

工作站为中南大学计算机学院使用。此工作站仅用于深度学习研究，请跑传统普通程序的同学老师不要占用该工作站的CPU资源。

依通知，本工作站为计算机学科建设内容，工作站仅限计算机系老师同学使用

请务必认真阅读管理系统使用说明，大多问题都会在以下说明中解决

管理系统

管理系统是通过Web页面管理使用工作站的账号和连接工作站途径的系统。

注册

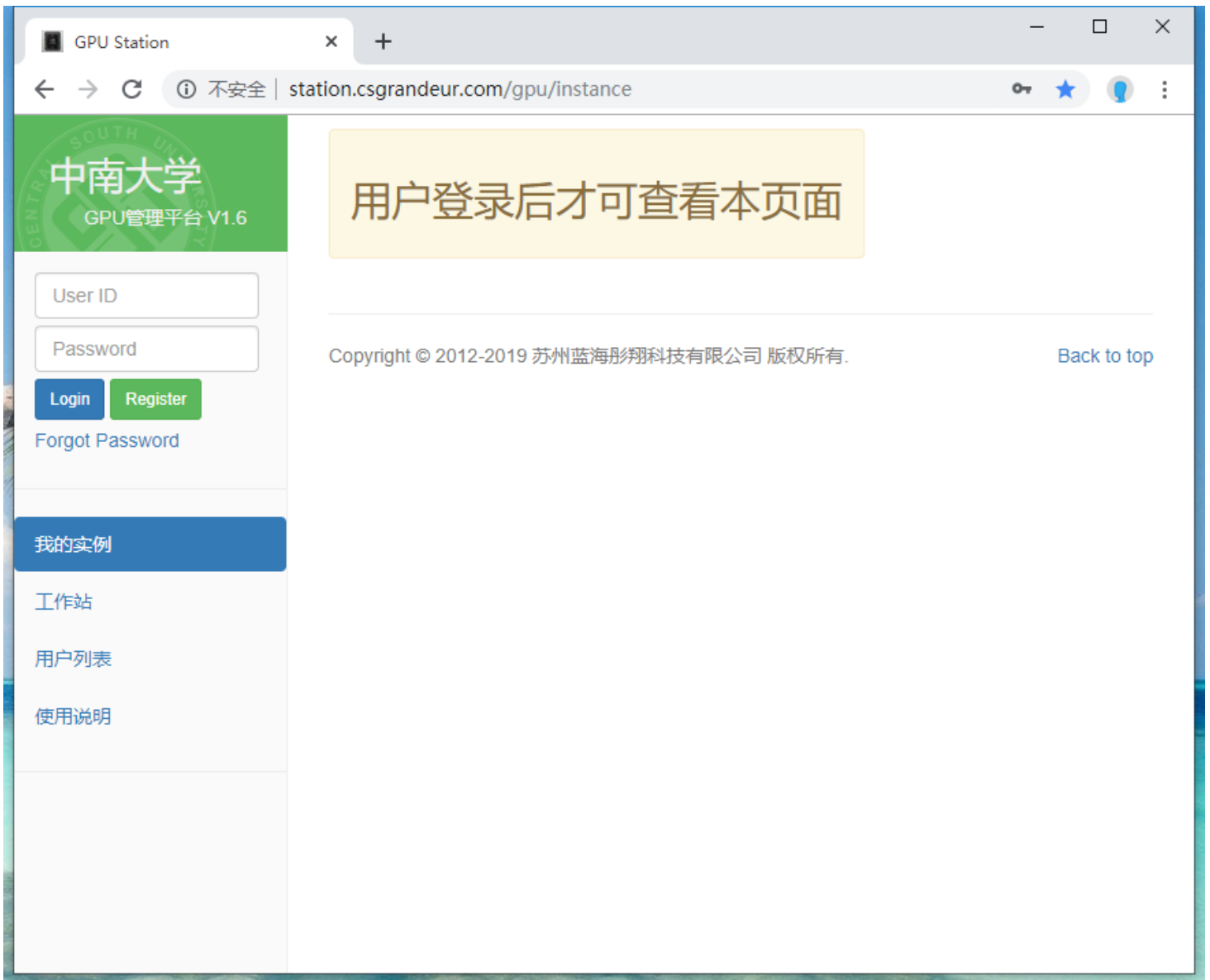
使用工作站需要先在管理系统中注册一个账号。

邀请码

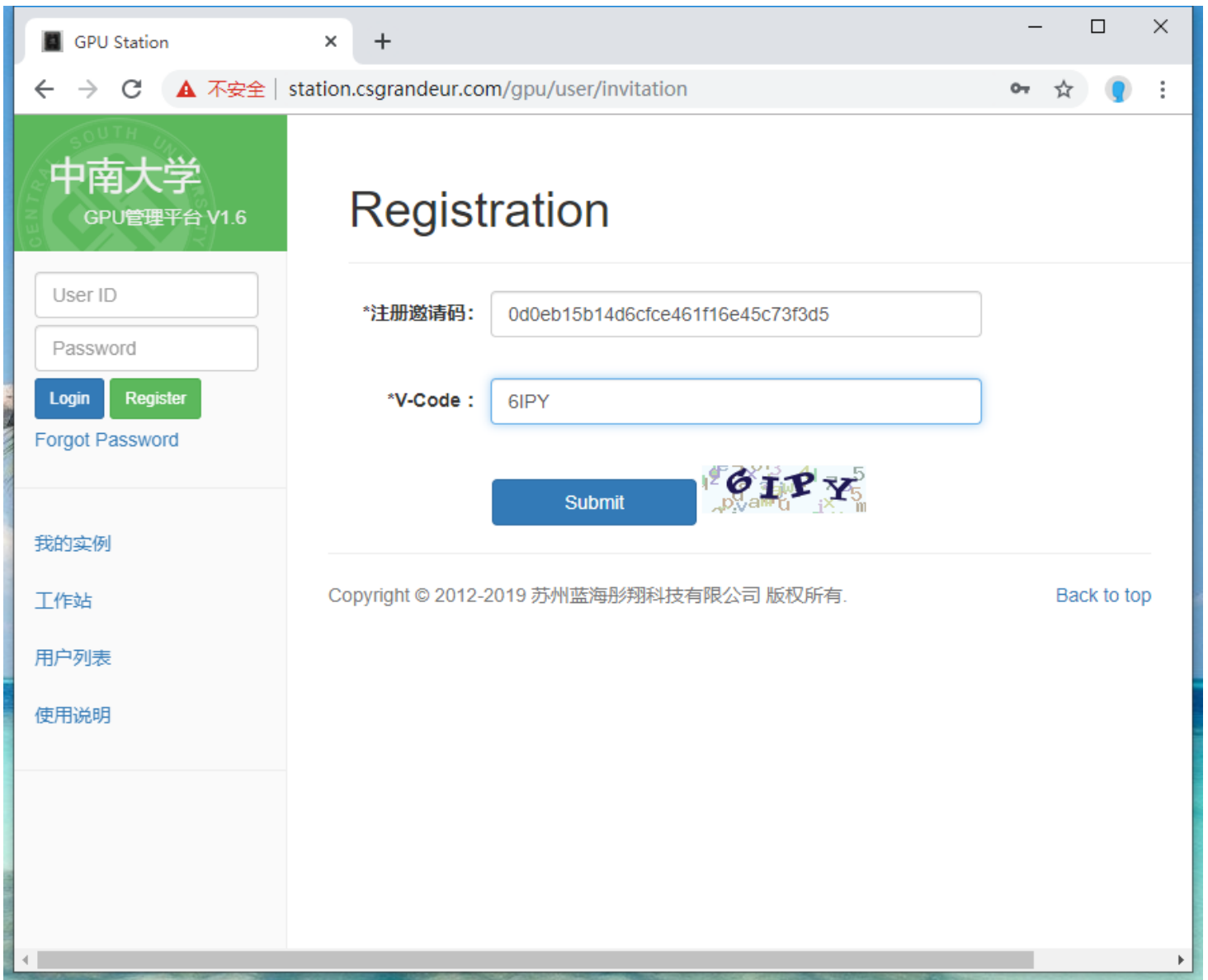
为控制使用工作站成员的资格，系统以邀请码形式注册，请需要使用工作站的老师同学加入信息院高性能工作站使用QQ群 346619322，在验证身份后由管理员给予邀请码，进入系统注册。

