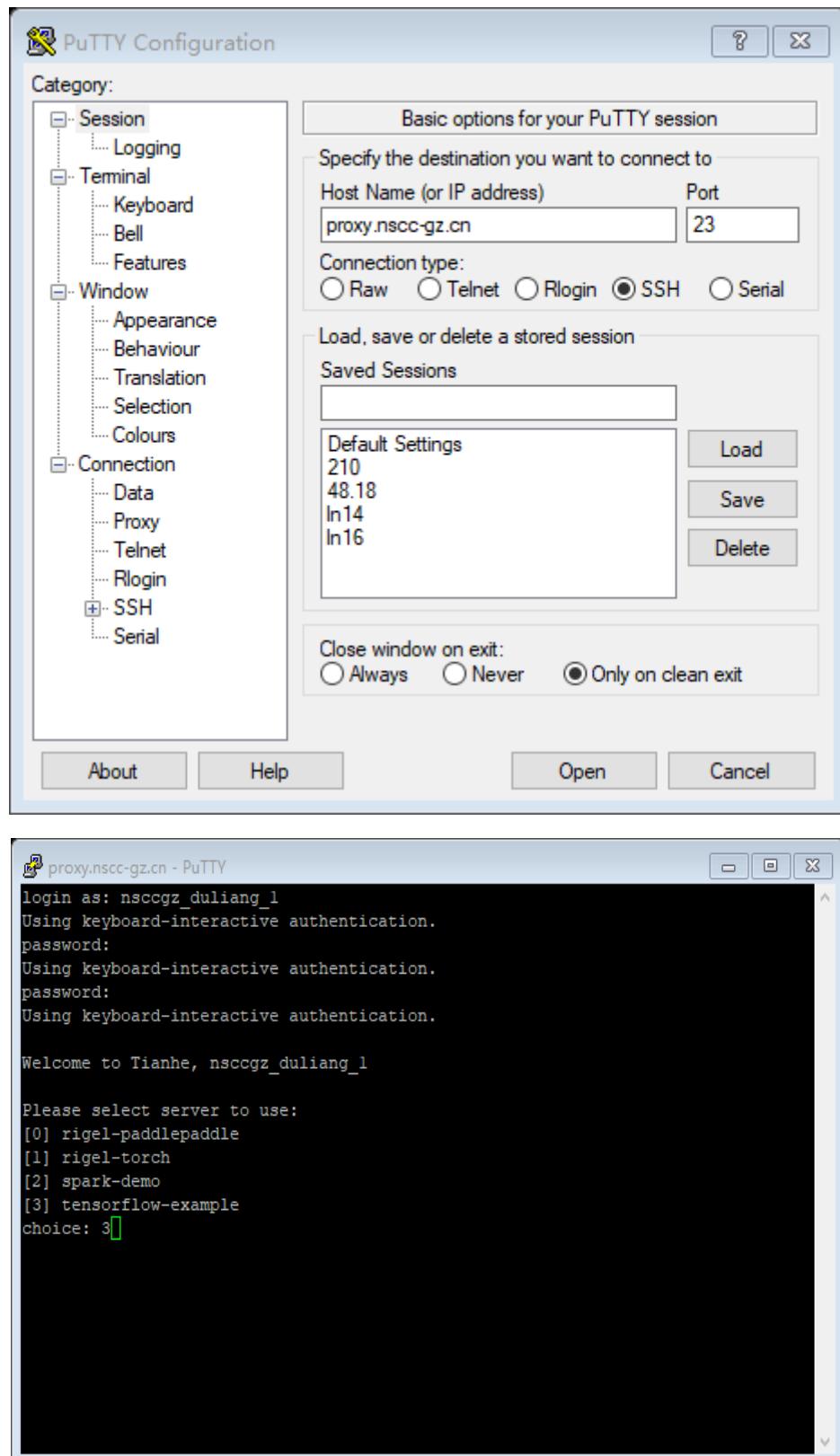


图 78 ssh 连接工具

以上以 [Tensorflow](#) 为例演示了创建深度学习类应用的创建以及使用流程，关于 Tensorflow 的相关知识请移步至 [Tensorflow 官网](#)，由于以下深度学习应用如 [Caffe](#)、[CNTK](#)、[MXNet](#)、[Torch](#)、[PaddlePaddle](#) 与 [tensorflow](#) 类似，其相应的示例不再做过多赘述。

此外，在用户创建完应用后，可以在作业列表中查看已创建的应用，可以通过此种方式再次返回至已创建的应用中。



| 作业id/sid/集群名称                    | 作业名称                 | 作业状态 | 提交时间                | 操作  |
|----------------------------------|----------------------|------|---------------------|---|
| jobid/sid<br>YKkkccITPAvbfjpo1RU | 7672 rstudio-test    | 运行中  | 2018-11-19 11:32:22 | <a href="#">查看详情</a> <a href="#">关闭作业</a> |
| jobid/sid<br>Tnogcs1lindenNBfHo  | 7672 nsccgz_duoliang | 运行中  | 2018-12-06 11:35:24 | <a href="#">查看详情</a> <a href="#">关闭作业</a> |

图 79 作业管理

## 6. Slurm 使用说明

SLURM 是一种可用于大型计算节点集群的高度可伸缩和容错的集群管理器和作业调度系统。SLURM 维护着一个待处理工作的队列并管理此工作的整体资源利用。它还以一种排他或非排他的方式管理可用的计算节点（取决于资源的需求）。最后，SLURM 将作业分发给一组已分配的节点来执行工作并监视平行作业至其完成。

本质上，SLURM 是一个强健的集群管理器（更关注于对功能丰富性的需求方面），它高度可移植、可伸缩至大型节点集群、容错好，而且更重要的是它是开源的。SLURM 最早是一个开源的资源管理器，由几家公司（包括 Lawrence Livermore National Laboratory）协作开发。如今，SLURM 已经成为了很多最强大的超级计算机上使用的领先资源管理器。

下面以 slurm 中常见命令和应用场景为例，演示创建和使用 slurm 步骤和相关流程

## Step1 创建 slurm 应用

The screenshot shows the 'Slurm' application creation interface. It includes fields for job name, cluster selection, resource configuration, application details, and a note about Kubernetes storage management.

- 作业名称:** 唯一-不可重复
- 集群选择:** 天河 CPU 集群
- 作业配置:** 6核/6GB (selected)
- 镜像选择:** 暂无可用镜像
- 费用:** 6 核时/小时
- 应用数据:**
  - 暂无slurm视频教程
  - 暂无slurm文档教程
- 提示:** kubernetes应用不使用存储管理页面上传数据

图 80 slurm 应用表单

## Step2 进入作业详情，点击 web-terminal 进入应用

The screenshot shows the 'Job Details' page for a submitted Slurm job. It displays basic information, file information, and access入口 (terminal and slurm-ssh).

**基本信息:**  
作业名称: slurm  
作业 ID: 790  
集群: k8s  
提交时间: 2018  
提交账号: rigel  
运行状态: 运行中

**文件信息:**  
未获取到相关文件信息

**访问入口:**  
terminal: <http://starlight.nscc-gz.cn:42180>  
slurm-ssh: <ssh://starlight.nscc-gz.cn:41806>

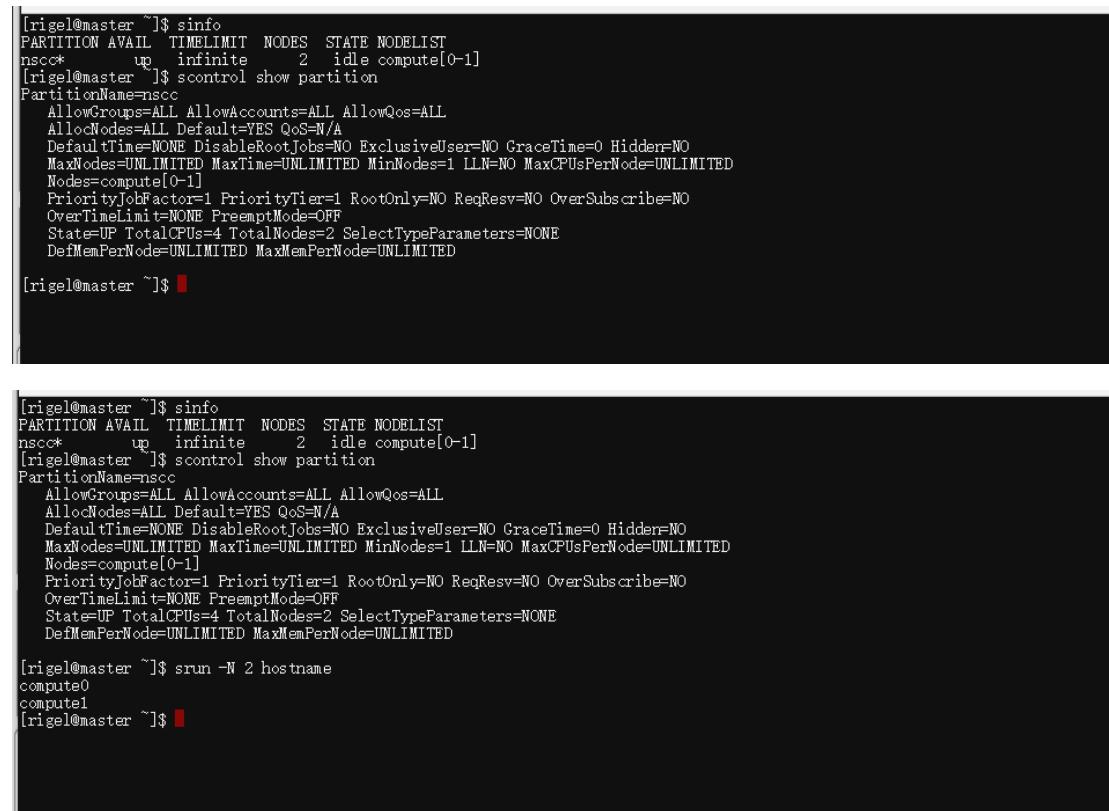
The screenshot shows a terminal window titled 'starlight.nscc-gz.cn:42180'. It displays the output of a 'hostname' command on a Slurm compute node.

```
rigel@master:~$ hostname
PARTITION_AVAIL TIMELIMIT NODES STATE MODELIST
rigel@master:~$ hostname
push: hostname: command not found
rigel@master:~$ run hostname
compute0
rigel@master:~$ run -k 2 hostname
compute0
compute0
rigel@master:~$
```

