

浪潮人工智能深度学习管理系统

AISStation

用户使用手册

浪潮（北京）电子信息产业有限公司

尊敬的用户：

衷心感谢您选用了浪潮人工智能深度学习管理系统AISStation！

本手册介绍了 AISStation 软件实现功能和使用方法，可使使用者更好地了解本软件的各种特性，充分的发挥本软件的作用。

浪潮（北京）电子信息产业有限公司拥有本手册的版权。

本手册中的内容如有变动恕不另行通知。

如果您对本手册有疑问或建议，请向浪潮（北京）电子信息产业有限公司垂询。

浪潮（北京）电子信息产业有限公司

二零一七年七月

技术服务电话：	800-860-0011
网 址：	http://www.inspur.com
地 址：	北京市海淀区上地信息路 2 号 创业园 C 栋 1 层
邮 编：	100085
E-mail	hpc@inspur.com

传 真：	86-10-62988886-6572
网 址：	http://www.inspur.com
地 址：	北京市海淀区上地 信息路 2 号创业园 C 栋 1 层

目录

第一章 整体介绍	5
1.1 软件介绍	5
1.2 系统要求	6
第二章 登录界面	6
第三章 数据及应用程序上传	7
3.1 上传数据	7
3.2 数据解压操作	9
第四章 提交 Tensorflow 训练任务与管理	9
4.1 任务提交	9
1)通过命令提交训练任务	9
2)通过启动脚本提交训练任务	10
3)通过应用参数提交训练任务	11
4.2 训练过程管理	12
1)查看训练任务运行情况	12
2)查看训练进度及结果	13
第五章提交 caffe 任务与管理	14
5.1 任务提交	14
1) 通过命令提交训练任务	14
2)通过启动脚本提交训练任务	15
3)通过应用参数提交训练任务	15
4.2 训练过程管理	16
1)查看训练任务运行情况	16
2)查看训练进度及结果	17
第六章 提交 mxnet 任务与管理	18
6.1 任务提交	18
1)通过命令提交训练任务	18
2)通过启动脚本提交训练任务	18
3)设置应用参数提交训练任务	19
6.2 训练过程管理	20
1)查看训练任务节点运行情况	20

2)查看训练进度及结果.....	21
第七章 完成作业管理.....	22
1)完成任务查看.....	22
2)查看任务训练信息.....	24
第八章 集群监控管理.....	24
1)GPU 集群监控.....	24
2)GPU 单卡实时运行监控.....	25
第九章 GPU 资源申请.....	27
1)容器资源申请.....	27
2)使用容器.....	28
第十章 应用模板.....	30
10.1 新建作业模板.....	30
10.2 保存作业模板.....	36
10.3 复制应用模板.....	36
10.4 删除应用模板.....	37
10.5 编辑应用模板.....	38
10.6 挂起应用模板.....	40
10.7 激活应用模板.....	41
第十一章 镜像管理.....	41
11.1 查看私有仓库镜像.....	42
11.2 上传镜像到私有仓库.....	42
11.3 从私有仓库 pull 镜像.....	44

第一章 整体介绍

1.1 软件介绍

AIStation ,是浪潮自主研发的人工智能深度学习管理系统，面向深度学习计算集群，可以支持多种深度学习框架，能够快速部署深度学习训练环境，全面管理深度学习训练任务，为深度学习用户提供高效易用的平台；对计算集群的 CPU 及 GPU 资源进行统一的管理、调度及监控，有效的提高计算资源的利用率和生产率。

本系统有如下功能特性：

★ **提供完整的深度学习业务流程，方便易用，快速上手**

深度学习训练任务包含步骤比较多，从数据预处理，调整模型参数，分配计算资源，启动训练任务，监控训练过程，分析训练结果等，AIStation 提供全流程支持

★ **一键部署深度学习计算环境，快速启动训练任务，提高工作效率**

深度学习框架和模型较多，每个框架依赖的环境不同，当多个模型交叉使用时，对开发环境的要求比较复杂，AIStation 可以实现资源和工作环境的隔离及快速部署。

★ **训练任务监控及可视化，随时掌握训练进度及质量，及时发现模型问题**

深度学习模型训练耗时比较长，短则几个小时，长则几天，以往在训练完成后才意识到模型存在一些问题，为了避免这种问题，AIStation 实现了对训练过程实时监控并可视化训练过程，打印每一步的损失函数值的日志、训练误差或测试误差等。

★ **动态分配 GPU 资源，充分利用资源，提高资源利用率**

在 GPU 资源分配/调度方面，AIStation 兼顾大任务，小任务，长任务，短任务的需求，动态分配 GPU 资源，实现资源的合理共享，提高 GPU 资源使用率；

★ **全面的集群监控管理，实时掌握 CPU/GPU 资源使用情况及运行状态**

实时监控集群的使用情况，合理安排训练任务；及时发现运行中的问题，

提高集群的可靠性。

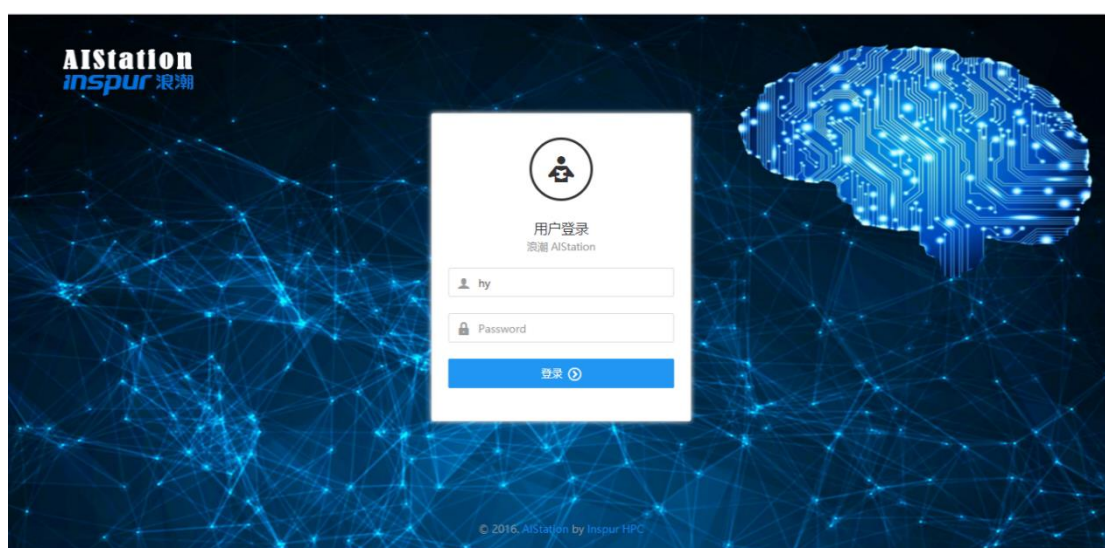


1.2 系统要求

浪潮 AIStation 人工智能深度学习管理系统需要以下软件环境支撑：

- 管理节点：
 - 操作系统：redhat linux 6/7.X或者Centos6/7.X版本。
 - 数据库：mysql5.1以上版本server端及client端。
 - web server：tomcat8版本。
- 客户端：
 - 操作系统：windows 7以上版本。
 - 浏览器：chrome50.1及以上版本或firefox50.1以上版本，并需要打开本地cookie功能。

第二章 登录界面



登录界面的用户和密码是跟管理服务器用户是同步的，安装完成后可以用

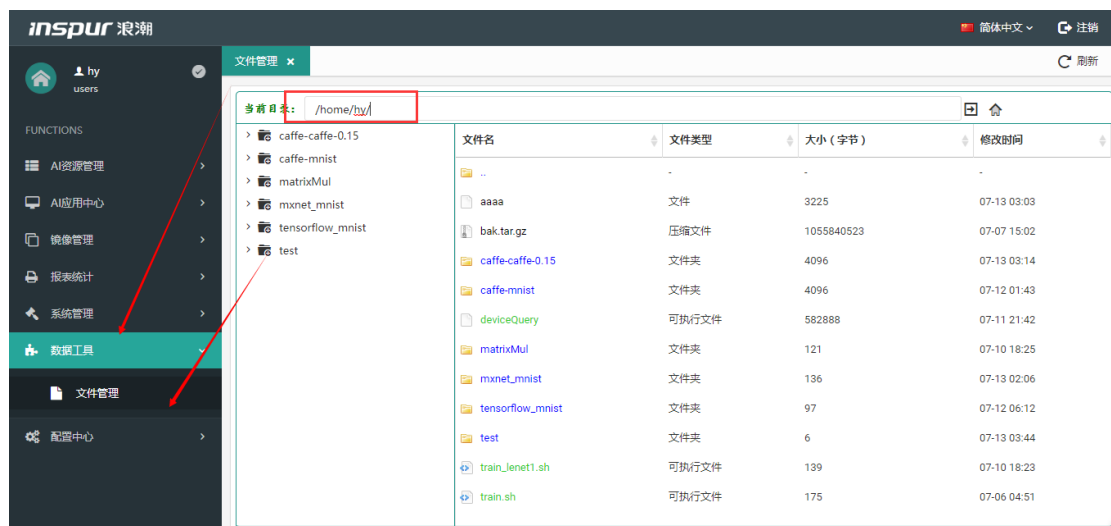
root 登录也可以按照普通用户进行登录。普通用户只能看到自己用户的提交的作业和完成作业，root 用户可以看到所有用户提交作业和完成作业。