注册账号

- 工作站使用系统链接为<http://station.csgrandeur.com>，点击Register进行注册：



- 输入管理员给予的邀请码和页面的验证码，确认：



- 如果邀请码正确，则会自动跳转至账号注册页面，填写自己的学号姓名密码等相关信息，请务必正确填写邮箱，以便于忘记密码时找回。

Registration

*学号 :	<input type="text" value="1"/>
*姓名 :	<input type="text"/>
导师 :	<input type="text"/>
*E-Mail :	<input type="text" value="c"/>
*密码 :	<input type="password" value="....."/>
*确认密码 :	<input type="password" value="....."/>
*V-Code :	<input type="text" value="tagp"/>



Copyright © 2016-2017 CSGrandeur. All Rights Reserved.
Designer & Developer : CSGrandeur.

查看工作站状态

由于工作站是共享的，随时都可能有其他人在运行程序，在进行自己的操作前，请先看工作站的 GPU、CPU、Memory 的使用情况，如果有他人在使用，请等待或在交流群中沟通，盲目运行程序可能导致程序冲突，会中断他人程序或得到非正常的运行结果。

工作站状态页面未来会逐步进行更新，使展示的信息更加全面、友好。

The screenshot displays the 'RTX2080_00 状态' page in a browser. The top navigation bar includes the university logo and 'GPU管理平台 V1.6'. The main content area is divided into two tabs: '工作站状态' (Workstation Status) and '工作站任务队列' (Workstation Task Queue). The '工作站状态' tab shows a table with columns for ID, Name, IP, Start Port, End Port, Permissions, Remarks, Responsible Person, and Supported Container Types. A single instance 'RTX2080_00' is listed with IP 122.207.82.55 and ports 10100-12999.

Below the table is a detailed GPU information section for 'Wed Jan 8 14:05:08 2020'. It lists six GeForce RTX 2080 series GPUs with their respective IDs, names, persistence modes, bus IDs, display adapters, fan speeds, temperatures, power usage, and memory usage. All GPUs are currently at 22% power usage and 0% GPU utilization.

The right sidebar contains a '添加实例' (Add Instance) button and a '帮助' (Help) button. Below these are two sections titled '该工作站我的' (This workstation my), each containing a table with fields for Instance ID, Instance Name, Open Port, Creation Time, and Available Operations.