图 81 slurm 作业详情

### Step3 以 sinfo 、 srun 等命令为例，获取 slurm 集群信息和提交作业

依次输入 sinfo、scontrol show partition 命令



```
[rigel@master ~]$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
nscc* up infinite 2 idle compute[0-1]
[rigel@master ~]$ scontrol show partition
PartitionName=nscc
AllowGroups=ALL AllowAccounts=ALL AllowQos=ALL
AllocNodes=ALL Default=YES QoS=N/A
DefaultTime=NONE DisableRootJobs=NO ExclusiveUser=NO GraceTime=0 Hidden=NO
MaxNodes=UNLIMITED MaxTime=UNLIMITED MinNodes=1 LLN=NO MaxCPUsPerNode=UNLIMITED
Nodes=compute[0-1]
PriorityJobFactor=1 PriorityTier=1 RootOnly=NO ReqResv=NO OverSubscribe=NO
OverTimeLimit=NONE PreemptMode=OFF
State=UP TotalCPUs=4 TotalNodes=2 SelectTypeParameters=NONE
DefMemPerNode=UNLIMITED MaxMemPerNode=UNLIMITED

[rigel@master ~]$ ■

[rigel@master ~]$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
nscc* up infinite 2 idle compute[0-1]
[rigel@master ~]$ scontrol show partition
PartitionName=nscc
AllowGroups=ALL AllowAccounts=ALL AllowQos=ALL
AllocNodes=ALL Default=YES QoS=N/A
DefaultTime=NONE DisableRootJobs=NO ExclusiveUser=NO GraceTime=0 Hidden=NO
MaxNodes=UNLIMITED MaxTime=UNLIMITED MinNodes=1 LLN=NO MaxCPUsPerNode=UNLIMITED
Nodes=compute[0-1]
PriorityJobFactor=1 PriorityTier=1 RootOnly=NO ReqResv=NO OverSubscribe=NO
OverTimeLimit=NONE PreemptMode=OFF
State=UP TotalCPUs=4 TotalNodes=2 SelectTypeParameters=NONE
DefMemPerNode=UNLIMITED MaxMemPerNode=UNLIMITED

[rigel@master ~]$ srun -N 2 hostname
compute0
compute1
[rigel@master ~]$ ■
```

图 82 slurm 命令操作

以上演示了使用 Slurm 高性能集群调度系统的使用流程和相关必要步骤，关于 slurm 的相关知识请移步至 [slurm 官网](#)

## 7. Caffe 使用说明

caffe 是一个开源的深度学习框架，可读性高，它允许利用自己的 CPU 或者 GPU 训练网络，不需要自己编写程序，只需要通过配置文件来指定网络。下面通过 caffe 官方示例演示使用 caffe 的相关流程和步骤，[其中程序和相关数据来自 caffe 官方 git 仓库](#)

## Step1 创建 caffe 应用

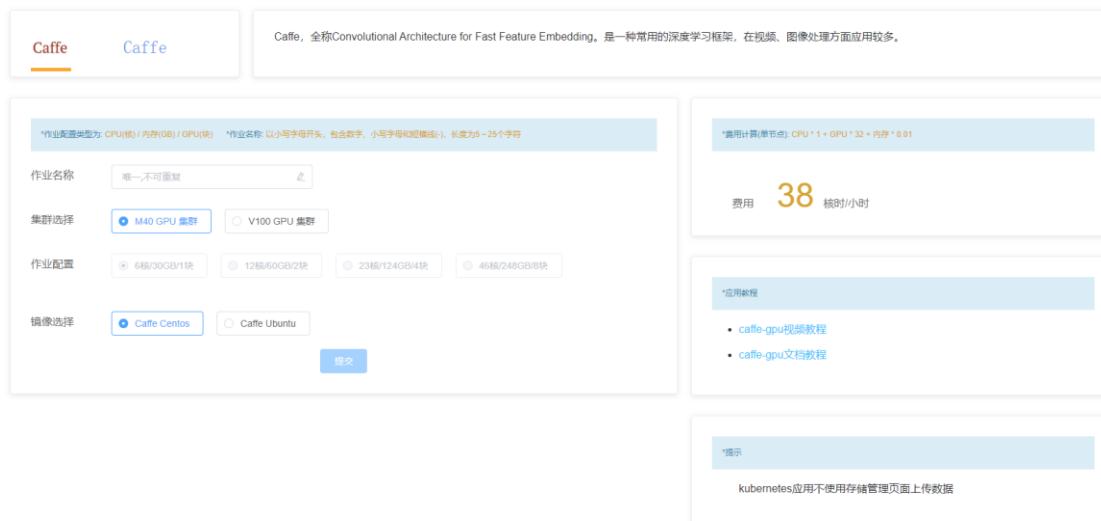


图 83 caffe 应用表单

## Step2 进入作业详情，点击 web-terminal 进入应用

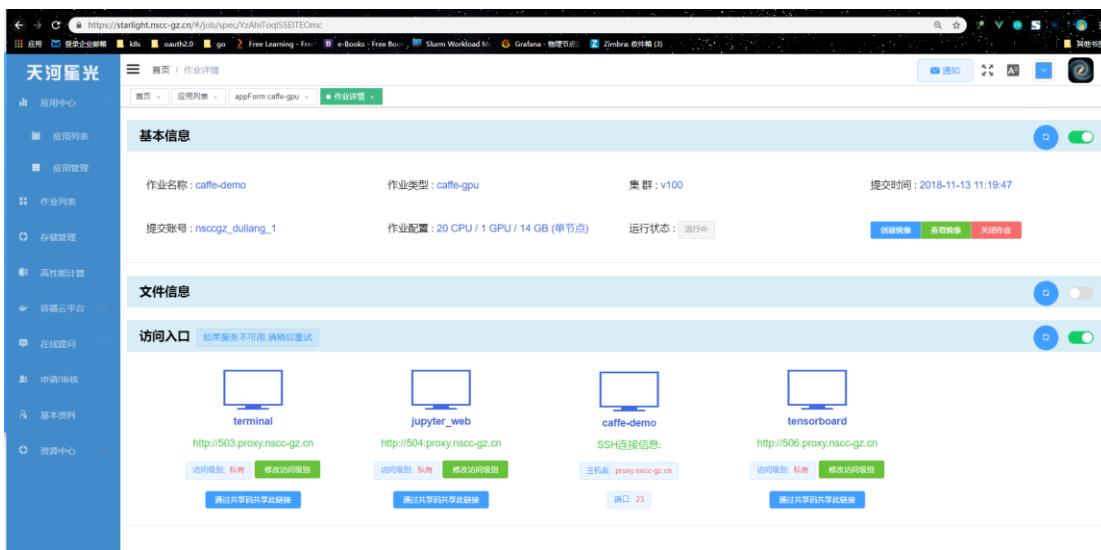
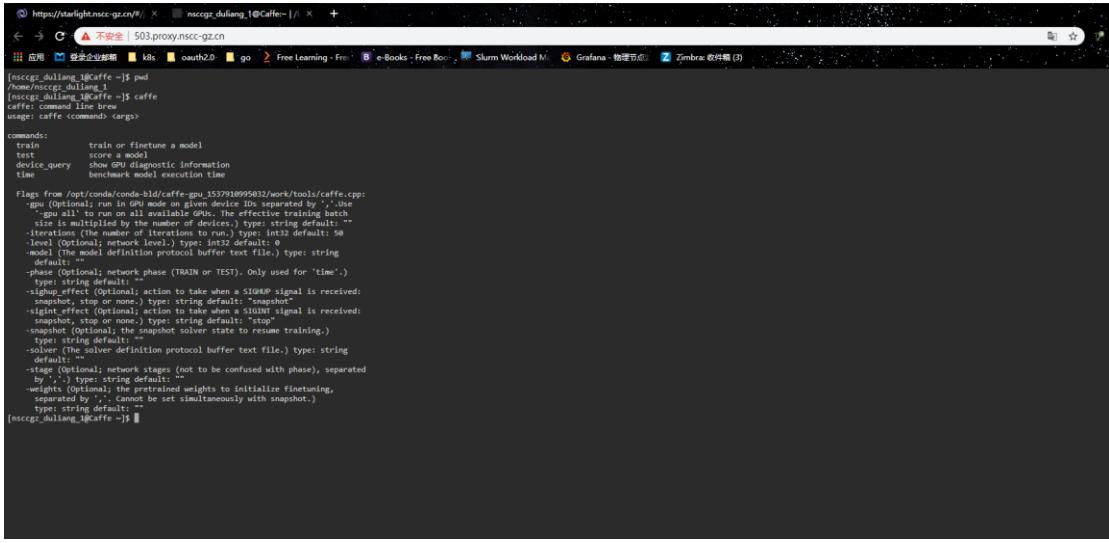