第三章 数据及应用程序上传

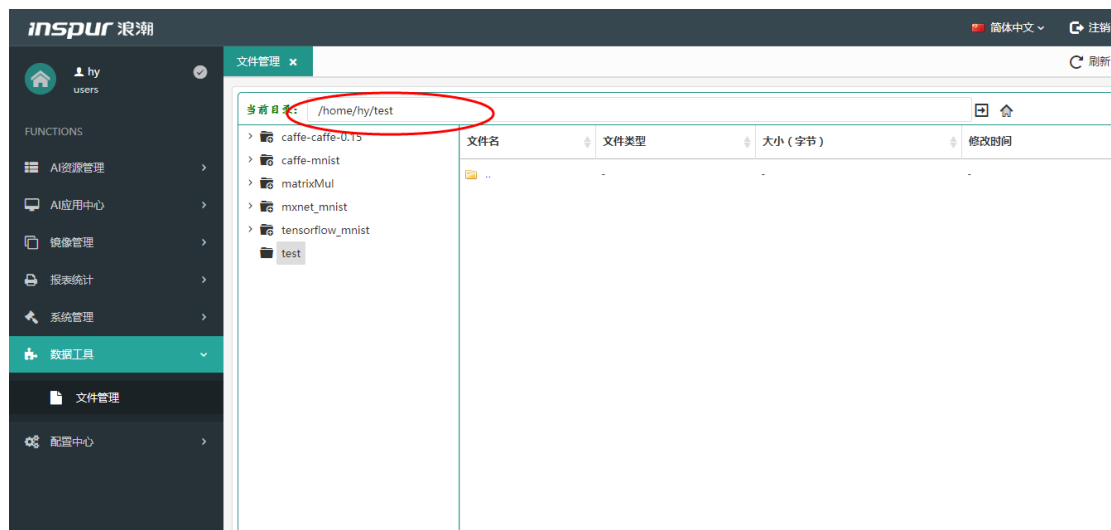
当用户需要把自己的数据或者应用程序放到集群中，可以使用文件管理模块来实现，可以将本地的数据和应用程序文件上传到集群中，也可以通过系统管理中的集群 shell 对文件进行解压操作。