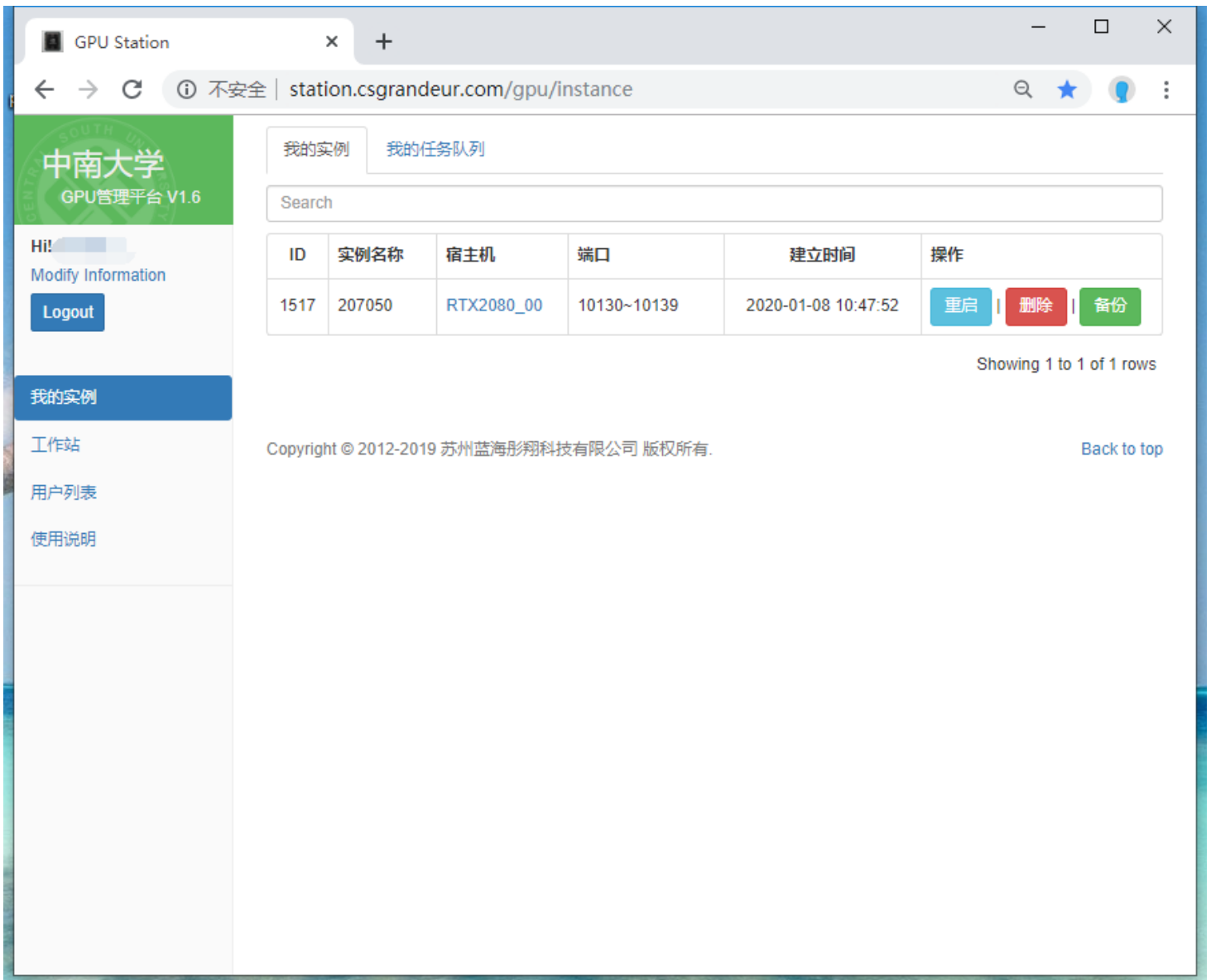
管理自己的实例

为保证每个人使用系统时的自主性（完整系统权限）、独立性（互相之间的文件系统不干扰），每个人会在主机中开启单独的容器实例，通过各自的实例使用计算资源。

下文中“容器”和“实例”可视为等价的指代。

我的实例

- 我的实例页面会显示在各工作站上已拥有的实例，具体的信息解释将在下文连接工作站部分进行。



开启新实例

- 在工作站状态页面，点击添加实例，选择开启的实例类型。

RTX2080_01 状态

station.csgrandeur.com/gpu/station/station?station_id=23

中南大学 GPU管理平台 V1.6

Hi! [用户名]
[Modify Information](#)
[Logout](#)

我的实例
工作站
 用户列表
 使用说明

工作站状态 | 工作站任务队列

ID	名称	IP	起始端口	结束端口	权限	备注	负责人	支持容器类型
23	RTX2080_01	122.207.82.55	13100	15999	公有	中南大学计算机学院	梁毅雄	cuda8,cuda9,caffe,cuda10

```

Wed Jan 8 14:11:41 2020
-----+-----
| NVIDIA-SMI 440.33.01    Driver Version: 440.33.01    CUDA Version: 10.2    |
|-----+-----|
| GPU Name      Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|-----+-----|
|  0  GeForce RTX 208...  Off  | 00000000:1B:00:0 Off |             N/A |
| 22%  41C   P0   62W / 250W |  0MiB / 11019MiB |    0%    Default |
|-----+-----|
|  1  GeForce RTX 208...  Off  | 00000000:1C:00:0 Off |             N/A |
| 22%  35C   P0   66W / 250W |  0MiB / 11019MiB |    0%    Default |
|-----+-----|
|  2  GeForce RTX 208...  Off  | 00000000:1D:00:0 Off |             N/A |
| 22%  40C   P0   61W / 250W |  0MiB / 11019MiB |    0%    Default |
|-----+-----|
|  3  GeForce RTX 208...  Off  | 00000000:1E:00:0 Off |             N/A |
| 22%  40C   P0   65W / 250W |  0MiB / 11019MiB |    0%    Default |
|-----+-----|
|  4  GeForce RTX 208...  Off  | 00000000:3D:00:0 Off |             N/A |
| 22%  36C   P0   48W / 250W |  0MiB / 11019MiB |    0%    Default |
|-----+-----|
|  5  GeForce RTX 208...  Off  | 00000000:3F:00:0 Off |             N/A |
| 22%  38C   P0   65W / 250W |  0MiB / 11019MiB |    0%    Default |
|-----+-----|
|  6  GeForce RTX 208...  Off  | 00000000:40:00:0 Off |             N/A |
|-----+-----|

```

[添加实例](#) [帮助](#)