图 84 caffe 作业详情

A screenshot of a web browser window titled "nscg\_dujiang\_1@Caffe-:~\$". The address bar shows "http://starlight.nscg-gz.cn/#/" and "nscg\_dujiang\_1@Caffe-:~\$". The content area displays the Caffe command-line help text. It includes sections for "commands:", "train", "test", "device\_query", and "time". Under "train", it lists options like "-gpu", "-size", "-iterations", and "-model". Under "test", it lists "-gpu", "-size", and "-model". Under "device\_query", it lists "-gpu". Under "time", it lists "-model". The text ends with "(nscg\_dujiang\_1@Caffe-:~\$)".

```
[nscg_dujiang_1@Caffe-:~$]ls
[nscg_dujiang_1@Caffe-:~$]jps
[nscg_dujiang_1@Caffe-:~$]java
[nscg_dujiang_1@Caffe-:~$]java
usage: java [options] <args>

commands:
  train      train or finetune a model
  test       score a model
  device_query show GPU diagnostic information
  time       benchmark model execution time

Flags from /opt/code/bld/caffe-gpu_1537918995032/work/tools/caffe.cpp:
  -gpu (Optional): run in GPU mode on given device IDs separated by commas. Use
    -gpu 0,1 to run on both devices. Note that the batch size is multiplied by the number of devices.) type: string default: ""
  -size (The number of iterations to run.) type: int32 default: 50
  -iterations (The number of iterations to run.) type: int32 default: 50
  -model (The model definition protocol buffer text file.) type: string
  -solver (The solver definition protocol buffer text file.) type: string
  -weights (Optional; network stages (not to be confused with phase), separated
            by ".") type: string default: ""
  -snapshot (Optional; the snapshot solver state to resume training.)
  -stop (Optional; the number of iterations to stop training.) type: int32
  -solver (The solver definition protocol buffer text file.) type: string
  -weights (Optional; network stages (not to be confused with phase), separated
            by ".") type: string default: ""
  -snapshot (Optional; the pretrained weights to initialize finetuning,
            separated by ".") type: string default: "" Cannot be set simultaneously with snapshot.
  -type (String defining the type of solver.) type: string default: "SGD"
  -solver (The solver definition protocol buffer text file.) type: string
(nscg_dujiang_1@Caffe-:~$)
```

图 85 caffe web-terminal

### Step3 Caffe 使用示例

示例程序位于/app/common/example/caffe 目录下，该程序来自 [caffe 官方 Git 仓库](#)。

首先需要拷贝示例程序到用户目录，执行命令为：

```
cp -r /app/comom/ex ample/caffe /home/your_home_dir/TargetDir/
```

## 远程登录，参考第九章 1.2 节 ssh 工具远程登录作业

```
nsccgz_duliang_1@Caffe:~  
login as: nsccgz_duliang_1  
Using keyboard-interactive authentication.  
password:  
Using keyboard-interactive authentication.ssh登录  
Welcome to Tianhe, nsccgz_duliang_1  
  
Please select server to use:  
[0] buyaoshan-duzong  
[1] cntk-example-ubuntu  
[2] rstudio-example  
[3] galaxy-example  
[4] caffe-demo  
[5] mxnet-example  
[6] torch-example  
[7] tensorflow-example  
[8] spark-example  
choice: 4  
Last login: Wed Nov 14 15:13:55 2018 from 192.168.1.4  
[nsccgz_duliang_1@Caffe ~]$ cp -r /app/common/example/caffe/ example/caffe/
```

选择caffe  
拷贝例子程序

The screenshot shows a web-based interface for managing files. The URL is 504.proxy.mcc.gz.cn/tree/caffe/caffe/examples. The interface includes a sidebar with various links like '首页', '豆瓣企业邮箱', 'kbs', 'cauth2.0', 'go', 'Free learning', 'Free', 'e-Books', 'Free Books', 'Slurm Workload M', 'Gratians', '敏捷开发', 'Zabbix 监控'. The main area shows a file tree under '/caffe/caffe/examples' with several notebooks listed:

| Name           | Last Modified | File size |
|----------------|---------------|-----------|
| 几秒钟.ipynb      | 1小时前          | 813 kB    |
| 1小时前.ipynb     | 1小时前          | 376 kB    |
| 1小时前.ipynb     | 1小时前          | 481 kB    |
| 1小时前.ipynb     | 1小时前          | 453 kB    |
| 1小时前.ipynb     | 1小时前          | 702 kB    |
| 1小时前.ipynb     | 1小时前          | 583 kB    |
| 1小时前.ipynb     | 1小时前          | 1.54 MB   |
| CMakeLists.txt | 1小时前          | 1.08 kB   |

```

In [1]: 1 caffe_root = '...' # this file should be run from /caffe_root/examples (otherwise change this line)
2
3 import os
4 os.chdir(caffe_root) # set the current directory to the root of Caffe
5 import caffe
6
7 caffe.set_device(0)
8 caffe.set_mode_gpu()
9
10 import numpy as np
11 from pylab import *
12 import cv2
13 import tensorflow
14
15 # Helper function for deprocessing preprocessed images, e.g., for display.
16 def deprocess_net_image(image):
17     image = image.copy() # don't modify directly
18     image -= 128 # scale to RGB
19     image = image.transpose(1, 2, 0) # CBR -> BRC
20     image += [128, 111, 104] # (approximately) undo mean subtraction
21
22     # clamp values in [0, 255]
23     image[image < 0], image[image > 255] = 0, 255
24
25     # round and cast from float32 to uint8
26     image = np.round(image)
27     image = image.astype(np.uint8)
28
29 return image

```

1. Setup and dataset download  
Download data required for this exercise.

图 86 caffe 例程

以上演示了使用 Caffe 的流程和相关必要步骤，关于 Caffe 的相关知识请移步至 [Caffe 官网](#)

## 8. 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为 mxnet 官方网站](#)

## Step1 创建 Mxnet 应用

The screenshot shows the configuration interface for creating an MXNet application. Key fields include:

- 作业名称 (Job Name): 唯一, 不可重复
- 集群选择 (Cluster Selection): M40 GPU 集群
- 作业配置 (Job Configuration): 64GB/300GB/1块
- 镜像选择 (Image Selection): MXNet CentOS
- 费用 (Cost): 38 核时·小时
- 应用教程 (Application Tutorials): mxnet-gpu 视频教程, mxnet-gpu 文档教程
- 提示 (Tip): kubernetes 应用不使用存储管理页面上传数据

图 87 mxnet 应用表单

## Step2 进入作业详情，点击 web-terminal 进入应用

The screenshot shows the job details page for the 'mxnet-example' job. Key details include:

- 基本信息 (Basic Information):
  - 作业名称: mxnet-example
  - 作业类型: mxnet-gpu
  - 集群: v100
  - 提交时间: 2018-11-13 11:22:56
  - 提交账号: nsccg2\_duliang\_1
  - 作业配置: 10 CPU / 1 GPU / 10 GB (单节点)
  - 运行状态: 运行中
- 文件信息 (File Information):
  - 访问入口 (Access Port):
    - terminal: http://507.proxy.nscc-gz.cn
    - jupyter\_web: http://508.proxy.nscc-gz.cn
    - mxnet-example: SSH连接信息: 主机: proxy.nscc-gz.cn, 端口: 23
    - tensorboard: http://510.proxy.nscc-gz.cn

图 88 mxnet 作业详情

```
ps aux | grep mxnet
echo "this is mxnet web-terminal"
echo "通过共享链接此终端"
```

图 89 mxnet web-terminal

### Step3 Mxnet 使用示例

示例程序位于/app/common/example/mxnet 目录下, 程序脚本为 run\_example.sh

首先需拷贝至用户目录下 执行:

```
cp -r /app/common/example/mxnet /home/your_home_dir/TargetDir
```