3.1 上传数据

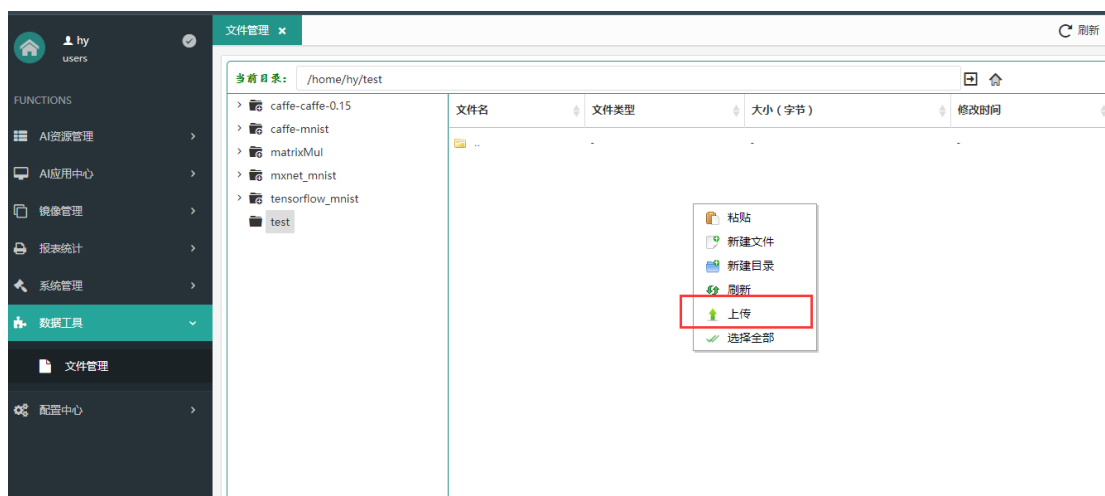
1) 点击数据工具中文件管理菜单，打开本地用户目录



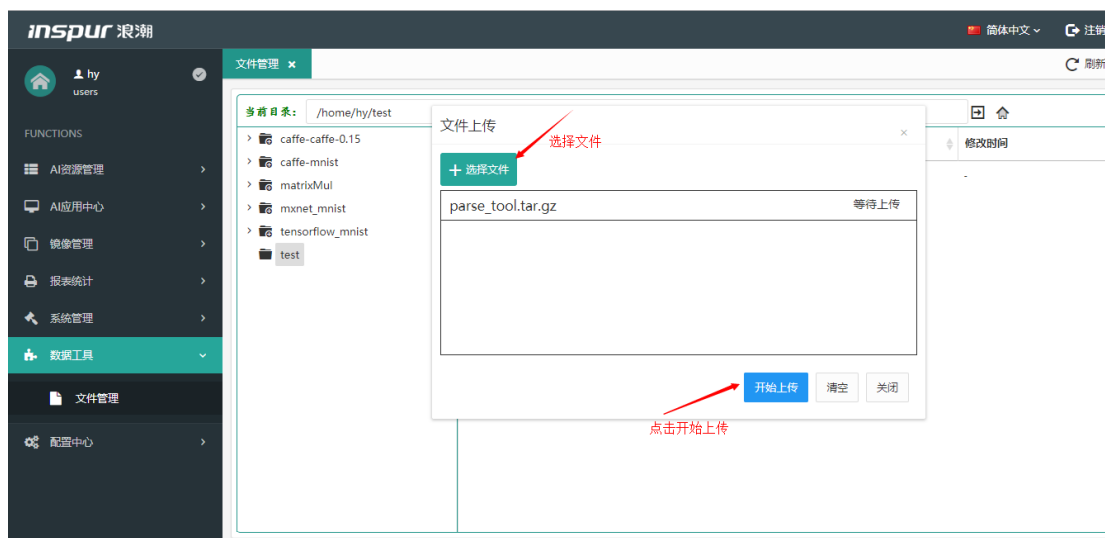
2) 进入你要上传数据的文件夹



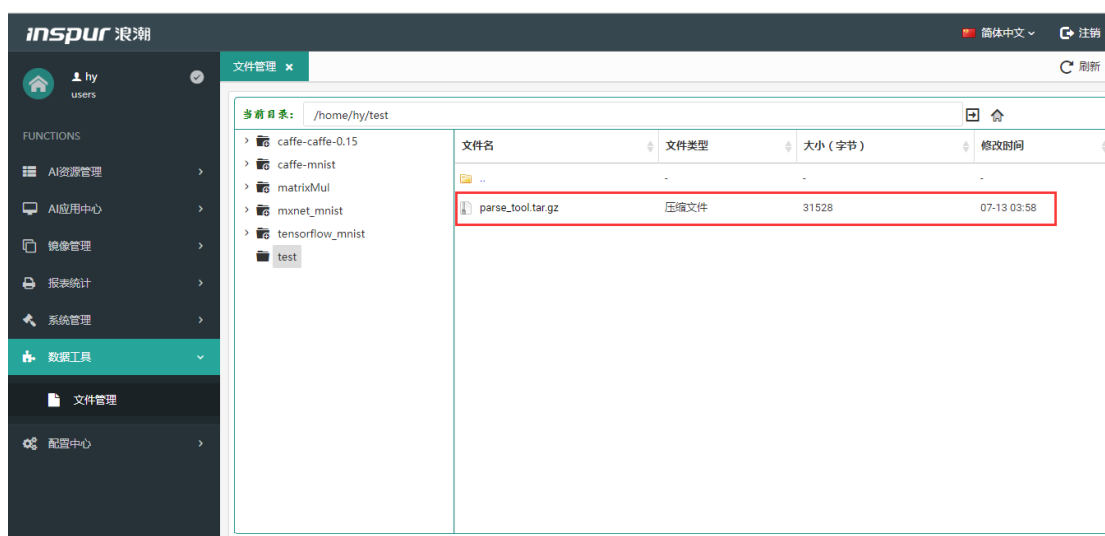
3 右侧框点击鼠标右键选择上传按钮



5) 选择上传的文件，并点击上传按钮。

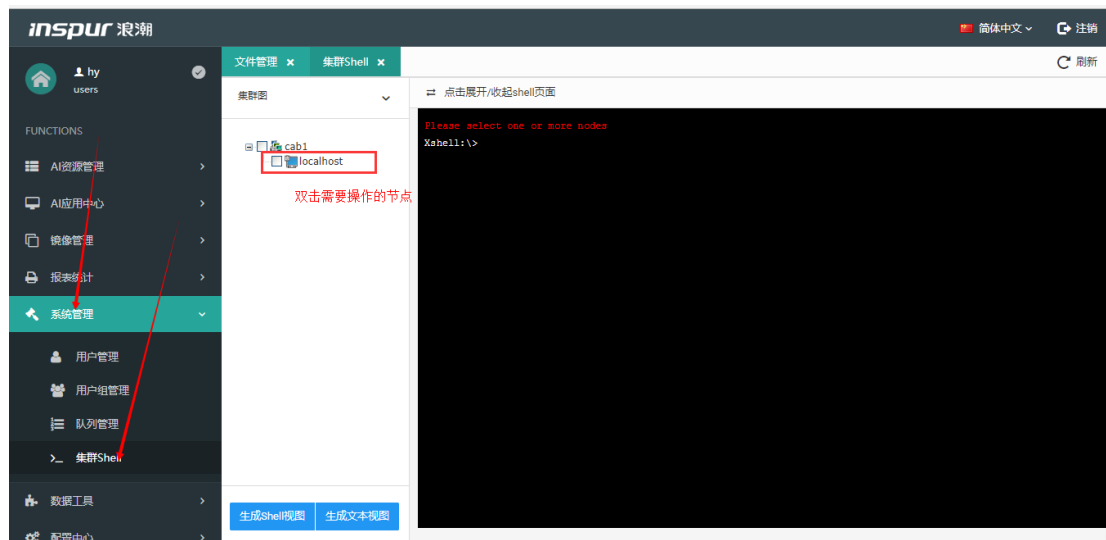


6) 上传成功后可在文件夹中看到已上传的文件。

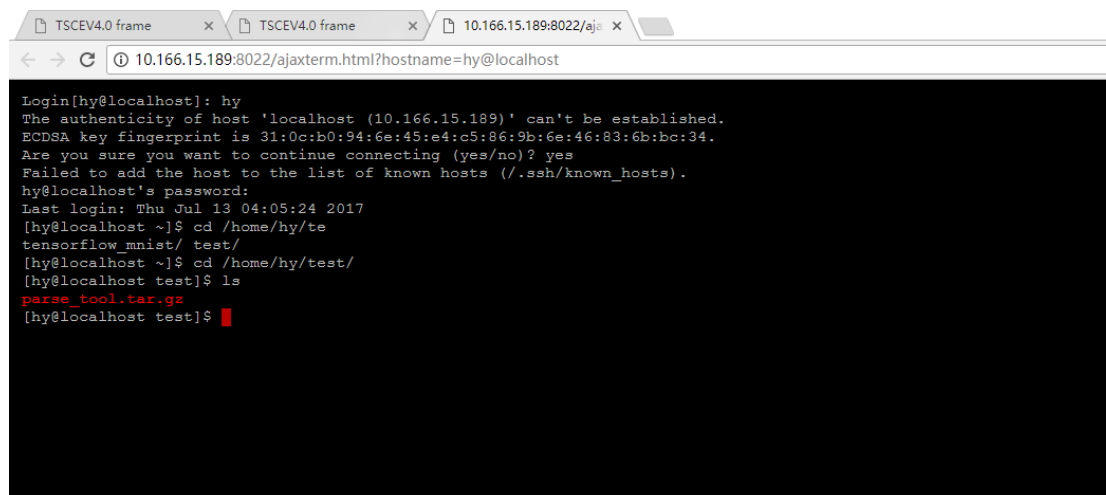


3.2 数据解压操作

1). 点击系统管理中的集群 shell，双击选择某个节点，



2) 进入 shell 节点，输入用户密码，进入刚才上传的文件夹中，对文件进行操作。



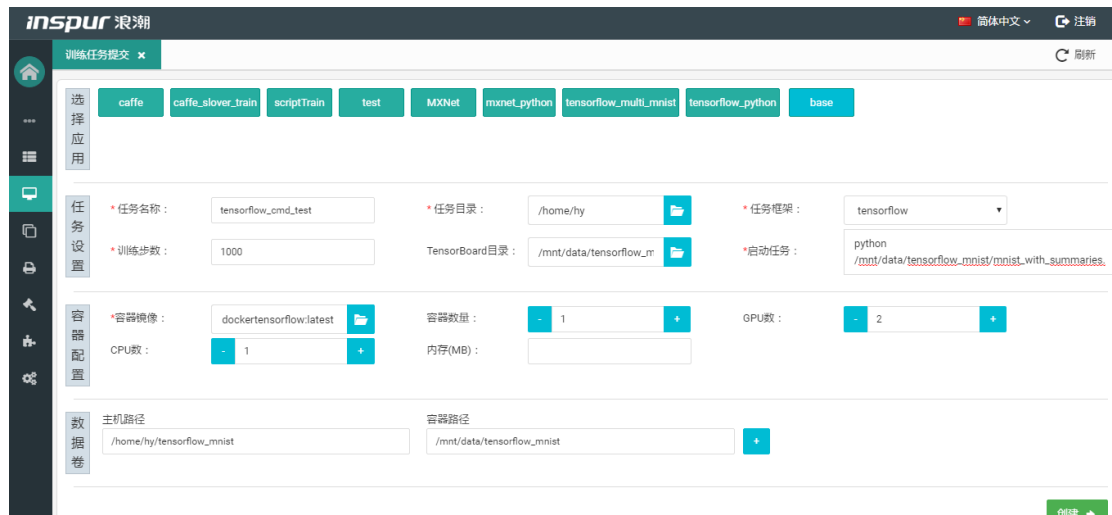
第四章 提交 Tensorflow 训练任务与管理

4.1 任务提交

--使用 *DeepLearning.tar.gz* 中 *tensorflow* 训练 *mnist* 的示例

1) 通过命令提交训练任务

a 打开“训练任务提交”页面，选择“base”应用并进行以下设置：



b 输入任务名称“tensorflow_cmd_test”、选择任务框架为” tensorflow”；

c 输入实际要训练的任务步数，系统将根据输入步数估算训练进度；

d 选择 TensorBoard 目录如” /mnt/data/tensorflow_mnist /logs”；

注意：该选项为可选输入，如果训练过程会生成 TensorFlow 的 event 文件则需设置该目录，系统会在该目录中查找 event 日志文件并进行解析

e 输入启动命令如下：

```
Python          /mnt/data/tensorflow_mnist/mnist_with_summaries.py      --
data_dir=/mnt/data/tensorflow_mnist/MNIST_data/                        --
log_dir=/mnt/data/tensorflow_mnist/logs/
```

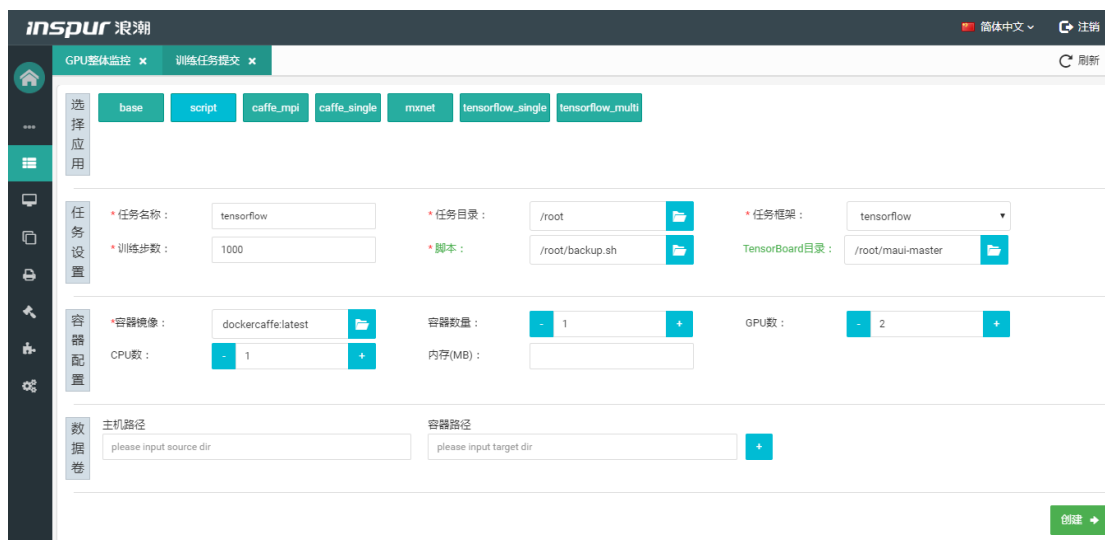
f 选择要启动的容器属性（容器镜像、容器数量、容器所需资源）；

g 设置数据卷目录 -- 映射 “ /home/hy/ tensorflow_mnist ” 到 “/mnt/data/tensorflow_mnist”

注意：此处需映射任务运行涉及到的所有目录，系统默认加载“任务目录”，该路径不需要重复加载

2)通过启动脚本提交训练任务

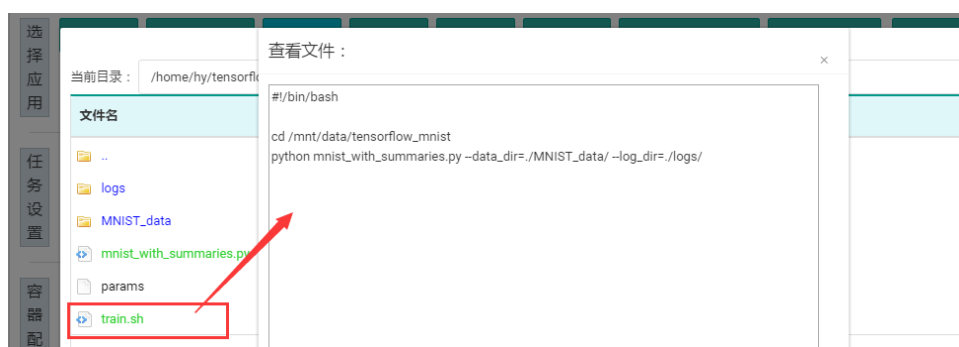
a 打开“训练任务提交”页面，选择“script”应用并进行以下设置：



b 输入任务名称、任务步数、选择任务框架为 tensorflow；

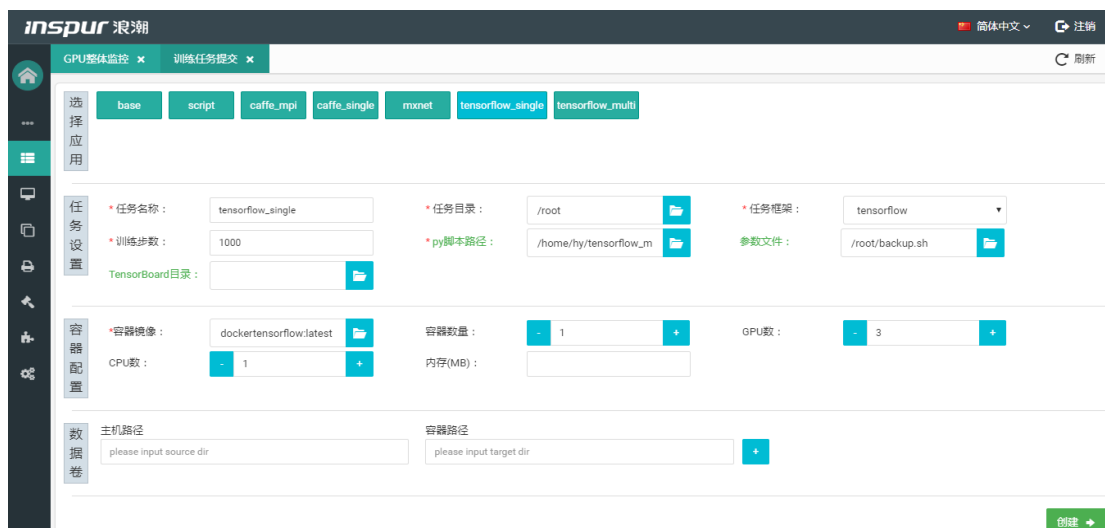
c 选择 TensorBoard 目录、容器属性以及加载数据卷目录（如例 A）；

d 选择要启动的脚本文件（如将例 A 中的命令保存为脚本文件并选择）；



3)通过应用参数提交训练任务

a 打开“训练任务提交”页面，选择“tensorflow_single”应用并进行以下设置：



- b 输入任务名称、任务步数、选择任务框架为 tensorflow;
 - c 选择 TensorBoard 目录 (如例 A);
 - d 选择要运行的 python 文件, 如 “/mnt/data/tensorflow_mnist/mnist_with_summaries.py”;
 - e 选择运行 python 使用的参数文件, 如 “/mnt/data/tensorflow_mnist/params”
- 注意: 参数文件格式为 “参数名 参数值”, 任务运行时系统会判断是否存在参数文件, 如果存在则按照文件内容增加 python 运行时的参数