该工作站我的

实例ID	
实例名称	
开放端口	
建立时间	
可用操作	

该工作站我的

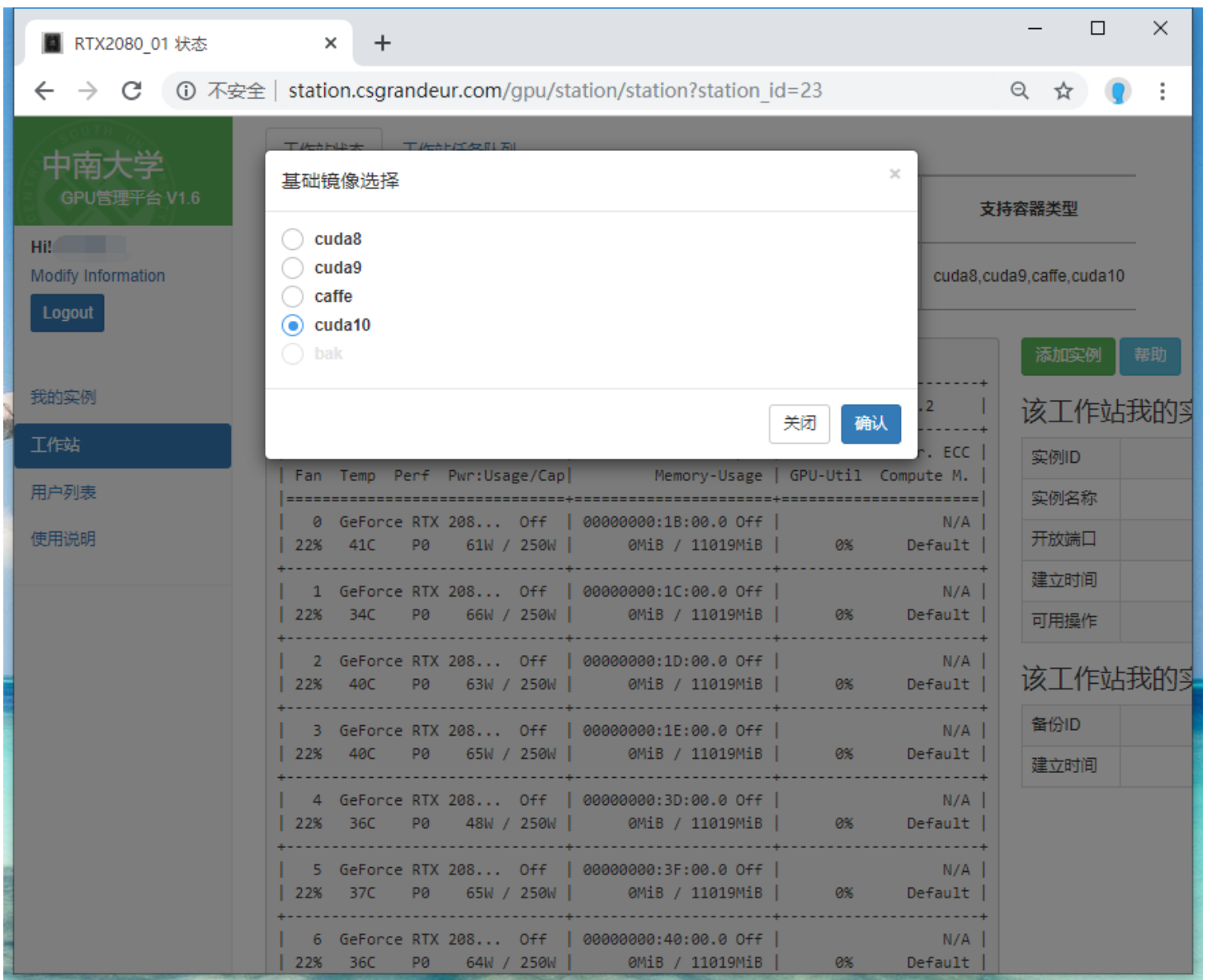
备份ID	
建立时间	

系统初期只提供已安装 **cuda** 的实例，未来或可增加提供已经安装好 **Tensorflow**、**Caffe**等深度学习库的实例。

“从备份新建”是指之前已经在相应的工作站配置过自己的实例，并对实例进行过备份的用户，可以通过此选项新建自己配置过的实例。

由于系统资源有限，每个人每台工作站只可开启一个实例，如果要重新开实例，请先删除当前实例，再开启新实例。

系统初期每个人每台工作站只可做一份实例备份，新的备份将覆盖旧的备份。



任务队列

- 当执行实例的 新建、重启、删除、备份 等操作时，需要等待工作站主机应答。

系统会有一个任务队列，当执行某项操作时，可在任务队列查看是否执行完毕，以开始自己的下一步操作。如果执行过程存在问题，系统会在“执行结果”列给予提示。

如果一个实例执行 2 分钟以上依然没有结束，“重试”按钮会亮起，可以等待也可以点击该按钮重试该操作。

如果有非常规问题出现，可在交流群反馈咨询。

GPU host Station

station.csgrandeur.com/gpu/station/task?station_id=22

中南大学 GPU管理平台 V1.6

Hi! [User Name]

Modify Information

Logout

我的实例

工作站

用户列表

使用说明

工作站状态 | 工作站任务队列

ID	名称	IP	起始端口	结束端口	权限	备注	负责人	支持容器类型
22	RTX2080_00	122.207.82.55	10100	12999	公有	中南大学计算机学院	梁毅雄	cuda8,cuda9,caffe,cuda10

Search

ID	操作名称	用户	实例ID	操作参数	任务状态	执行结果	提交时间	操作
4526	添加实例	[User]	-	cuda9	执行完毕	OK	2020-01-08 13:51:18	-
4525	添加实例	[User]	-	cuda9	执行完毕	OK	2020-01-08 13:25:08	-
4523	添加实例	[User]	-	cuda10	执行完毕	OK	2020-01-08 10:47:30	-
4521	添加实例	[User]	-	cuda10	执行完毕	OK	2020-01-08 10:21:40	-
4520	删除实例	[User]	1512	-	执行完毕	OK	2020-01-08 10:16:47	-
4516	添加实例	[User]	-	cuda10	执行完毕	OK	2020-01-08 10:16:08	-

Showing 1 to 6 of 6 rows

Copyright © 2012-2019 苏州蓝海彤翔科技有限公司 版权所有. [Back to top](#)

管理后台

管理后台与普通用户无关，或会招募 2~3 名特定功能（如邀请码分发、密码协助修改等）的管理员，协助系统维护人员为大家服务。详细功能会单独介绍，不在此说明书中进行。

连接并使用计算资源

系统环境

系统统一为 Ubuntu Server 16.04 x64，默认的cuda为最新版，但用户可自行修改系统内的软件环境。

Ubuntu Server

针对不了解 Server 系统的老师与同学简单介绍一下。

Ubuntu（友帮拓、优般图、乌班图）是一个开源GNU/Linux操作系统，Ubuntu 是基于Debian GNU/Linux，支持x86、amd64（即x64）和ppc架构，由全球化的专业开发团队（Canonical Ltd）打造的。

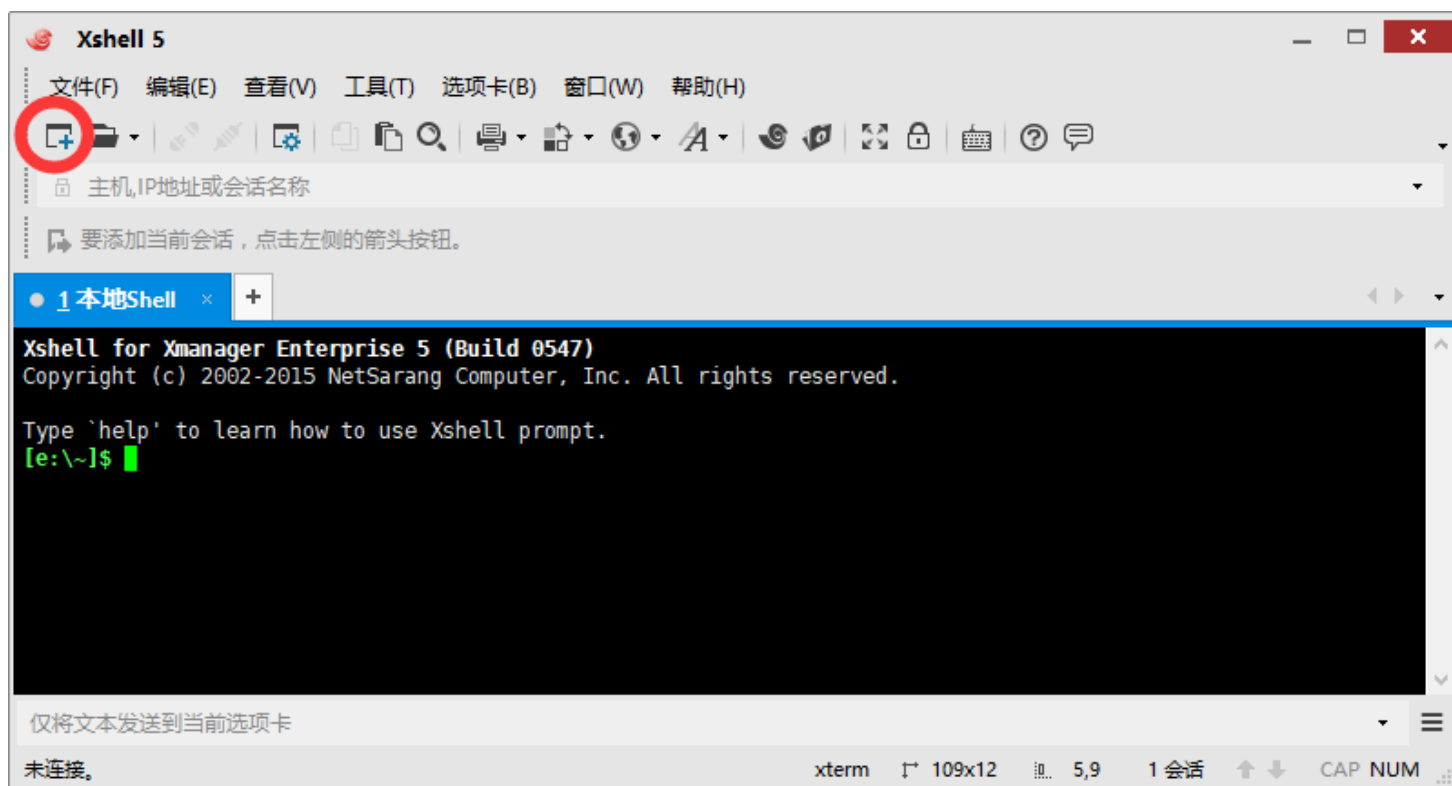
Ubuntu Server 系统没有图形界面，由SSH连接和命令行进行操作，是进行GPU计算的最佳系统环境，目前大多科研工作是基于 Ubuntu Server 系统进行算法原型测试的。