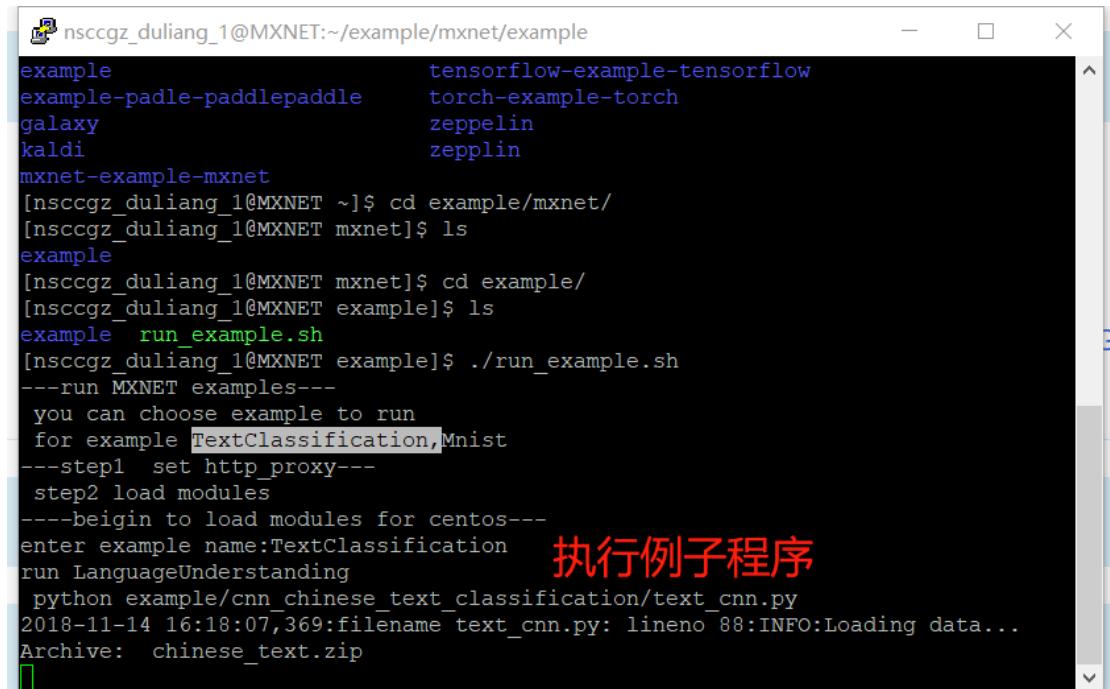
### 远程登陆, 参考第九章 1.2 节 ssh 工具远程登录作业

The screenshot shows a terminal window titled "nsccgz\_duliang\_1@MXNET:~". The session starts with a password prompt. The text "SSH登录" (SSH Login) is overlaid in red at the top right. Below the password prompt, the message "Welcome to Tianhe, nsccgz\_duliang\_1" appears. The terminal then displays a list of applications to choose from, with "[5] mxnet-example" highlighted. The text "选择应用" (Select Application) is overlaid in red in the center of the list. At the bottom of the list, the command "choice: 5" is shown. The final line of the terminal output is "[nsccgz\_duliang\_1@MXNET ~]\$ cp -r /app/common/example/mxnet/ example/mxnet/". The text "拷贝例子程序" (Copy example program) is overlaid in red at the bottom right of the terminal window.

```
login as: nsccgz_duliang_1
Using keyboard-interactive authentication.
password: SSH登录
Using keyboard-interactive authentication.

Welcome to Tianhe, nsccgz_duliang_1

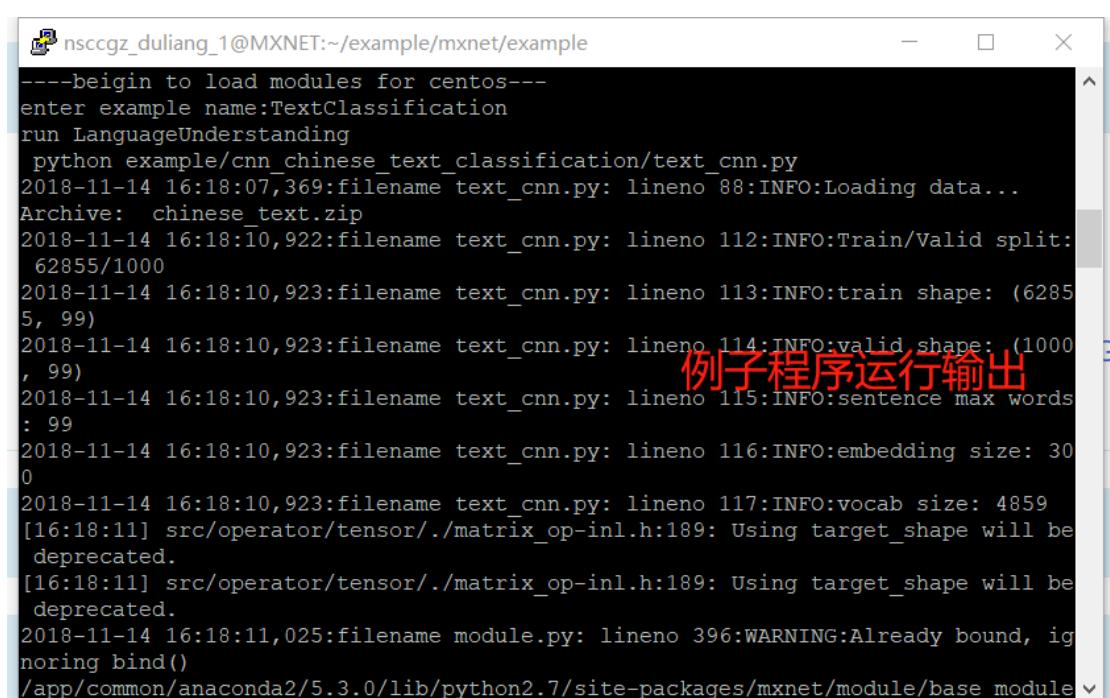
Please select server to use:
[0] buyaoshan-duzong
[1] cntk-example-ubuntu
[2] rstudio-example
[3] galaxy-example
[4] caffe-demo
[5] mxnet-example
[6] torch-example
[7] tensorflow-example
[8] spark-example
choice: 5
Last login: Wed Nov 14 14:22:12 2018 from 192.168.0.4
[nsccgz_duliang_1@MXNET ~]$ cp -r /app/common/example/mxnet/ example/mxnet/ 拷贝例子程序
```



```

nscggz_duliang_1@MXNET:~/example/mxnet/example
example          tensorflow-example-tensorflow
example-paddle-paddlepaddle  torch-example-torch
galaxy           zeppelin
kaldi            zeppelin
mxnet-example-mxnet
[nscggz_duliang_1@MXNET ~]$ cd example/mxnet/
[nscggz_duliang_1@MXNET mxnet]$ ls
example
[nscggz_duliang_1@MXNET mxnet]$ cd example/
[nscggz_duliang_1@MXNET example]$ ls
example  run_example.sh
[nscggz_duliang_1@MXNET example]$ ./run_example.sh
---run MXNET examples---
you can choose example to run
for example TextClassification,Mnist
---step1  set http_proxy---
step2 load modules
----begin to load modules for centos---
enter example name:TextClassification      执行例子程序
run LanguageUnderstanding
python example/cnn_chinese_text_classification/text_cnn.py
2018-11-14 16:18:07,369:filename text_cnn.py: lineno 88:INFO:Loading data...
Archive:  chinese_text.zip

```

```

nscggz_duliang_1@MXNET:~/example/mxnet/example
----begin to load modules for centos---
enter example name:TextClassification
run LanguageUnderstanding
python example/cnn_chinese_text_classification/text_cnn.py
2018-11-14 16:18:07,369:filename text_cnn.py: lineno 88:INFO:Loading data...
Archive:  chinese_text.zip
2018-11-14 16:18:10,922:filename text_cnn.py: lineno 112:INFO:Train/Valid split:
62855/1000
2018-11-14 16:18:10,923:filename text_cnn.py: lineno 113:INFO:train shape: (6285
5, 99)
2018-11-14 16:18:10,923:filename text_cnn.py: lineno 114:INFO:valid shape: (1000
, 99)
2018-11-14 16:18:10,923:filename text_cnn.py: lineno 115:INFO:sentence max words
: 99
2018-11-14 16:18:10,923:filename text_cnn.py: lineno 116:INFO:embedding size: 30
0
2018-11-14 16:18:10,923:filename text_cnn.py: lineno 117:INFO:vocab size: 4859
[16:18:11] src/operator/tensor./matrix_op-inl.h:189: Using target_shape will be
deprecated.
[16:18:11] src/operator/tensor./matrix_op-inl.h:189: Using target_shape will be
deprecated.
2018-11-14 16:18:11,025:filename module.py: lineno 396:WARNING:Already bound, ig
noring bind()
/app/common/anaconda2/5.3.0/lib/python2.7/site-packages/mxnet/module/base_module

```

图 90 MXnet 例程

以上演示了使用 Mxnet 深度学习框架的使用流程和相关必要步骤，关于 Mxnet 的相关知识请移步至 [Mxnet 官网](#)

## 9. Cntk 使用说明

Microsoft Cognitive Toolkit 是一个快速，易用的深度学习软件包，但与 TensorFlow 相比，其范围有限。它有各种各样的模型和算法，极好的支持 Python 和 Jupyter 笔记本，一个有趣的声明性神经网络配置语言 BrainScript，以及在 Windows 和 Ubuntu Linux 环境下自动部署，CNTK 可以解决如类别分析，语音识别，图像识别等问题，并且具有快速、训练模型简单等优点，下面以官方示例演示使用 Cntk 的相关流程和步骤，[其中程序代码和数据源自官方 git 仓库](#)

### Step1 创建 cntk 应用

The screenshot shows the Microsoft CNTK application configuration interface. It includes several sections:

- 作业名称:** 唯一,不可重复 (唯一,不可重复)
- 集群选择:** M40 GPU 集群 (selected) / V100 GPU 集群
- 作业配置:** 6核/30GB/1块 (selected) / 12核/60GB/2块 / 23核/124GB/4块 / 46核/248GB/8块
- 镜像选择:** CNTK CentOS (selected) / CNTK Ubuntu
- 费用:** 38 核时/小时
- 应用教程:**
  - cntk-gpu 视频教程
  - cntk-gpu 文档教程
- 提示:** kubernetes 应用不使用存储管理页面上传数据

图 91 cntk 应用表单

## Step2 进入作业详情，点击 web-terminal 进入应用

The screenshot shows the 'Job Details' page of the Tianhe Starlight system. Key information includes:

- 作业名称: cntk-example-ubuntu
- 作业类型: cntk-gpu
- 集群: y100
- 提交时间: 2018-11-09 15:35:07
- 作业账号: nsccgz\_duliang\_1
- 作业配置: 24 CPU / 1 GPU / 8 GB (单节点)
- 运行状态: 运行中

The '访问入口' (Access Port) section lists four options:

- terminal: http://465.proxy.nscc-gz.cn (highlighted with a red box)
- jupyter\_web: http://466.proxy.nscc-gz.cn
- cntk-example-ubuntu: SSH连接信息 (SSH connection information)
- tensorboard: http://468.proxy.nscc-gz.cn

图 92 cntk 作业详情

The top screenshot shows a terminal session on the CNTK web-terminal:

```
nsccgz_duliang_1@CNTK:~$ echo "this is web terminal"
this is web terminal
nsccgz_duliang_1@CNTK:~$ pwd
/home/nsccgz_duliang_1
nsccgz_duliang_1@CNTK:~$
```

The bottom screenshot shows the Jupyter notebook interface with a file list:

| Name                               | Last Modified | File size    |
|------------------------------------|---------------|--------------|
| logs                               | 4小时前          |              |
| moveData                           | 19小时前         |              |
| recommender_system.inference.model | 17小时前         | 4.7MB 15.1kB |
| Untitled.ipynb                     | Running 17小时前 | 12.3 kB      |
| Untitled.ipynb                     | Running 17小时前 | 3.1 kB       |
| jupyter.log                        | 几秒钟前          |              |

图 93 cntk web-terminal

## Step3 Cntk 使用示例

```
proxy.nscc-gz.cn - PuTTY
login as: nsccgz_duliang_1
Using keyboard-interactive authentication.
password: ssh登录
Using keyboard-interactive authentication.

Welcome to Tianhe, nsccgz_duliang_1

Please select server to use:
[0] buyashan-duzong
[1] cntk-example-pyunit
[2] istrudio-example
[3] galaxy-example
[4] caffe-demo
[5] mxnet-example
[6] torch-example
[7] tensorflow-example
[8] spark-example
choice: 1 选择cntk应用
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-29-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Wed Nov 14 14:17:08 2018 from 192.168.0.4
nsccgz_duliang_1@CNTK:~$ cp -r /app/common/example/cntk/ example/cntk/
```