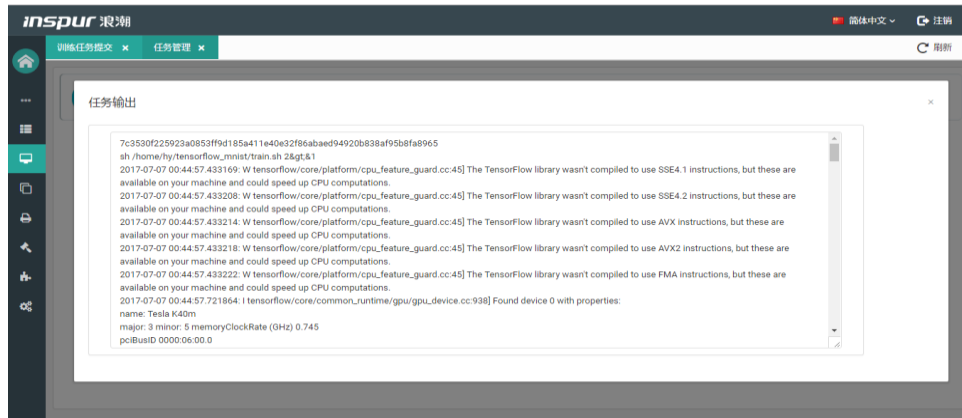
4.2 训练过程管理

1) 查看训练任务运行情况

打开 “任务管理” 页面可以查看当前在运行的任务列表



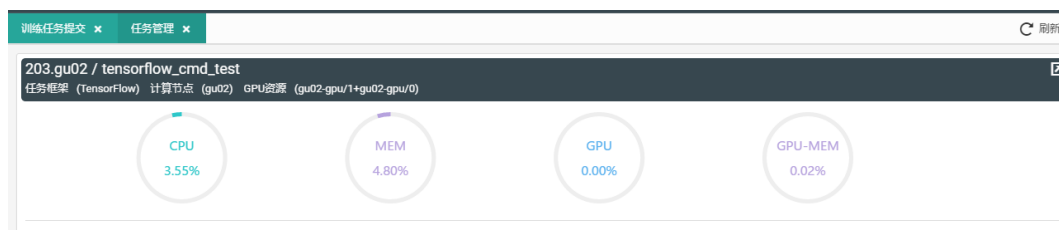
点击任务记录后面的操作按钮可查看相关任务信息, 如:



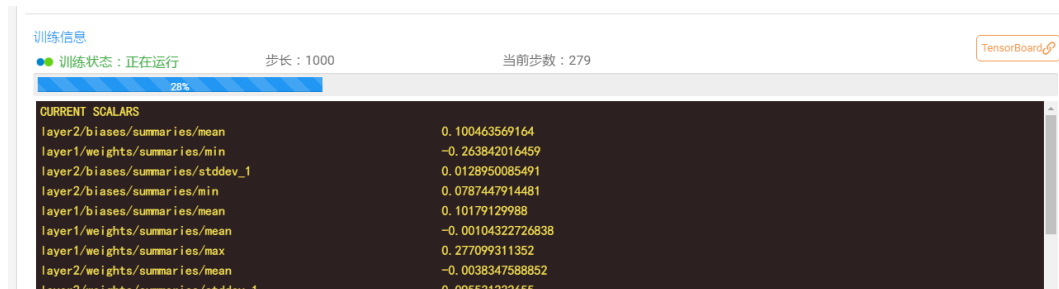
2)查看训练进度及结果

双击任务记录可查看任务详情信息，包括任务节点信息、训练信息以及容器信息。

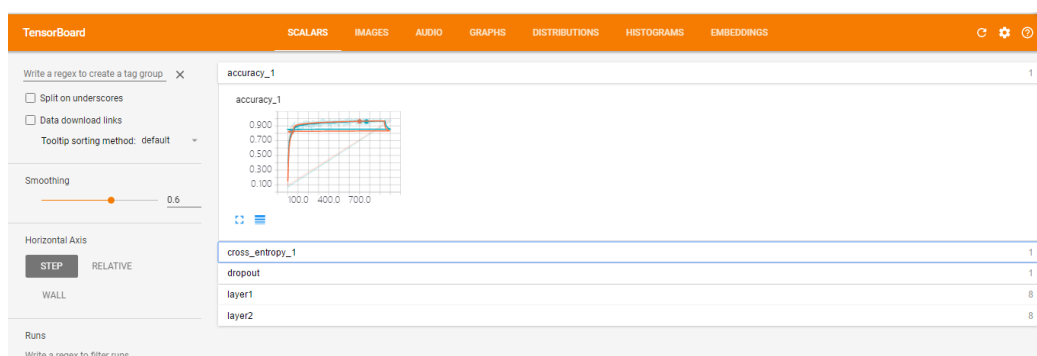
查看任务节点信息



查看训练信息



点击“TensorBoard”按钮可通过 TensorBoard 工具查看训练信息



容器列表信息

容器列表						
203.gu02@gu02	CPU利用率 (100.04%)	内存使用 (1.203GiB/125.8GiB)	GPU利用率 (1.0%+0.0%)	显存使用 (1.84.12/11439.88+0.267.1 2/11439.88)	Tesla*2	≡ ☰ 🖨

第五章提交 caffe 任务与管理

5.1 任务提交

--测试 caffe 自带的示例；使用 lenet 训练 mnist 数据集

1) 通过命令提交训练任务

a 打开“训练任务提交”页面，选择“base”应用并进行以下设置：

The screenshot shows the 'inspur 浪潮' AIStation interface. The 'Training Task Submission' page is active. The 'Application' dropdown is set to 'base'. The 'Task Name' is 'base', 'Task Directory' is '/home/hy', and 'Task Framework' is 'caffe'. The 'Task Steps' is 2000. The 'Container Image' is 'caffe.gpu', 'Container Quantity' is 1, and 'GPU Count' is 2. The 'Host Path' is 'please input source dir' and 'Container Path' is 'please input target dir'. A 'Create' button is at the bottom right.

b 输入任务名称“caffe-cmd-test”、选择任务框架为“caffe”；

c 输入实际要训练的任务步数，系统将根据输入步数估算训练进度；

d 输入启动命令如下：

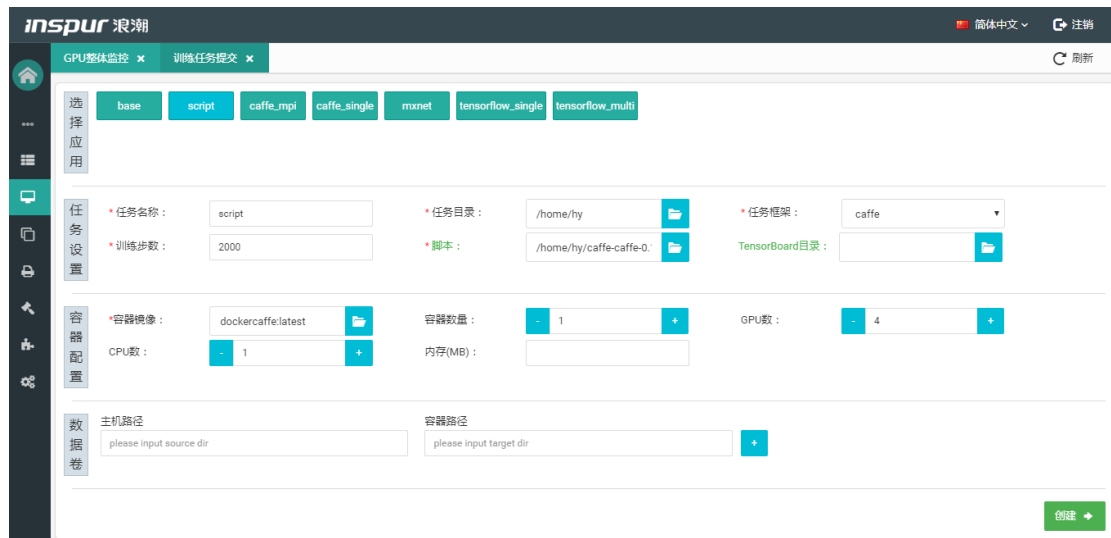
```
cd /home/hy/caffe-caffe-0.15 && ./build/tools/caffe train --solver=/home/hy/caffe-caffe-0.15/examples/mnist/lenet_solver.prototxt
```