命令行系统并不可怕，我们可借助许多工具来实现我们的目的，比如文件传递、代码复制、远程调试、通过SSH隧道的X11连接在本地显示软件的图形界面等，这些方法与工具都可以通过搜索引擎查找和学习。

SSH 连接

SSH连接Linux系统的工具有多种，这里推荐一个很好的软件套装 XManager，为方便下载，提供百度网盘连接：<https://pan.baidu.com/s/1mhLinbi>。交流群文件中也提供了下载。

- 打开XShell并新建连接



- 查看前面管理系统中新建的实例信息中的端口号

The screenshot shows the GPU Station web interface. The browser address bar displays 'station.csgrandeur.com/gpu/instance'. The page header includes '中南大学 GPU管理平台 V1.6'. The main content area features a table with the following data:

ID	实例名称	宿主机	端口	建立时间	操作
1517	207050	RTX2080_00	30~39	2020-01-08 10:47:52	重启 删除 备份

The '端口' column value '30~39' is circled in red. Below the table, it says 'Showing 1 to 1 of 1 rows'. The footer contains 'Copyright © 2012-2019 苏州蓝海彤翔科技有限公司 版权所有.' and a 'Back to top' link.

- 填写连接信息

每个实例会开放 10 个自由使用的端口，使用端口能做些什么事，可以利用搜索引擎进行学习，也欢迎在交流群中交流。

而提供的 SSH 连接则占用第一个端口。上图中提供的端口号为 21010~21019，那么我们SSH连接的端口号即 21010。

服务器所在机房的IP地址会在交流群和管理系统中提供。

实例直接使用 root 账号连接，默认密码是 123456 即拥有实例内几乎全部权限。

类别(C):

- [-] 连接
 - [-] 用户身份验证
 - 登录提示符
 - 登录脚本
 - [-] SSH
 - 安全性
 - 隧道
 - SFTP
 - TELNET
 - RLOGIN
 - SERIAL
 - 代理
 - 保持活动状态
- [-] 终端
 - 键盘
 - VT 模式
 - 高级
- [-] 外观
 - 边距
- [-] 高级
 - 跟踪
 - 日志记录
- ZMODEM

连接

常规

名称(N):

协议(P):

主机(H):

端口号(O):

说明(D):