**拷贝例子程序到用户目录**

```
[0] spark-example
choice: 1
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-29-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Wed Nov 14 14:17:08 2018 from 192.168.0.4
nsccgz_duliang_1@CNTK:~$ cd example/cntk/
nsccgz_duliang_1@CNTK:~/example/cntk$ ls
examples run_example.sh
nsccgz_duliang_1@CNTK:~/example/cntk$ ls
examples run_example.sh
nsccgz_duliang_1@CNTK:~/example/cntk$ ./run_example.sh
examples run_example.sh
you can choose example to run
for example [language]understanding LogisticRegression_FunctionalAPI
--step1 see http_proxy---
step2 load model
use pretrained modules for ubuntu---
enter example name[language]understanding
run LanguageUnderstanding
python examples/LanguageUnderstanding/ATIS/Python/languageUnderstanding.py
/app/common/anaconda2/5.3.0/lib/python2.7/site-packages/cntk/_cntk_py_init.py:56: UserWarning: Unsupported Linux distribution (ubuntu-18.04). CNTK supports Ubuntu 16.04 and above, only.
warnings.warn('Unsupported Linux distribution (%s-%s. CNTK supports Ubuntu 16.04 and above, only.' % (_my_distro_, _my_distro_ver_))
/app/common/anaconda2/5.3.0/lib/python2.7/site-packages/cntk/_cntk_py_init.py:50: UserWarning:
#####
##### Missing optional dependency ( MKL ) #####
CNTK may crash if the component that depends on those dependencies is loaded.
Visit https://docs.microsoft.com/en-us/cognitive-toolkit/Setup-Linux-Python#mkl for more information.
#####
warning(WARNING MSG % (' MKL ', 'https://docs.microsoft.com/en-us/cognitive-toolkit/Setup-Linux-Python#mkl'))#
#####
##### Missing optional dependency ( GPU-Specific ) #####
CNTK may crash if the component that depends on those dependencies is loaded.
Visit https://docs.microsoft.com/en-us/cognitive-toolkit/Setup-Linux-Python#optional-gpu-specific-packages for more information.
#####
If you intend to use CNTK without GPU support, you can ignore the (likely) GPU-specific warning!
#####
warnings.warn(WARNING MSG % ('GPU ONLY ', 'https://docs.microsoft.com/en-us/cognitive-toolkit/Setup-Linux-Python#optional-gpu-specific-packages'))
/app/common/anaconda2/5.3.0/lib/python2.7/site-packages/cntk/_cntk_py_init.py:102: UserWarning:
#####
##### Missing optional dependency ( OpenCV ) #####
例程程序运行并输出
```

```

https://starlight.nscc-gz.cn/?# /bin/bash (CNTK)
C ▲ 不安全 | 465.proxy.nscc-gz.cn
应用 登录企业邮箱 k8s oauth2.0 go Free Learning - Fre e-Books - Free Books Slurm Workload M Grafana - 物理节点 Zimbra 收件箱 (3)

nsccg_duliang_0@CNTK:~$ cd example/cntk/
nsccg_duliang_0@CNTK:~/example/cntk$ ./run_example.sh
--run CNTK examples
you can choose example to run
for example: LogisticRegressionUnderstanding,LogisticRegression_FunctionalAPI
--step1 set http_proxy --
step2 Load module
----begin to load modules for ubuntu--
enter example name:logisticRegression_FunctionalAPI
run LogisticRegression_FunctionalAPI
python example/examples/1stSteps/LogisticRegress_FunctionalAPI.py
/app/common/anaconda2/5.3.0/lib/python2.7/site-packages/cntk/cntk_py_init.py:56: UserWarning: Unsupported Linux distribution (ubuntu-18.04). CNTK supports Ubuntu 16.04 and above, only.
    warnings.warn('Unsupported Linux distribution (%s-%s). CNTK supports Ubuntu 16.04 and above, only.' % (_my_distro__, __my_distro_ver__))
/app/common/anaconda2/5.3.0/lib/python2.7/site-packages/cntk/cntk_py_init.py:90: UserWarning:
#####
##### Missing optional dependency ( MKL ) #####
CNTK may crash if the component that depends on those dependencies is loaded.
Visit https://docs.microsoft.com/en-us/cognitive-toolkit/Setup-Linux-Python#mkl for more information.
#####
If you intend to use CNTK without GPU support, you can ignore the (likely) GPU-specific warning!
#####

warnings.warn(WARNING_MSG % ('GPU-Specific', 'https://docs.microsoft.com/en-us/cognitive-toolkit/Setup-Linux-Python#optional-gpu-specific-packages'))
/app/common/anaconda2/5.3.0/lib/python2.7/site-packages/cntk/cntk_py_init.py:182: UserWarning:
#####
##### Missing optional dependency ( OpenCV ) #####
CNTK may crash if the component that depends on those dependencies is loaded.
Visit https://docs.microsoft.com/en-us/cognitive-toolkit/Setup-Linux-Python#optional-opencv for more information.
#####
warnings.warn(WARNING_MSG % (' OpenCV ', 'https://docs.microsoft.com/en-us/cognitive-toolkit/Setup-Linux-Python#optional-opencv'))
Selected GPU[0] Tesla V100-SW2-16GB as the process wide default device.

Build info:
Built time: Sep 14 2018 01:27:52
Last modified date: Fri Sep 14 00:54:25 2018
Build type: release
Build target: GPU
With ASGD: yes
Math lib: mkl
CUDA version: 9.0.0
CUDNN version: 7.2.1

```

图 94 cntk 例程

以上演示了使用 Cntk 深度学习框架的使用流程和相关必要步骤，关于 Cntk 的相关知识请移步至 [Cntk 官网](#)

## 10. 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】](#)

## Step1 创建 paddle 应用

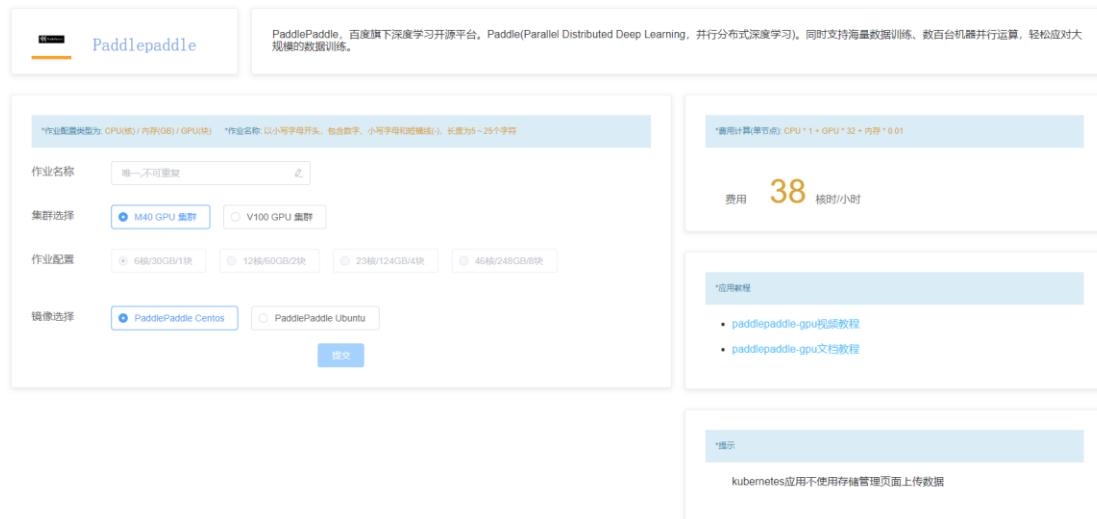


图 95 paddle 应用表单

## Step2 进入作业详情，点击 web-terminal 进入应用

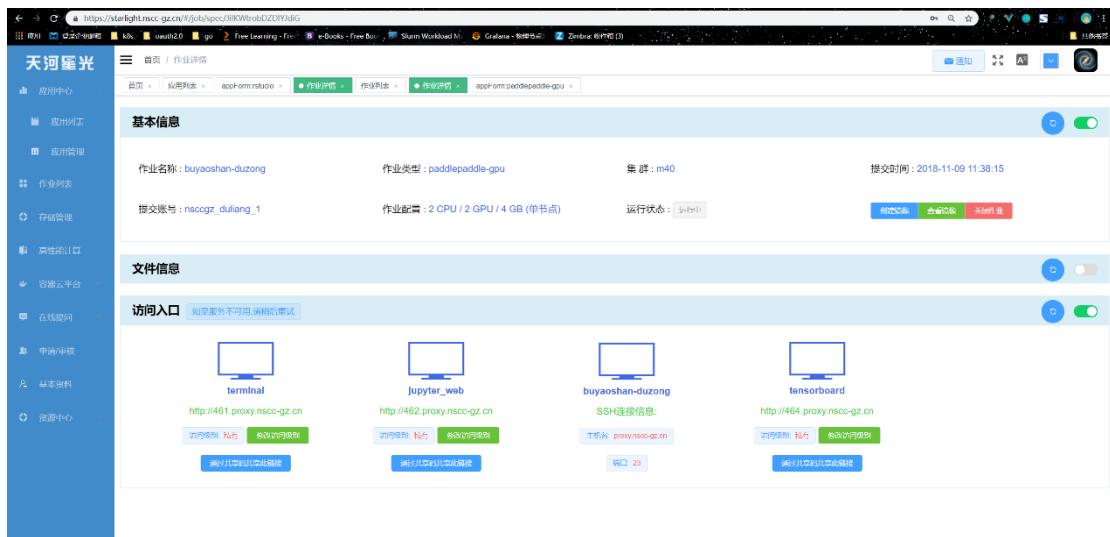
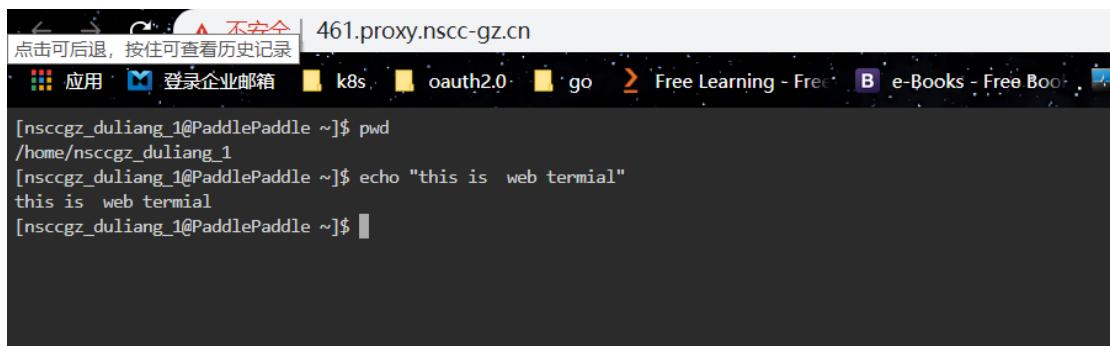