e 选择要启动的容器属性（容器镜像、容器数量、容器所需资源）；

f 点击“创建”按钮；

2)通过启动脚本提交训练任务

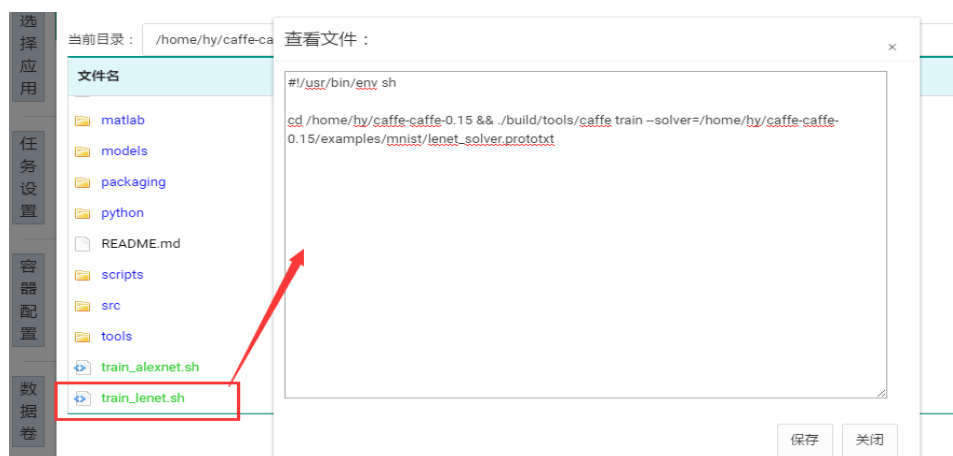
a 打开“训练任务提交”页面，选择“script”应用并进行以下设置：



b 输入任务名称、任务步数、选择任务框架为 caffe；

c 选择容器属性以及加载数据卷目录（如例 A）；

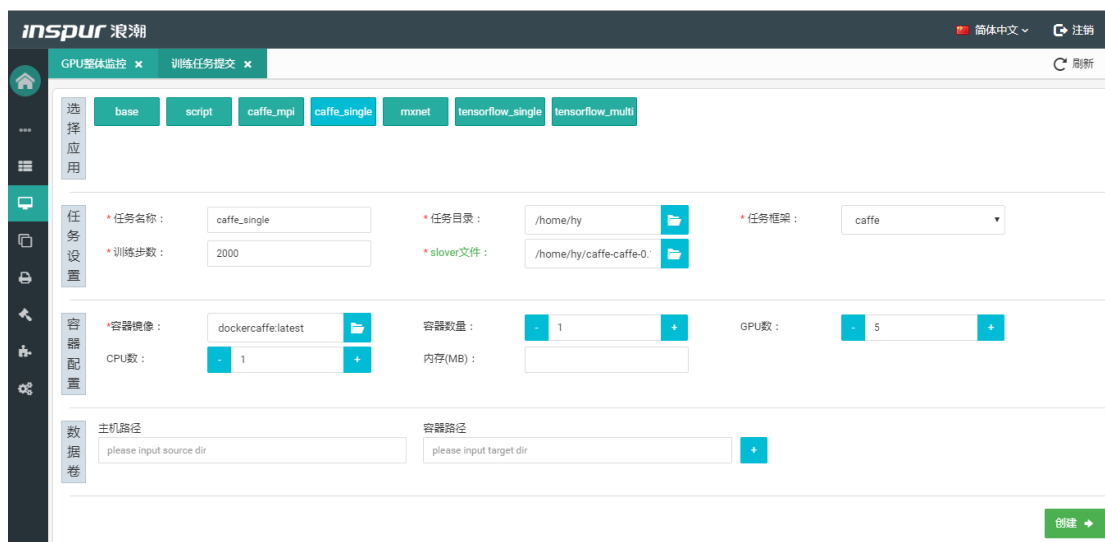
d 选择要启动的脚本文件（如将例 A 中的命令保存为脚本文件并选择）；



e 点击“创建”按钮；

3)通过应用参数提交训练任务

a 打开“训练任务提交”页面，选择“caffe_single”并进行以下设置：



- b 输入任务名称、任务步数、选择任务框架为 caffe;
 - c 选择容器属性以及加载数据卷目录（如例 A）;
 - d 选择要使用的 solver 文件,如/home/hy/caffe-mnist/lenet_solver.prototxt;
- 注意: solver 文件及 net 模型文件中的路径需设置为绝对路径,并确保该路径已被加载至数据卷。



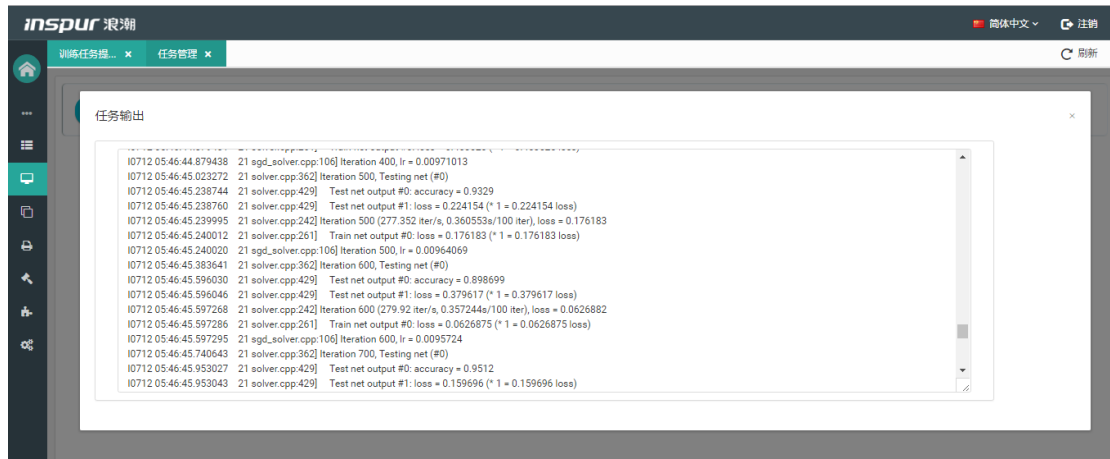
4.2 训练过程管理

1)查看训练任务运行情况

打开“任务管理”页面可以查看当前在运行的任务列表



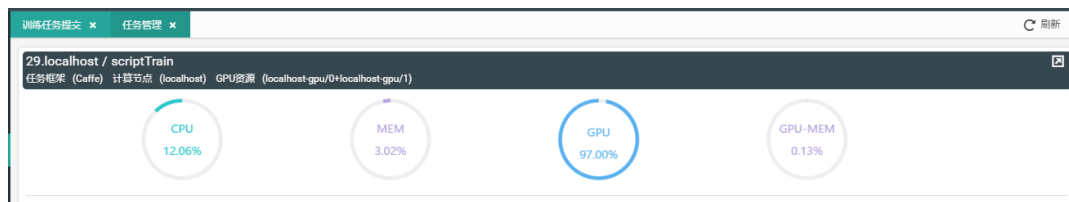
点击任务记录后面的操作按钮可查看相关任务信息，如：



2) 查看训练进度及结果

双击任务记录可查看任务详情信息，包括任务节点信息、训练信息以及容器信息。

查看任务节点信息



查看训练信息



容器列表信息

容器列表

29.localhost@localhost	CPU利用率 (0.03%)	内存使用 (7.355MiB/251.6GiB)	GPU利用率 (0.69%+1.75%)	显存使用 (0.2313.0/16276.25+1.1915.0/ 16276.25)	Tecla*2
------------------------	-------------------	-----------------------------	-------------------------	---	---------

第六章 提交 mxnet 任务与管理

6.1 任务提交

1)通过命令提交训练任务

a 打开“训练任务提交”页面，选择“base”应用并进行以下设置：

The screenshot shows the 'inspur 浪潮' AIStation interface for submitting training tasks. The 'Training Task Submission' tab is active. Under 'Select Application', 'base' is chosen. The configuration fields are as follows:

- Task Name: base
- Task Directory: /home/hy
- Task Framework: mxnet
- Training Steps: 2000
- Container Image: (empty)
- Container Quantity: 1
- GPU Count: 5
- Host Path: please input source dir
- Container Path: please input target dir

A 'Create' button is located at the bottom right of the form.

b 输入任务名称、选择任务框架为”mxnet”；

c 输入实际要训练的任务步数，系统将根据输入步数估算训练进度；

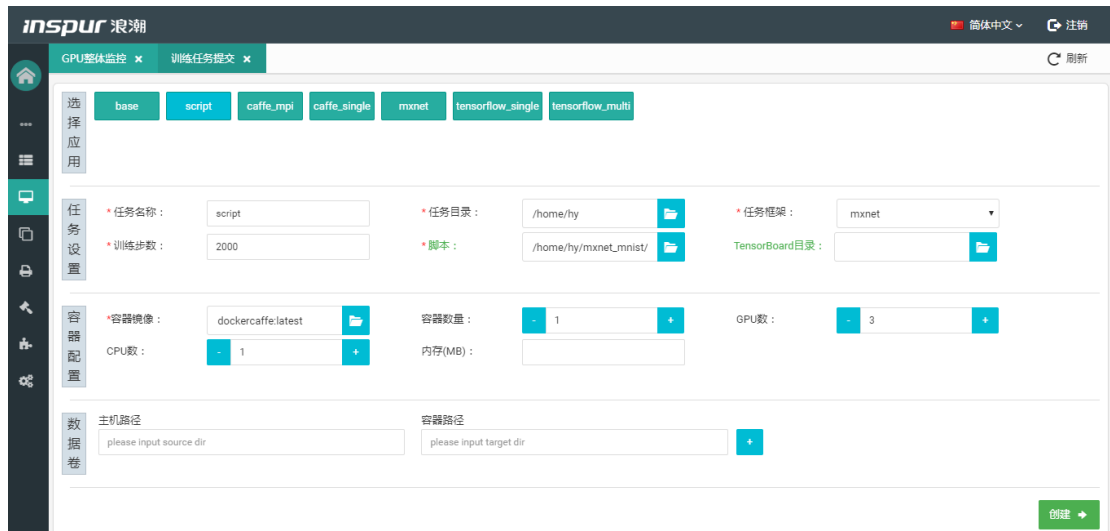
d 输入启动命令如下：

```
python /home/hy/mxnet_mnist/train.py --data-dir /home/hy/mxnet_mnist/data --batch-size 64 --lr 0.001 --num-epochs 20 --gpus 0
```

e 选择要启动的容器属性（容器镜像、容器数量、容器所需资源）；

2)通过启动脚本提交训练任务

a 打开“训练任务提交”页面，选择“script”应用并进行以下设置：

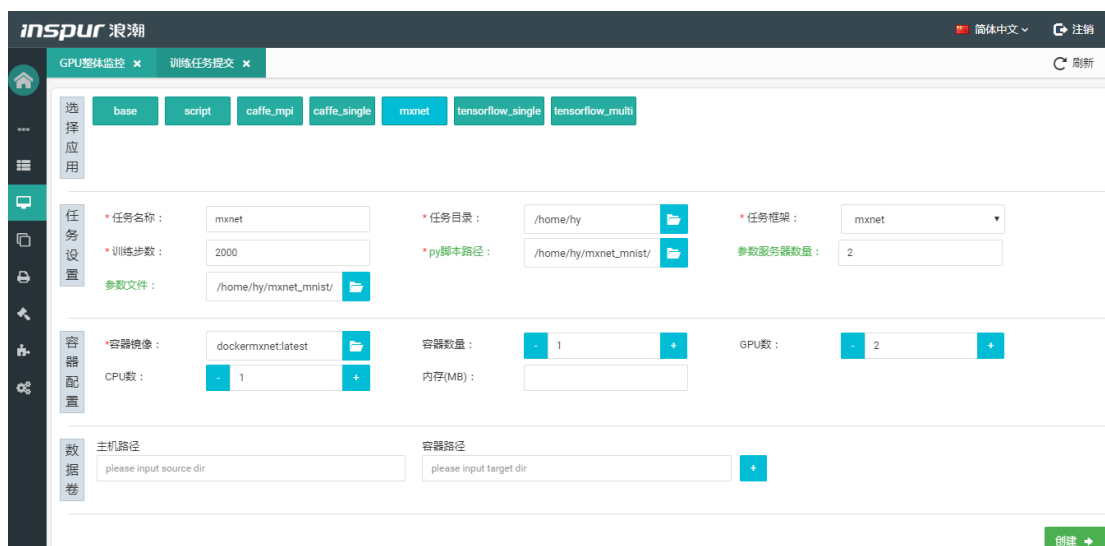


- b 输入任务名称、任务步数、选择任务框架为 mxnet;
- c 选择要启动的容器属性（容器镜像、容器数量、容器所需资源）
- d 选择要启动的脚本文件（如将例 A 中的命令保存为脚本文件并选择）;