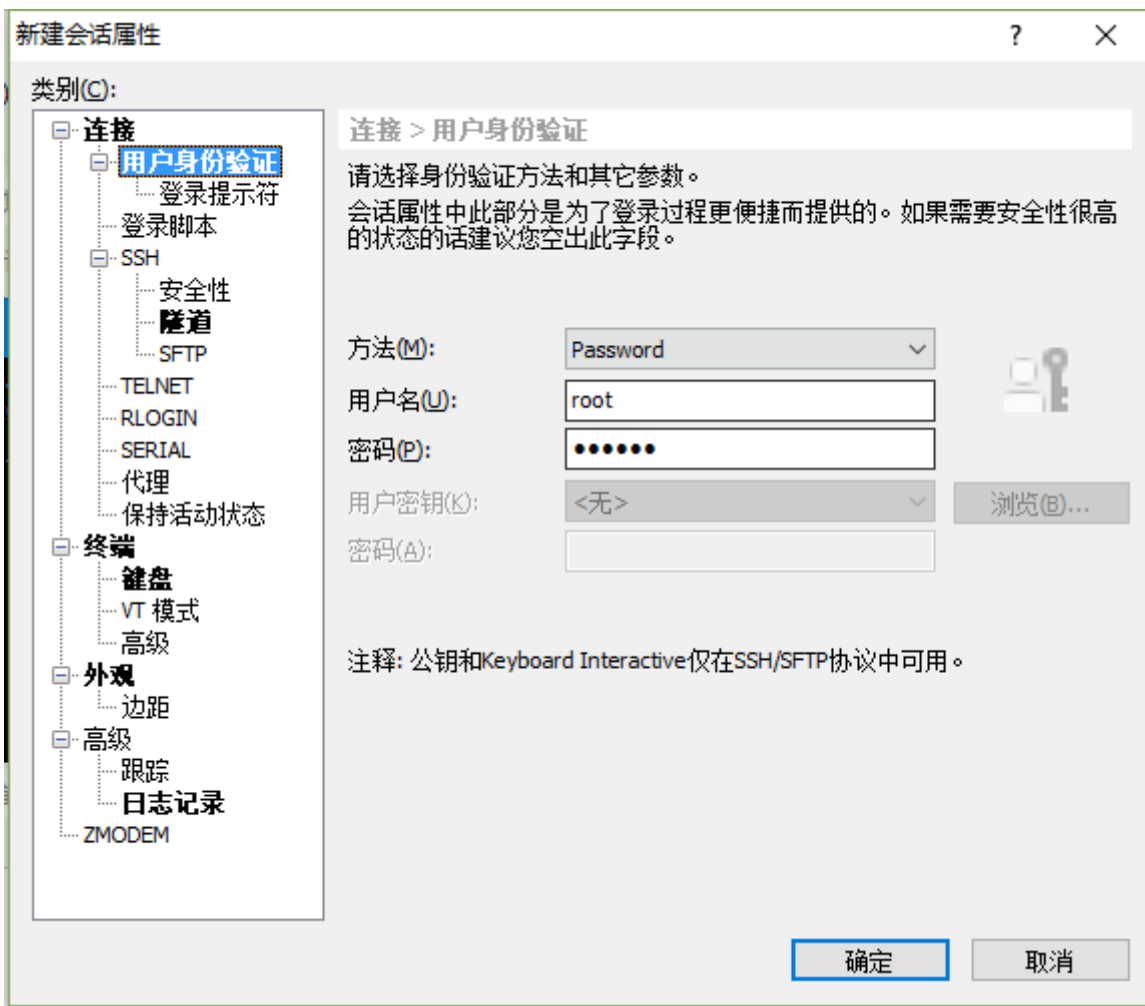
重新连接

连接异常关闭时自动重新连接(A)