图 96 paddle 作业详情



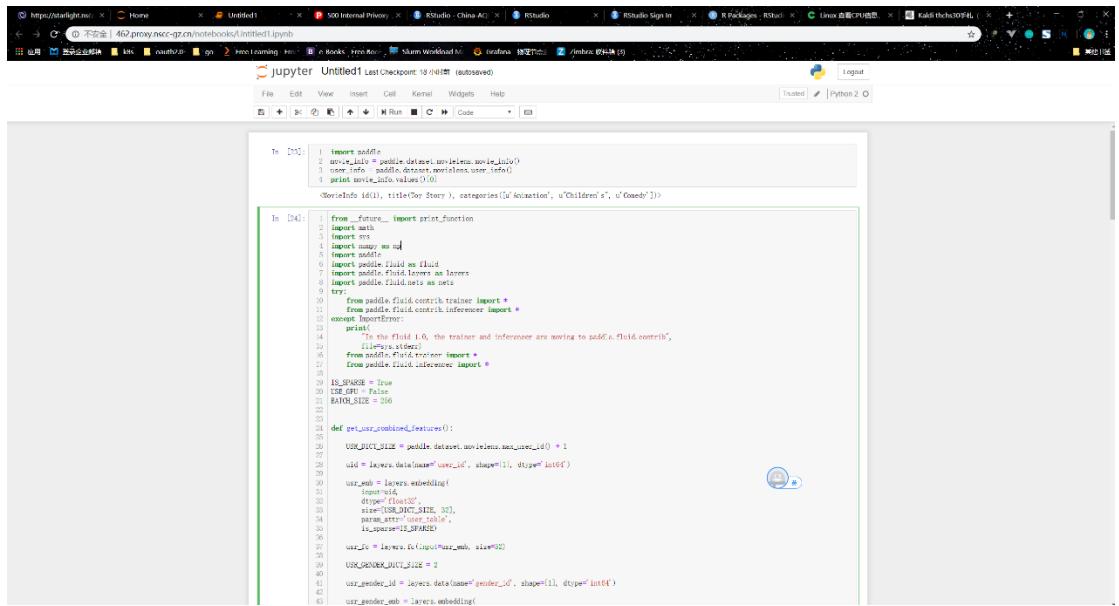


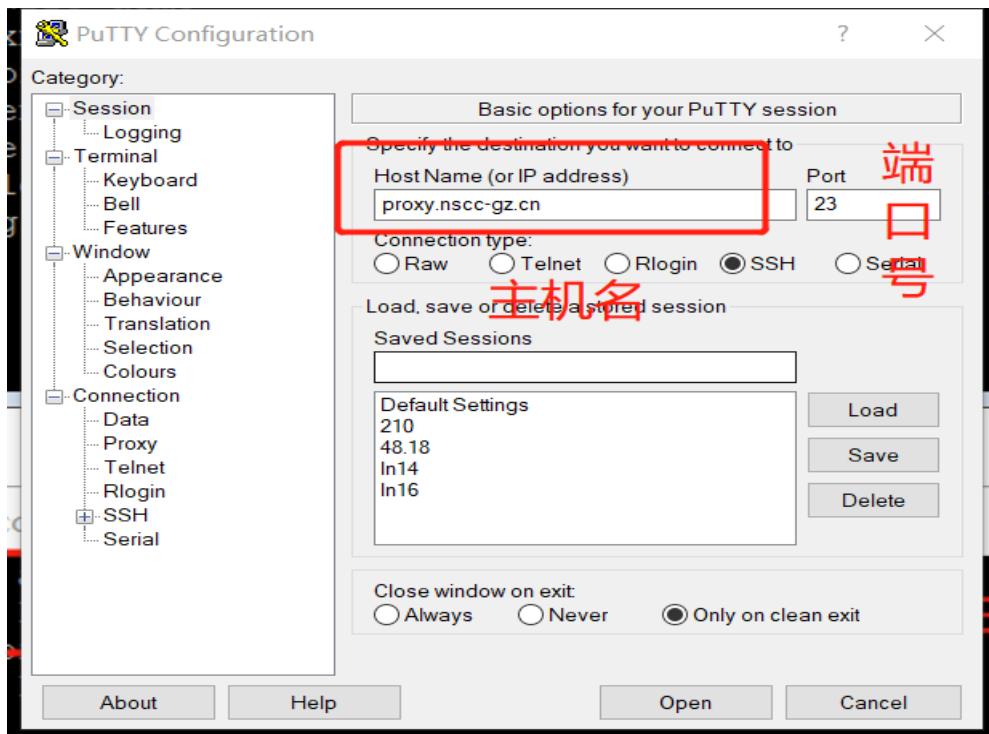
图 97 paddle web-terminal

### Step3 PaddlePaddle 使用示例

示例程序位于/app/common/example/paddlepaddle 目录下，程序脚本为 **run\_example.sh**

执行：`cp -r /app/common/example/paddlepaddle /home/your_home_dir/targetdir/`

远程登录，参考第九章 1.2 节 ssh 工具远程登录作业



The screenshot shows a terminal window titled 'nsccgq\_duliang\_1@PaddlePaddle:~'. The text in the window is as follows:

```
login as: nsccgq_duliang_1
Using keyboard-interactive authentication.
password:
Using keyboard-interactive authentication.

Welcome to Tianhe, nsccgq_duliang_1

Please select server to use:
[0] buyaoshan-duzong
[1] cntk-example-ubuntu
[2] centos-spark
[3] rstudio-example
[4] galaxy-example
[5] caffe-demo
[6] mxnet-example
[7] torch-example
[8] tensorflow-example
choice: 0
Last login: Tue Nov 13 09:59:11 2018 from 192.168.0.2
[nsccgq_duliang_1@PaddlePaddle ~]$ cp -r /app/common/example/paddlepaddle/ paddle-example
```

Annotations in red highlight specific parts of the terminal output:

- A red box surrounds the first two lines of text: 'login as: nsccgq\_duliang\_1' and 'Using keyboard-interactive authentication.' with the text '用户登录' (User Login).
- The number '0' in the list of applications is highlighted with the text '选择应用' (Select Application).
- The final line of text '[nsccgq\_duliang\_1@PaddlePaddle ~]\$ cp -r /app/common/example/paddlepaddle/ paddle-example' is highlighted with the text '拷贝例子程序' (Copy Example Program).

```

login as: nsccgz_duliang_1
Using keyboard-interactive authentication.
password:
Using keyboard-interactive authentication.

Welcome to Tianhe, nsccgz_duliang_1

Please select server to use:
[0] buyaoshan-duzong
[1] cntk-example-ubuntu
[2] centos-spark
[3] rstudio-example
[4] galaxy-example
[5] caffe-demo
[6] mxnet-example
[7] torch-example
[8] tensorflow-example
choice: 0
Last login: Wed Nov 14 10:34:03 2018 from 192.168.2.6
[nsccgz_duliang_1@PaddlePaddle ~]$ cd example/paddlepaddle/
[nsccgz_duliang_1@PaddlePaddle 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
Train cost, Step 0, Cost 698.395325
Test cost, Step 0, Cost 289.150106
Train cost, Step 10, Cost 616.436951
Train cost, Step 20, Cost 394.303345
Train cost, Step 30, Cost 466.264832
Train cost, Step 40, Cost 710.620178
Train cost, Step 50, Cost 657.989624
Train cost, Step 60, Cost 573.563354
Train cost, Step 70, Cost 406.197693
Train cost, Step 80, Cost 447.583160
Train cost, Step 90, Cost 439.348206
Train cost, Step 100, Cost 541.102173
Test cost, Step 100, Cost 174.462802
Train cost, Step 110, Cost 648.183411
Train cost, Step 120, Cost 314.853333
Train cost, Step 130, Cost 402.010376
Train cost, Step 140, Cost 376.074463
Train cost, Step 150, Cost 421.073792
Train cost, Step 160, Cost 206.457184
Train cost, Step 170, Cost 345.537598
Train cost, Step 180, Cost 335.950012
Train cost, Step 190, Cost 272.736908
Train cost, Step 200, Cost 247.938675
Test cost, Step 200, Cost 118.644863
Train cost, Step 210, Cost 483.426575
Train cost, Step 220, Cost 269.894135
Train cost, Step 230, Cost 83.034637
Train cost, Step 240, Cost 240.733887
Train cost, Step 250, Cost 227.814202

```

```

https://starlight.nscc-gz.cn/?# nsccgz_duliang_1@PaddlePad... + [nsccgz_duliang_1@PaddlePad...]
不安全 | 461.proxy.nscc-gz.cn
应用 登录企业邮箱 k8s oauth2.0 go Free Learning - Free... e-Books - Free Books Slurm Workload Ma... Grafana - 物理节点

[nsccgz_duliang_1@PaddlePaddle ~]$ ls
2DT87RA04 buyaoshan-duzong paddlepaddle cntk-example-cntk      duiliang caffe-caffe duiliang_mxnet-mxnet   example          galaxy    mx
2DUITNGDP caffe-demo-cafe      cntk-example-ubuntu-cntk duiliang-cntk-cntk duiliang-test-tensorflow example-paddle-paddlepaddle kaldi    pac
[nsccgz_duliang_1@PaddlePaddle ~]$ cd example
[nsccgz_duliang_1@PaddlePaddle example]$ ls
caffe cntk galaxy kaldi mxnet paddlepaddle rstudio slurm spark tensorflow tensorflow-cluster torch
[nsccgz_duliang_1@PaddlePaddle example]$ cd paddlepaddle/
[nsccgz_duliang_1@PaddlePaddle paddlepaddle]$ ls
examples run_example.sh
[nsccgz_duliang_1@PaddlePaddle 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:recognize_digits
run recognize_digits
python examples/02.recognize_digits/train.py
In the fluid 1.0, the trainer and inferencer are moving to paddle.fluid.contrib
Pass 0, Batch 0, Cost 3.112504
Pass 100, Batch 0, Cost 0.220029