3) 设置应用参数提交训练任务

- a 打开“训练任务提交”页面，选择“mxnet”应用并进行以下设置：

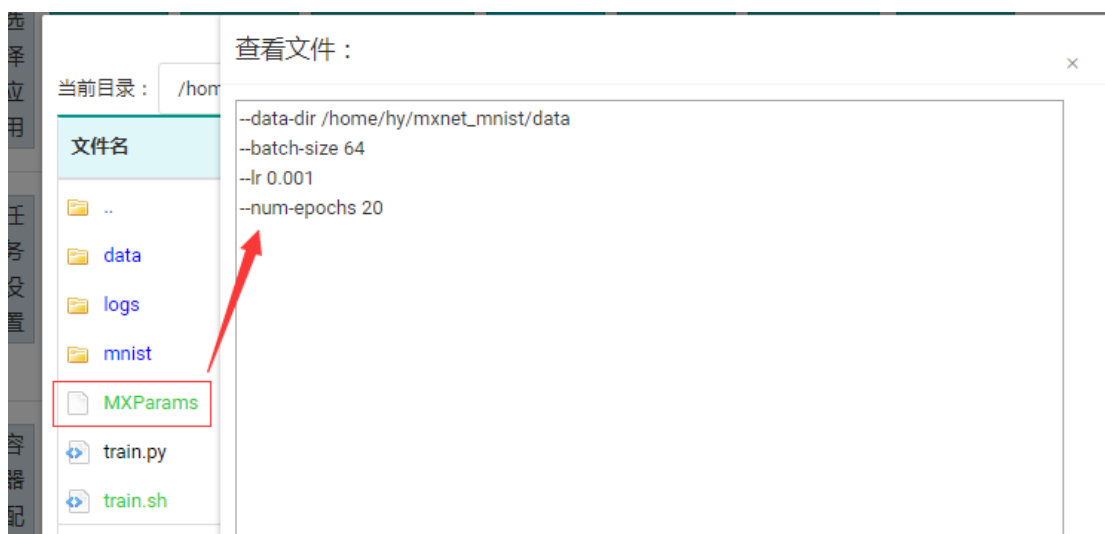


b 输入任务名称、任务步数、选择任务框架为 mxnet;

c 选择要运行的 python 文件，如 “/home/hy/mxnet_mnist/train.py”;

d 选择运行 python 使用的参数文件，如 “/home/hy/mxnet_mnist/ MXParams”

注意：参数文件格式为“参数名 参数值”，任务运行时系统会判断是否存在参数文件，如果存在则按照文件内容增加 python 运行时的参数



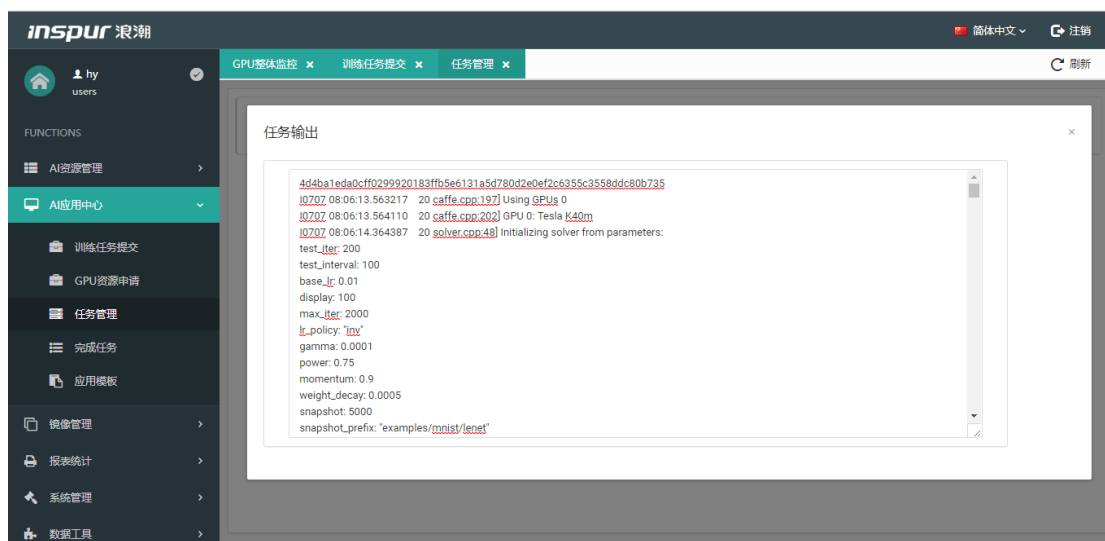
6.2 训练过程管理

1)查看训练任务节点运行情况

打开“任务管理”页面可以查看当前在运行的任务列表



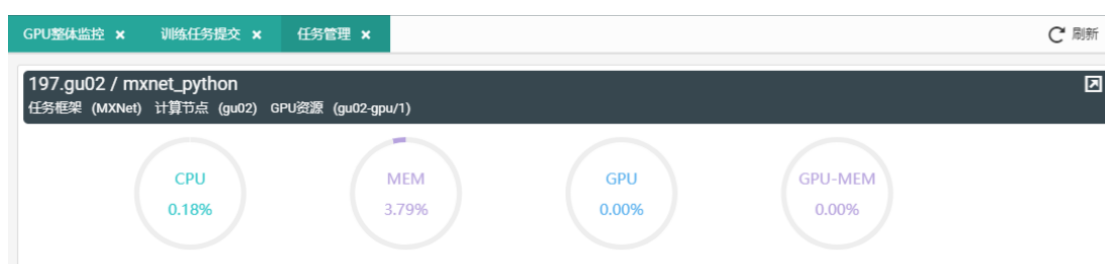
点击任务记录后面的操作按钮可查看相关任务信息，如：



2)查看训练进度及结果

双击任务记录可查看任务详情信息，包括任务节点信息、训练信息以及容器信息。

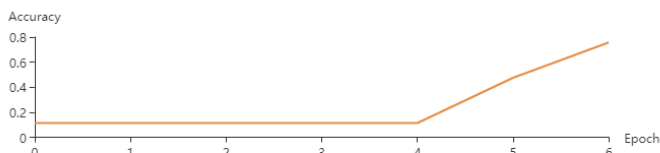
查看任务节点信息



查看训练信息

训练信息

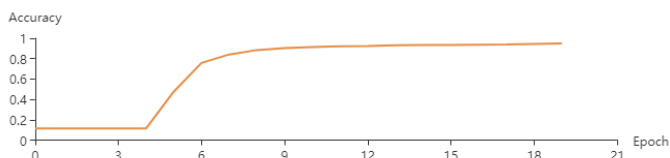
基本信息: (batch_size=64, data_dir='/home/hy/mxnet_mnist/data', gpus='0', init='uniform', kv_store='local', load_epoch=None, log_dir='log/', lr=0.001, lr_factor=1, lr_factor_epoch=1, model_prefix=None, name='uniform_relu', network='mlp', num_epochs=20, num_examples=60000, save_model_prefix=None)



● 训练状态: 正在运行 Epoch:6 Validation-accuracy:0.757212

训练信息

基本信息: (batch_size=64, data_dir='/home/hy/mxnet_mnist/data', gpus='0', init='uniform', kv_store='local', load_epoch=None, log_dir='log/', lr=0.001, lr_factor=1, lr_factor_epoch=1, model_prefix=None, name='uniform_relu', network='mlp', num_epochs=20, num_examples=60000, save_model_prefix=None)



☑ 训练状态: 已完成 Epoch:19 Validation-accuracy:0.948317

容器列表信息

容器列表

容器ID	CPU利用率	内存使用	GPU利用率	显存使用	GPU名称	操作
197.gu02@gu02	100.12%	1010MiB/125.8GiB	1.42%	1:161.0/11439.88	Tesla P4	