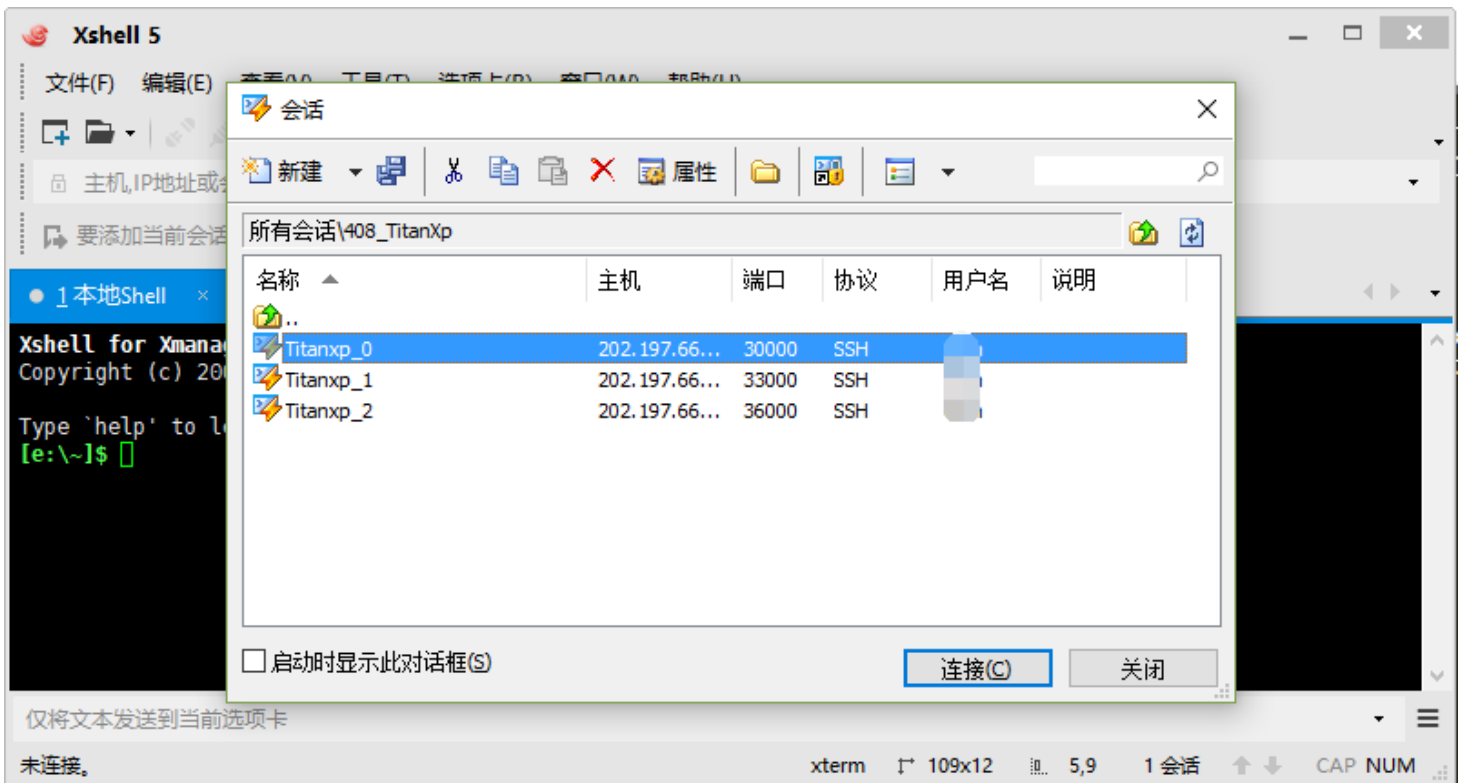
间隔(V): 秒 限制(L): 分钟

确定

取消



- 把连接信息保存，以后就可以直接点选连接了



使用

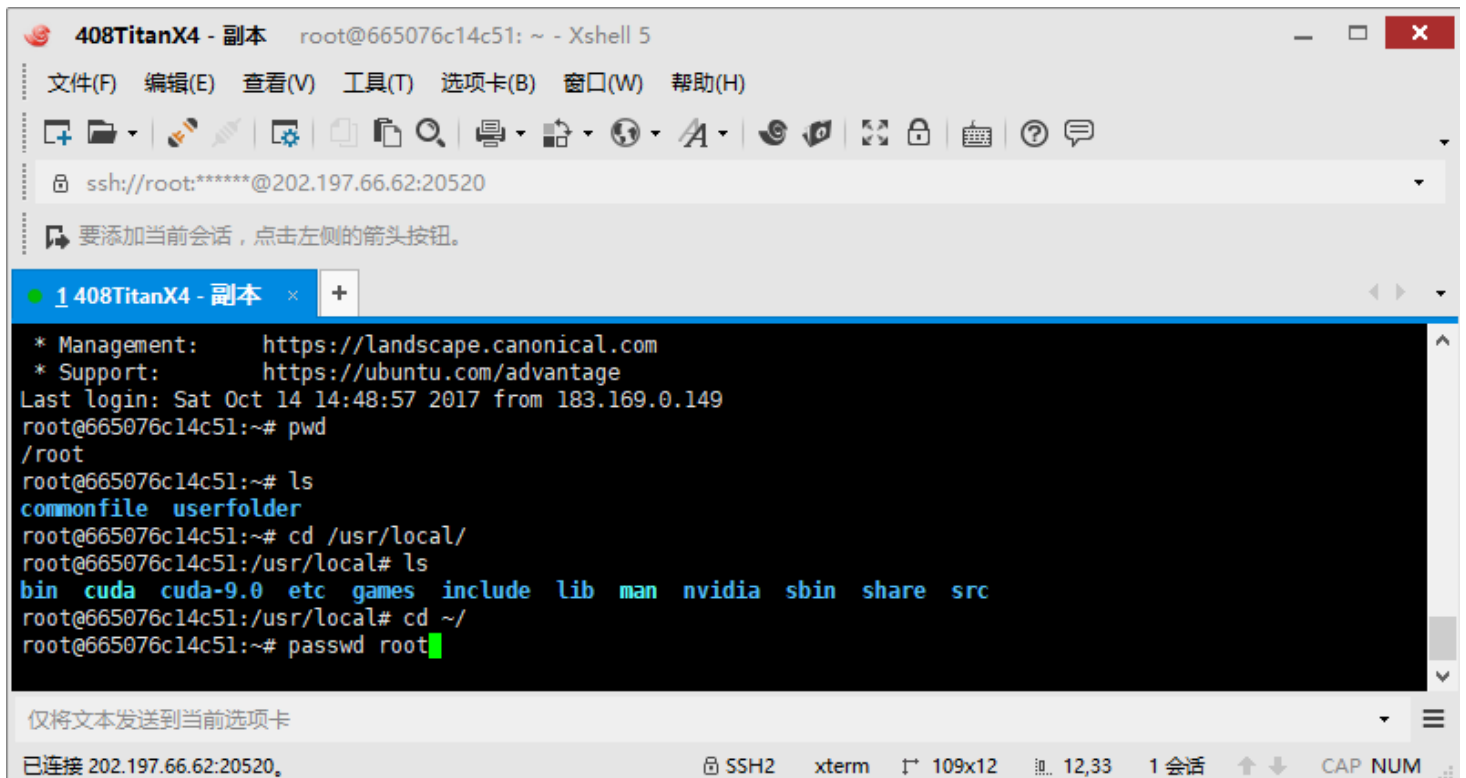
成功建立 SSH 连接之后，就可以开始命令行了，如图进行了几个简单的操作，可以看当前的目录、目录下的内容。

本身就是root账户，所以

不需要sudo指令！不需要sudo指令！不需要sudo指令！

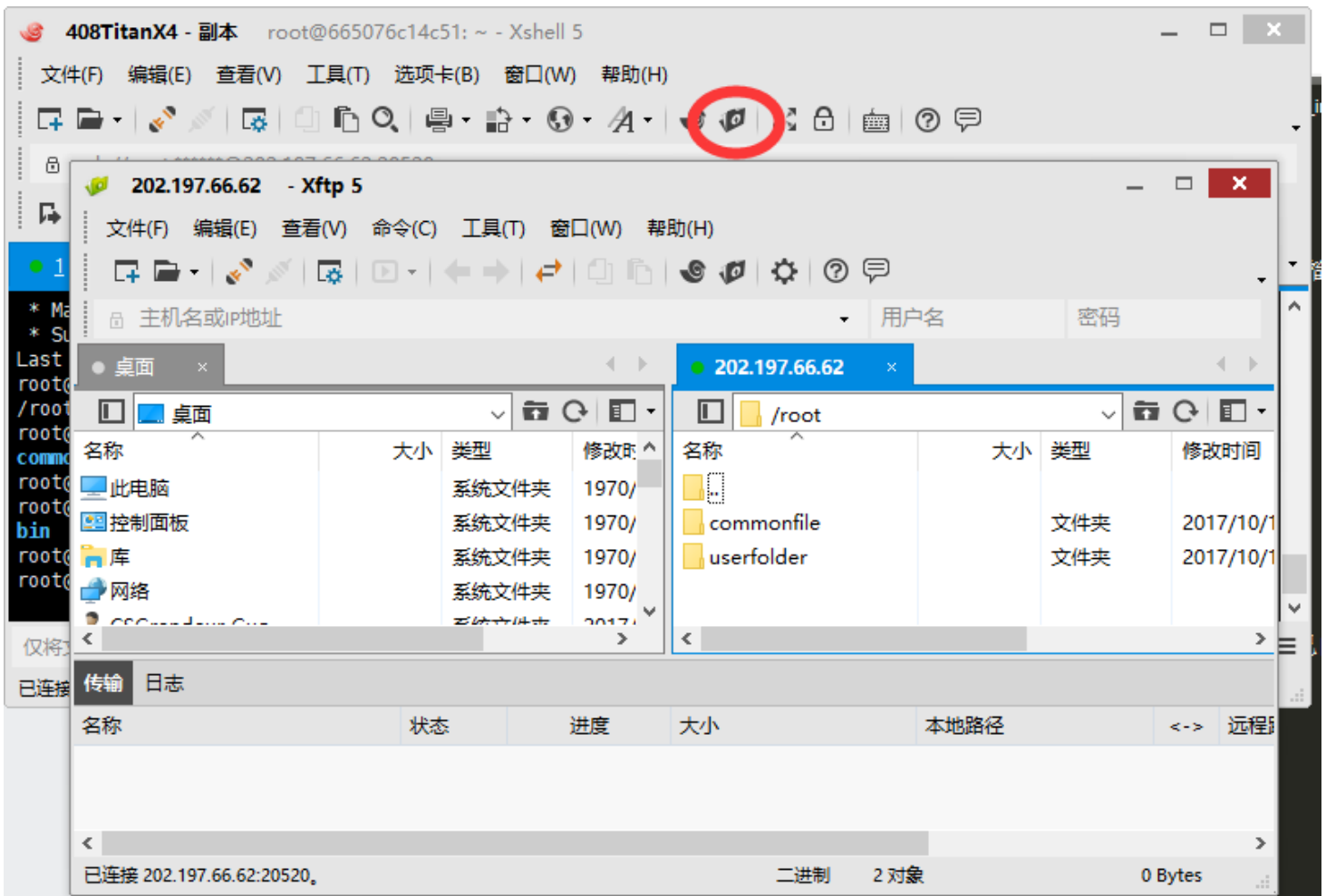
重要的事情说三遍。

建议进入自己的实例之后，修改root密码，避免其他人误登录自己的实例。



```
408TitanX4 - 副本  root@665076c14c51: ~ - Xshell 5
文件(F) 编辑(E) 查看(V) 工具(T) 选项卡(B) 窗口(W) 帮助(H)
ssh://root:*****@202.197.66.62:20520
要添加当前会话，点击左侧的箭头按钮。
1 408TitanX4 - 副本 x +
* Management:  https://landscape.canonical.com
* Support:     https://ubuntu.com/advantage
Last login: Sat Oct 14 14:48:57 2017 from 183.169.0.149
root@665076c14c51:~# pwd
/root
root@665076c14c51:~# ls
commonfile userfolder
root@665076c14c51:~# cd /usr/local/
root@665076c14c51:/usr/local# ls
bin  cuda  cuda-9.0  etc  games  include  lib  man  nvidia  sbin  share  src
root@665076c14c51:/usr/local# cd ~/
root@665076c14c51:~# passwd root
```