```

```

In [2]: 1 #!/usr/bin/python
2 import paddle.fluid as fluid
3 a = fluid.layers.data(name='a', shape=[1], dtype='float32')
4 b = fluid.layers.data(name='b', shape=[1], dtype='float32')
5 result = fluid.layers.elementwise_add(a, b)
6
7 #对X进行操作
8 cpc = fluid.core.CPUPlace() #如果想要使用GPU, 这里选择在GPU下运行
9 exe = fluid.Executor(cpc) #启动线程
10 exe.run(fluid.default_startup_program()) #初始化参数
11
12 #设置数据
13 import numpy
14 data_1 = input("x1")
15 data_2 = input("x2")
16 x = numpy.array([data_1])
17 y = numpy.array([data_2])
18
19 #执行计算
20 exe.run(exe, run=fluid.default_main_program())
21 fetch_list = [result.name]
22
23 #验证结果
24
25 print("%s + %s" % (data_1, data_2, result[0][0]))

```

```

In [32]: 1 from __future__ import print_function
2 import math
3 import numpy
4 import paddle
5 import paddle.fluid as fluid
6 import paddle.fluid.layers as layers
7 import paddle.fluid.net as nets
8
9
10 from paddle.fluid.contrib.trainer import *
11 from paddle.fluid.contrib.inferencer import *
12 except ImportError:
13     print()
14     print("In the version 1.0, the trainer and inferencer are moving to paddle.fluid.contrib",
15     "and paddle.fluid.contrib.inferencer respectively")
16
17
18 IS_MOBILE = True
19 IS_GPU = False
20 EPOCH_SIZE = 200
21
22

```

图 98 paddlepaddle 例程

以上演示了使用 PaddlePaddle 深度学习平台的使用流程和相关必要步骤，关于 paddlepaddle 框架的相关知识请移步至 [paddlepaddle 官网](#)

## 11.Torch 使用说明

torch 是一个深度学习的工具，它使用 lua 语言编写，集成了各种深度学习开发包，其支持 CPU, GPU 训练数据，具有使用简单，运行高效的特点，下面以 Torch 官方示例演示使用 Torch 框架的相关流程和步骤，[其中程序代码和数据源自 torch 官方 jit 仓库](#)。