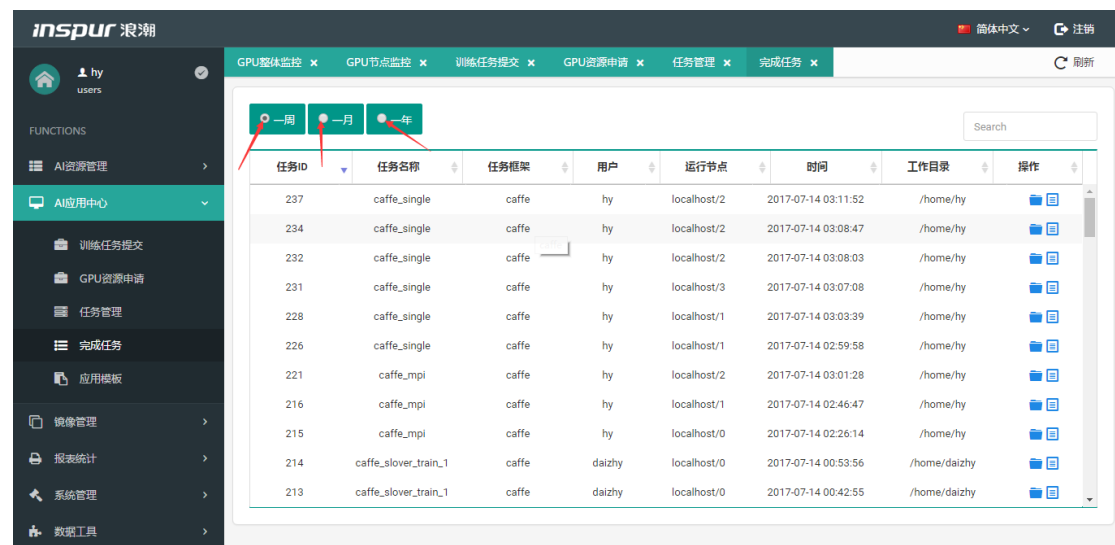
第七章 完成作业管理

1)完成任务查看

1 点击导航栏 AI 应用中心中的完成任务

任务ID	任务名称	任务框架	用户	运行节点	时间	工作目录	操作
237	caffe_single	caffe	hy	localhost/2	2017-07-14 03:11:52	/home/hy	
234	caffe_single	caffe	hy	localhost/2	2017-07-14 03:08:47	/home/hy	
232	caffe_single	caffe	hy	localhost/2	2017-07-14 03:08:03	/home/hy	
231	caffe_single	caffe	hy	localhost/3	2017-07-14 03:07:08	/home/hy	
228	caffe_single	caffe	hy	localhost/1	2017-07-14 03:03:39	/home/hy	
226	caffe_single	caffe	hy	localhost/1	2017-07-14 02:59:58	/home/hy	
221	caffe_mpi	caffe	hy	localhost/2	2017-07-14 03:01:28	/home/hy	
216	caffe_mpi	caffe	hy	localhost/1	2017-07-14 02:46:47	/home/hy	
215	caffe_mpi	caffe	hy	localhost/0	2017-07-14 02:26:14	/home/hy	
214	caffe_slover_train_1	caffe	daizhy	localhost/0	2017-07-14 00:53:56	/home/daizhy	
213	caffe_slover_train_1	caffe	daizhy	localhost/0	2017-07-14 00:42:55	/home/daizhy	

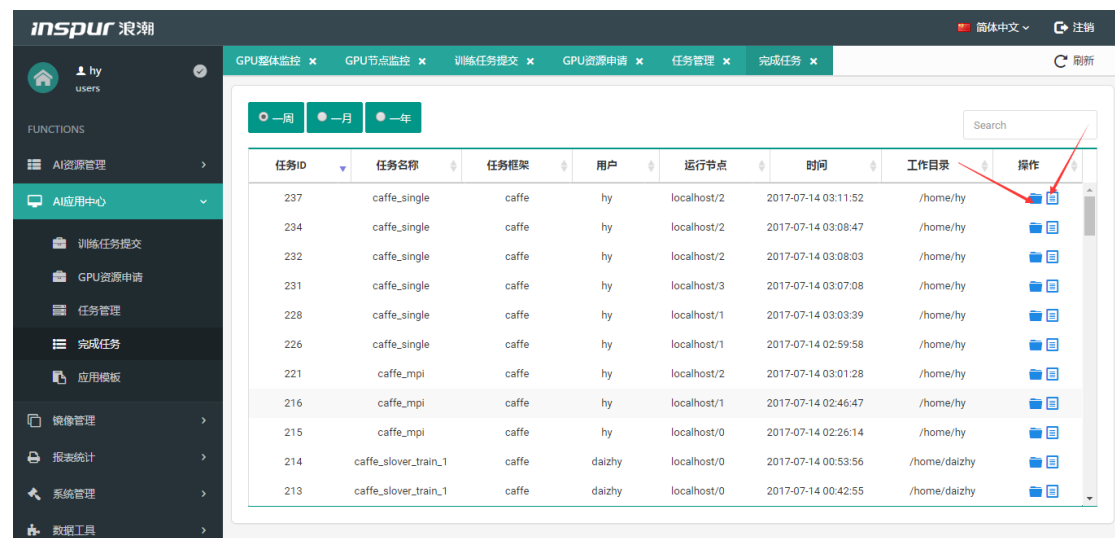
2 分别可以查看一周一月一年所执行过的任务结果的记录



The screenshot shows the AISTation interface with the '任务管理' (Task Management) tab selected. The left sidebar lists various functions under 'AI应用中心'. The main content area displays a table of task results. At the top of the table, there are three radio buttons for filtering by time period: '一周' (One Week), '一月' (One Month), and '一年' (One Year). Red arrows point to these buttons. The table has columns for Task ID, Task Name, Task Framework, User, Running Node, Time, Working Directory, and Actions. The data shows tasks executed by user 'hy' on various nodes, mostly using the 'caffe' framework.

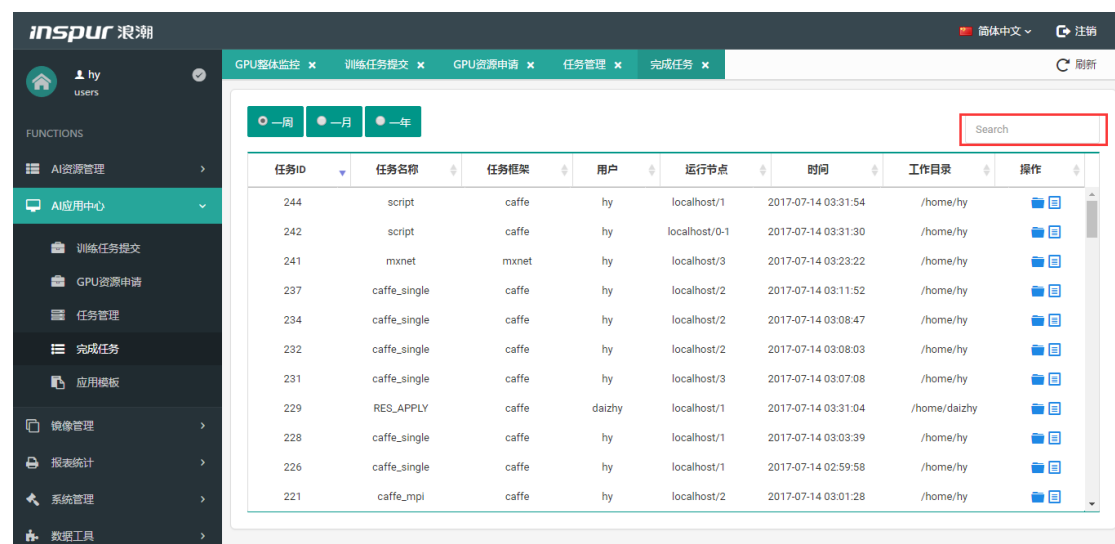
任务ID	任务名称	任务框架	用户	运行节点	时间	工作目录	操作
237	caffe_single	caffe	hy	localhost/2	2017-07-14 03:11:52	/home/hy	[Icon]
234	caffe_single	caffe	hy	localhost/2	2017-07-14 03:08:47	/home/hy	[Icon]
232	caffe_single	caffe	hy	localhost/2	2017-07-14 03:08:03	/home/hy	[Icon]
231	caffe_single	caffe	hy	localhost/3	2017-07-14 03:07:08	/home/hy	[Icon]
228	caffe_single	caffe	hy	localhost/1	2017-07-14 03:03:39	/home/hy	[Icon]
226	caffe_single	caffe	hy	localhost/1	2017-07-14 02:59:58	/home/hy	[Icon]
221	caffe_mpi	caffe	hy	localhost/2	2017-07-14 03:01:28	/home/hy	[Icon]
216	caffe_mpi	caffe	hy	localhost/1	2017-07-14 02:46:47	/home/hy	[Icon]
215	caffe_mpi	caffe	hy	localhost/0	2017-07-14 02:26:14	/home/hy	[Icon]
214	caffe_slover_train_1	caffe	daizhy	localhost/0	2017-07-14 00:53:56	/home/daizhy	[Icon]
213	caffe_slover_train_1	caffe	daizhy	localhost/0	2017-07-14 00:42:55	/home/daizhy	[Icon]

3 查看任务的输出信息，以及进入用户的工作目录



This screenshot is similar to the previous one, showing the task results table. A red arrow points to the '操作' (Actions) column, specifically to the icon that allows viewing the task's output information or entering the user's working directory.