可以通过 SFTP 进行文件传输



到此步，就可以开始自己的环境、代码部署与实验了。

望大家加强交流，互帮互助，不要因为命令行系统的一点门槛影响了科研进度。

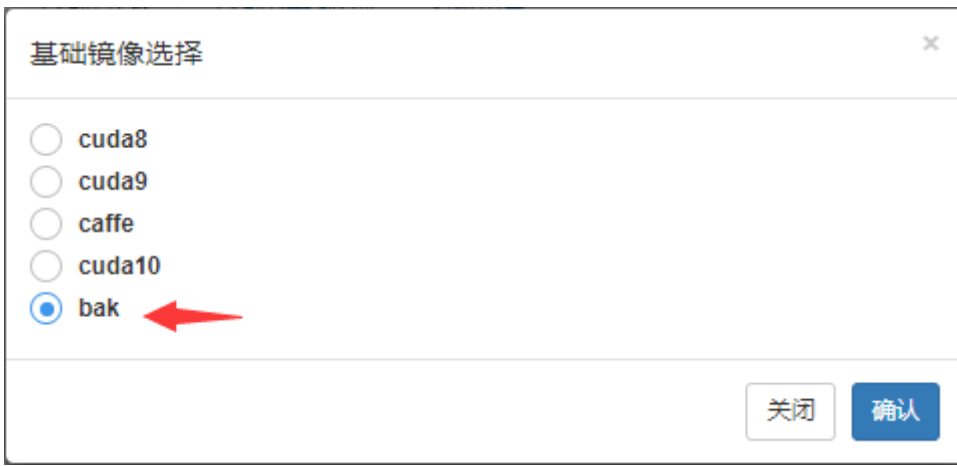
操作

- 重启：执行实例的重启，完成环境变量的生效等需求。
- 删除：当前每个用户在每台主机上只能开启一个实例，如果想重新开实例，则先删除旧实例。
- 备份：将当前开启的实例备份为一个镜像，以备份自己的环境配置

由于系统资源有限，每个用户在每个宿主机只能进行一份备份，新的备份会替换旧的备份。

由于只有一份备份，所以不再在系统中显示备份信息，在任务队列确认备份成功即可。

备份的使用方法：当任务队列中显示备份任务执行完成之后，如果要使用备份的镜像，则删除旧实例，然后添加实例，在对话框选择“从备份新建”。



注意事项

此部分非常重要，望周知

**** 所有个人文件请放在/root/userfolder目录内，否则不保证文件数据安全 ****

连入SSH后，会在/root目录中，目录下有commonfile和userfolder两个文件夹，commonfile是所有用户共享的文件夹，用于共享文件中转，userfolder是私人文件夹，只有这两个目录下的文件会保存在映射目录中，即使删除、重开实例，这些文件会依然保留。

一定不要把大型文件直接放在 /root 根目录下，这样会占用大量系统资源，此情况下管理员可能会酌情删除过大的实例。

**** 请养成本地备份数据的习惯，并适时清理自己在服务器实例中的数据 ****

服务器仅为提供高性能GPU计算，硬盘虽然较大，但用户人数众多，无法承担日积月累的数据堆积，请大家在完成自己的实验后，不要把老旧的数据继续留在服务器上。

**** 在进行自己的实验之前，先查看服务器运行状态 ****

如果有其他老师同学在运行实验（主要看GPU的显存占用情况），可在交流群中协商，不要贸然进行自己的实验，否则可能因显存冲突中断彼此的程序。

**** 按需设置自己的程序 ****

部分深度学习框架会自动占用所有可占用的显存，但一些学习任务可能仅需要一颗GPU就足够，望大家在程序中配置好所需占用的GPU，而不要盲目占用所有资源。

**** 禁止长期占用CPU运行纯CPU程序 ****

GPU工作站用以进行以深度学习为主的GPU计算，长期满载占用CPU资源会严重影响其他用户的使用，且该工作站硬件并未针对CPU计算优化，CPU程序在此工作站上并不一定能达到理想速度。

附录

Caffe 配置

这里提供一个Caffe配置的方法，但鼓励大家去各深度学习库的官方网站查看原文档，在Stack Overflow等论坛排查自己遇到的问题，多利用搜索引擎，培养良好的科研素养。

基本依赖项

```
apt-get install libprotobuf-dev libleveldb-dev \  
libsndpp-dev libopencv-dev libhdf5-serial-dev protobuf-compiler \  
apt-get install --no-install-recommends libboost-all-dev \  
  
apt-get install libatlas-base-dev
```

其他依赖项

```
apt-get install libgflags-dev libgoogle-glog-dev liblmbd-dev
```

OpenCV

```
#[compiler] \  
apt-get install build-essential \  
#[required] \  
apt-get install cmake git libgtk2.0-dev \  
pkg-config libavcodec-dev libavformat-dev \  
libswscale-dev \  
#[optional] \  
apt-get install python-dev python-numpy \  
libtbb2 libtbb-dev libjpeg-dev libpng-dev \  
libtiff-dev libjasper-dev libdc1394-22-dev
```

可以下载opencv的包解压，也可以用最新代码：

```
git clone https://github.com/Itseez/opencv.git
cd ~/opencv
mkdir build
cd build

cmake -D CMAKE_BUILD_TYPE=Release -D \
CMAKE_INSTALL_PREFIX=/usr/local
# make -j 后面的数字是并行个数，cpu厉害就设大点。一般是用4
make -j7
make install
```

CuDNN

解压cudnn的包（官网得申请，用网盘搜索能找到最新的），有include和lib64，里面文件复制到对应/usr/local/cuda/对应文件夹里

```
#进到对应文件夹
cp cudnn.h /usr/local/cuda/include/
#进到对应文件夹
cp lib* /usr/local/cuda/lib64/
#可能要再进行一次 ldconfig -v
```

不知道这里会不会有文件权限问题，暴力搞一下（这条可先不用）

```
chmod 777 -R /usr/local/cuda/lib64
```

Caffe

```
git clone https://github.com/BVLC/caffe.git
cp Makefile.config.example Makefile.config
```

修改Makefile.config，去掉cudnn的注释，其他的在当前应用场景可不变。

```
make all
make test
make runtest
```