### Step1 创建 Torch 应用



图 99 torch 应用表单

## Step2 进入作业详情，点击 web-terminal 进入应用

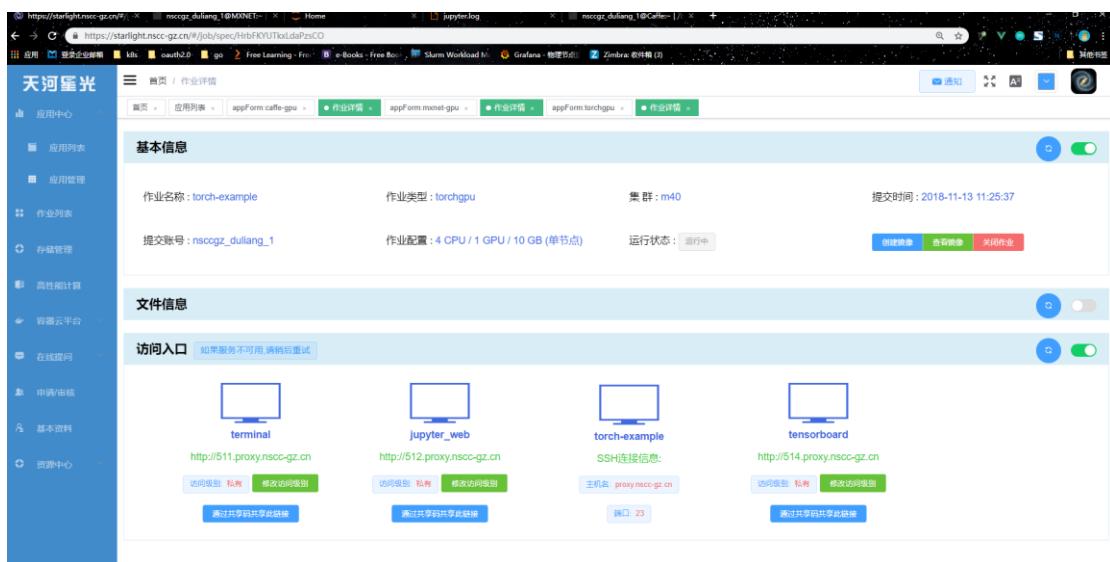


图 100 torch 作业详情

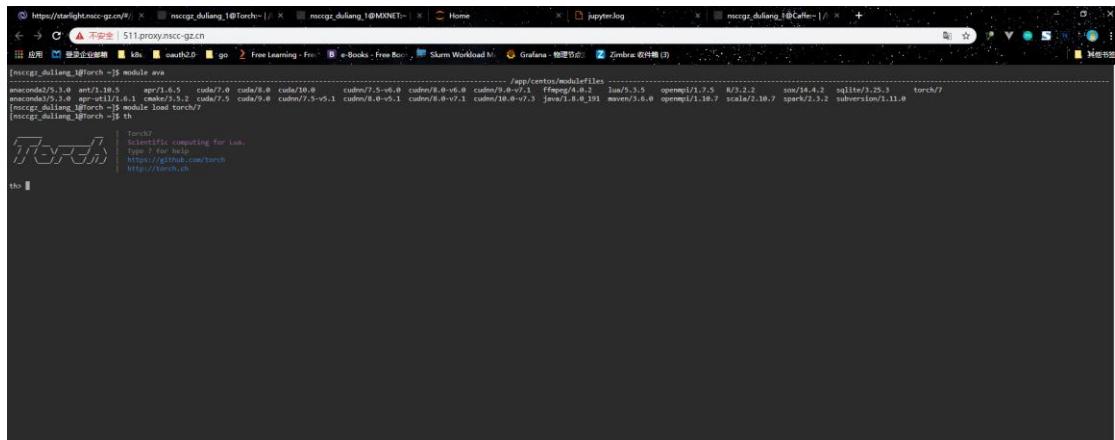


图 101 torch web-terminal

### Step3 采用 jupyter 进行远程代码开发，远程执行(pytorch)

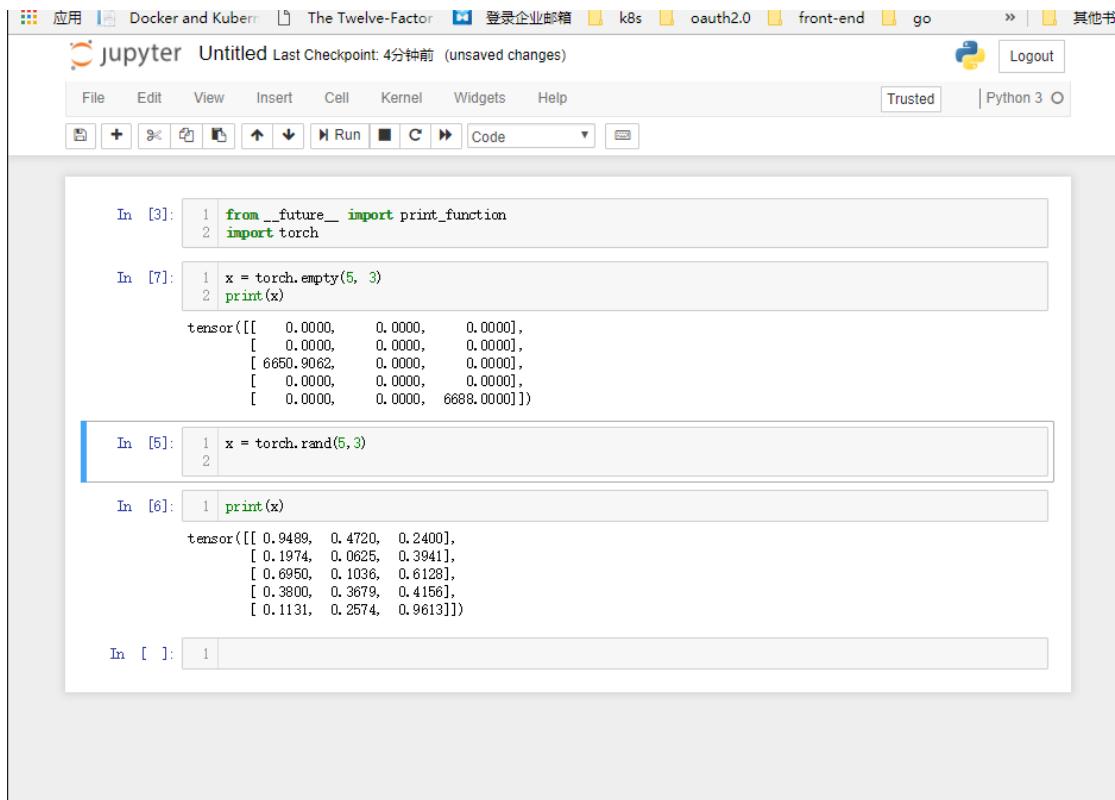


图 102 jupyter 开发 pytorch

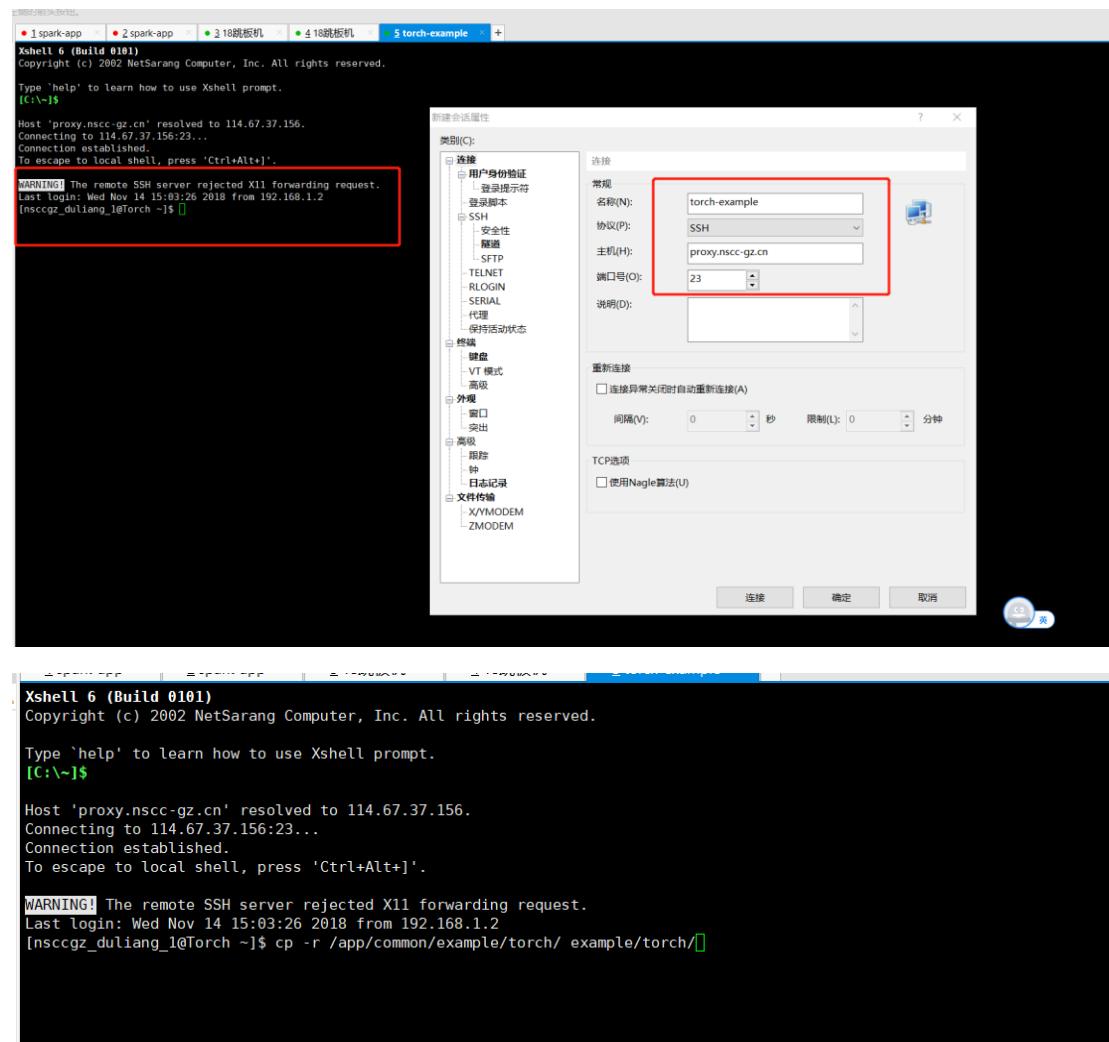
### Step4 pytorch 使用示例

示例程序位于/app/common/example/torch 目录下，示例程序脚本为 run\_example.sh

首先拷贝示例程序到用户目录。执行

```
cp -r /app/common/example/torch /home/your_home_dir/targetdir/
```

## 远程登录，参考第九章 1.2 节 ssh 工具远程登录作业



Ssh 终端运行示例程序

```

Connecting to 114.0.37.150.23...
Connection established.
To escape to local shell, press 'Ctrl+Alt+]'.

WARNING! The remote SSH server rejected X11 forwarding request.
Last login: Wed Nov 14 15:03:26 2018 from 192.168.1.2
[nsccgz_duliang_1@Torch ~]$ cd example/torch/
[nsccgz_duliang_1@Torch torch]$ pwd
/home/nsccgz_duliang_1/example/torch
[nsccgz_duliang_1@Torch torch]$ ls
run_example.sh torch-examples
[nsccgz_duliang_1@Torch torch]$ ./run_example.sh
---run torch examples---[demo can be found in https://github.com/torch/demos/]
you can choose example to run
for example Torch-Digit-Classifier,Torch-Logistic-Regres
---step1 set http_proxy---
step2 load modules
---begin to load modules for centos---
enter example name:Torch^H^H
exampleName can be 'Torch-Digit-Classifier,Torch-Logistic-Regres'
[nsccgz_duliang_1@Torch torch]$ ./run_example.sh
---run torch examples---[demo can be found in https://github.com/torch/demos/]
you can choose example to run
for example Torch-Digit-Classifier,Torch-Logistic-Regres
---step1 set http_proxy---
step2 load modules
---begin to load modules for centos---
enter example name:^H
exampleName can be 'Torch-Digit-Classifier,Torch-Logistic-Regres'
[nsccgz_duliang_1@Torch torch]$ ./run_example.sh
---run torch examples---[demo can be found in https://github.com/torch/demos/]
you can choose example to run
for example Torch-Digit-Classifier,Torch-Logistic-Regres
---step1 set http_proxy---
step2 load modules
---begin to load modules for centos---
enter example name:Torch-Logistic-Regres
run Torch Logistic-regression
cd torch-examples/logistic-regression; th example-logistic-regression.lua

=====
Constructing dataset

<csv>  parsing file: example-logistic-regression.csv
<csv>  tidyng up entries
<csv>  returning tidy table
  number of brands 3.000000
    min brand 1.000000
    max brand 3.000000
    min female 0.000000
    max female 1.000000
      min age 24.000000
      max age 38.000000

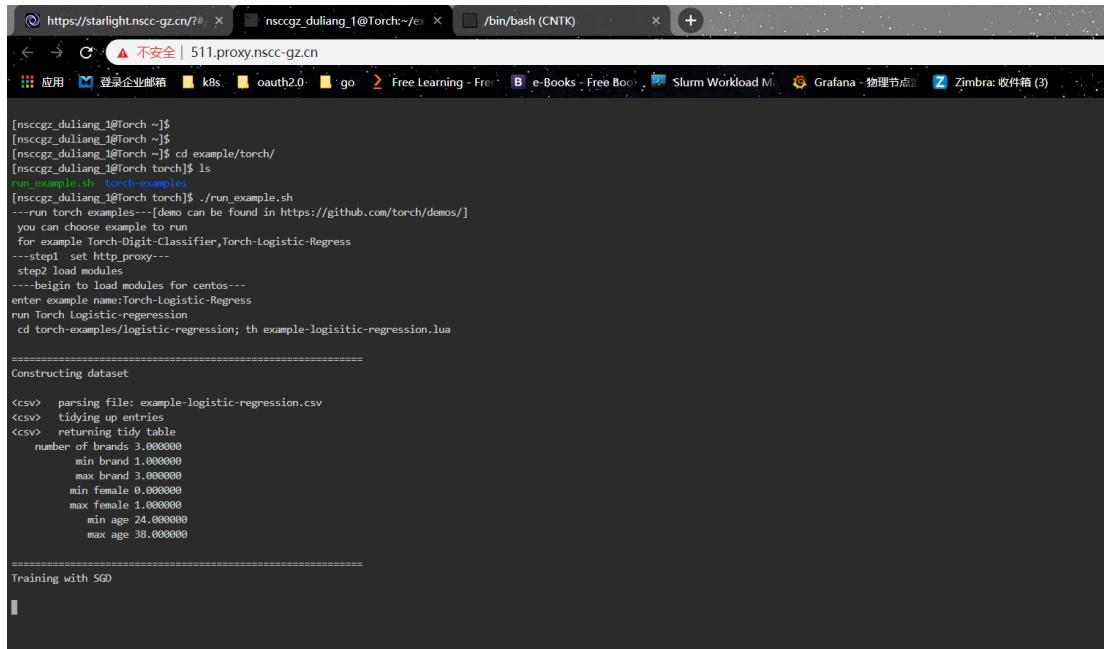
=====
Training with SGD

```

## 运行例子程序脚本

### 选择例子程序

## Web-terminal 终端运行示例程序



The screenshot shows a web browser window with multiple tabs open. The active tab is a terminal session titled 'nscggz\_duliang\_1@Torch:~\$'. The terminal window displays the same command-line session as the previous image, starting with 'run\_example.sh' and listing available examples like 'Torch-Digit-Classifier' and 'Torch-Logistic-Regres'. It then proceeds to construct a dataset from a CSV file and train a logistic regression model using Stochastic Gradient Descent (SGD). The interface includes standard browser controls at the top and a toolbar with various application icons below the address bar.

```

[nsccgz_duliang_1@Torch ~]$ 
[nsccgz_duliang_1@Torch ~]$ 
[nsccgz_duliang_1@Torch ~]$ cd example/torch/
[nsccgz_duliang_1@Torch torch]$ ls
run_example.sh torch-examples
[nsccgz_duliang_1@Torch torch]$ ./run_example.sh
---run torch examples---[demo can be found in https://github.com/torch/demos/]
you can choose example to run
for example Torch-Digit-Classifier,Torch-Logistic-Regres
---step1 set http_proxy---
step2 load modules
---begin to load modules for centos---
enter example name:Torch^H^H
exampleName can be 'Torch-Digit-Classifier,Torch-Logistic-Regres'
[nsccgz_duliang_1@Torch torch]$ ./run_example.sh
---run torch examples---[demo can be found in https://github.com/torch/demos/]
you can choose example to run
for example Torch-Digit-Classifier,Torch-Logistic-Regres
---step1 set http_proxy---
step2 load modules
---begin to load modules for centos---
enter example name:^H
exampleName can be 'Torch-Digit-Classifier,Torch-Logistic-Regres'
[nsccgz_duliang_1@Torch torch]$ ./run_example.sh
---run torch examples---[demo can be found in https://github.com/torch/demos/]
you can choose example to run
for example Torch-Digit-Classifier,Torch-Logistic-Regres
---step1 set http_proxy---
step2 load modules
---begin to load modules for centos---
enter example name:Torch-Logistic-Regres
run Torch Logistic-regression
cd torch-examples/logistic-regression; th example-logistic-regression.lua

=====
Constructing dataset

<csv>  parsing file: example-logistic-regression.csv
<csv>  tidyng up entries
<csv>  returning tidy table
  number of brands 3.000000
    min brand 1.000000
    max brand 3.000000
    min female 0.000000
    max female 1.000000
      min age 24.000000
      max age 38.000000

=====
Training with SGD

```

以上演示了使用 Torch 深度学习平台的使用流程和相关必要步骤，关于 Torch 框架的相关知识请移步至 [Torch 官网](#)。