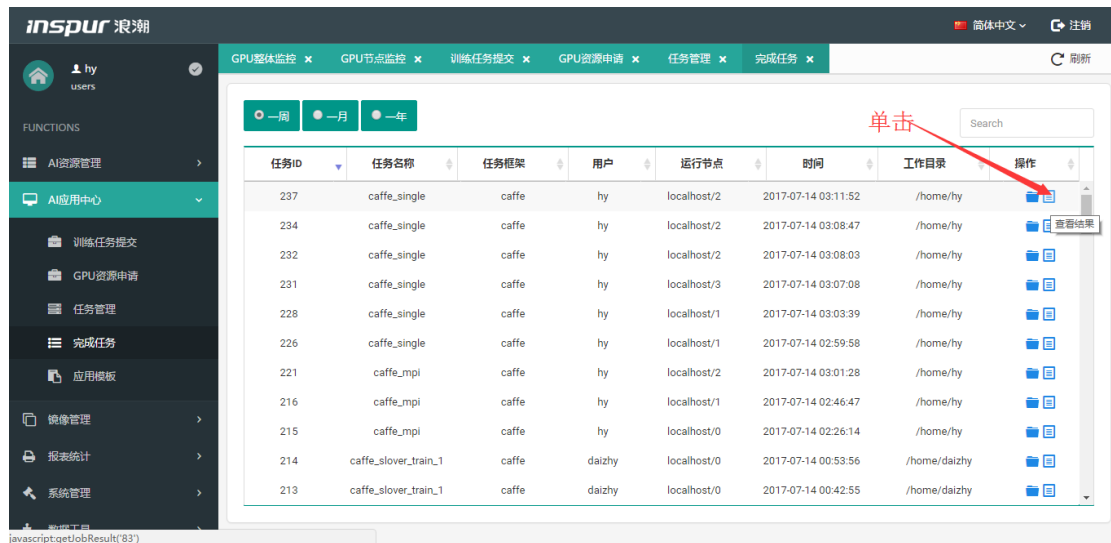
4 全字段搜索所完成的任务



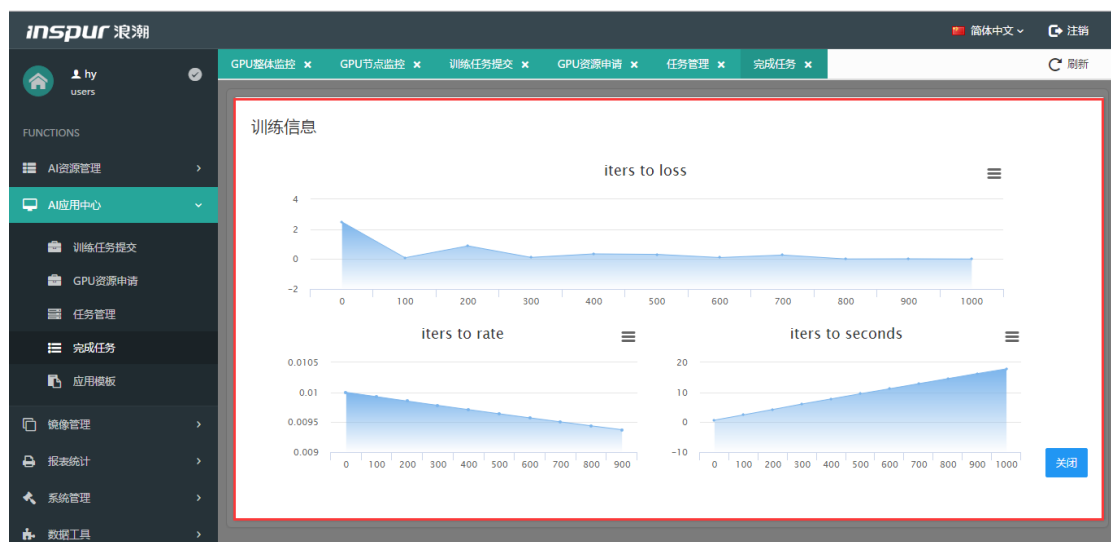
The screenshot shows the AISTation interface with the '任务管理' (Task Management) tab selected. The left sidebar lists various functions under 'AI应用中心'. The main content area displays a table of task results. At the top of the table, there are three radio buttons for filtering by time period: '一周', '一月', and '一年'. A red box highlights the search bar. The table has columns for Task ID, Task Name, Task Framework, User, Running Node, Time, Working Directory, and Actions. The data shows tasks executed by user 'hy' on various nodes, mostly using the 'caffe' framework.

任务ID	任务名称	任务框架	用户	运行节点	时间	工作目录	操作
244	script	caffe	hy	localhost/1	2017-07-14 03:31:54	/home/hy	[Icon]
242	script	caffe	hy	localhost/0-1	2017-07-14 03:31:30	/home/hy	[Icon]
241	mxnet	mxnet	hy	localhost/3	2017-07-14 03:23:22	/home/hy	[Icon]
237	caffe_single	caffe	hy	localhost/2	2017-07-14 03:11:52	/home/hy	[Icon]
234	caffe_single	caffe	hy	localhost/2	2017-07-14 03:08:47	/home/hy	[Icon]
232	caffe_single	caffe	hy	localhost/2	2017-07-14 03:08:03	/home/hy	[Icon]
231	caffe_single	caffe	hy	localhost/3	2017-07-14 03:07:08	/home/hy	[Icon]
229	RES_APPLY	caffe	daizhy	localhost/1	2017-07-14 03:31:04	/home/daizhy	[Icon]
228	caffe_single	caffe	hy	localhost/1	2017-07-14 03:03:39	/home/hy	[Icon]
226	caffe_single	caffe	hy	localhost/1	2017-07-14 02:59:58	/home/hy	[Icon]
221	caffe_mpi	caffe	hy	localhost/2	2017-07-14 03:01:28	/home/hy	[Icon]

2)查看任务训练信息



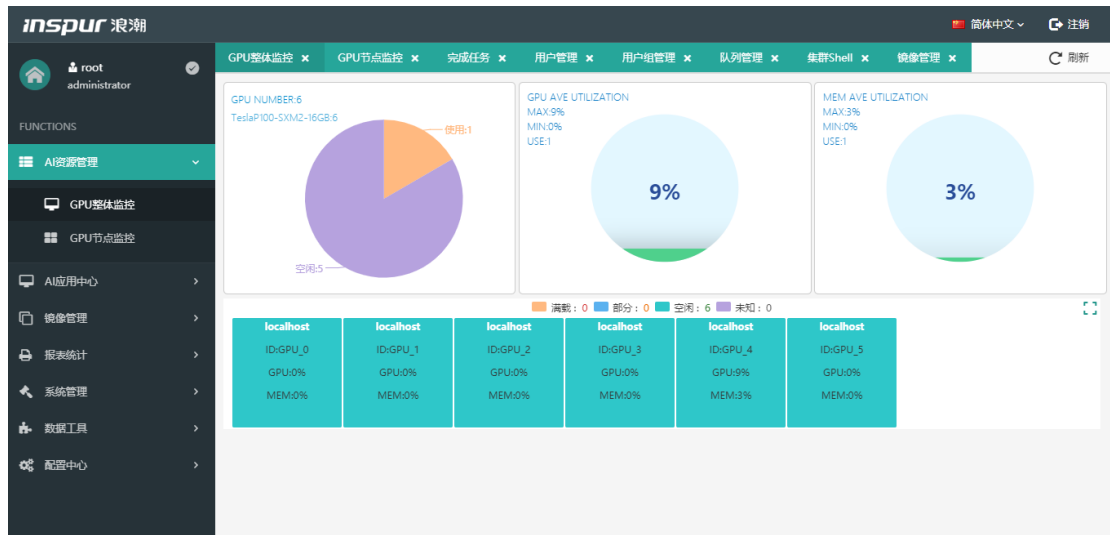
点击查看结果后，显示对应任务的分析视图



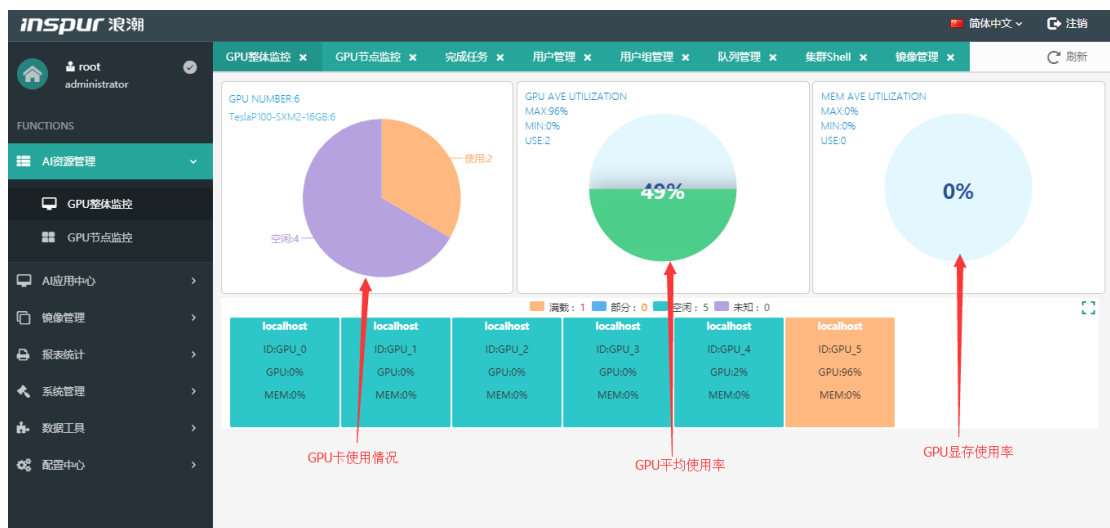
第八章 集群监控管理

1)GPU 集群监控

1 点击导航栏 AI 应用中心中的 GPU 整体监控，该页面显示整个 Gpu 集群使用情况，

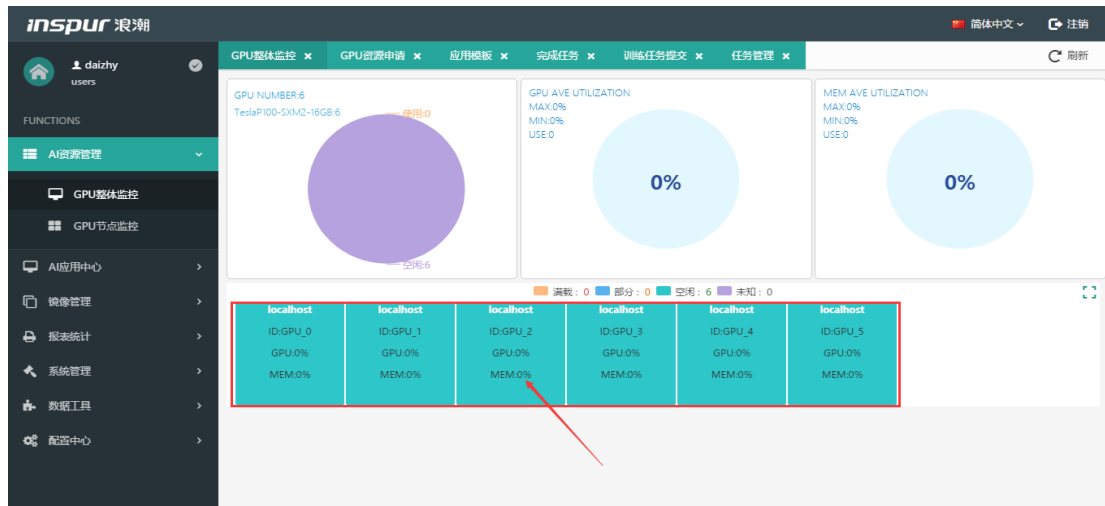


2 下图指出了整体 GPU 使用情况，统计有 gpu 卡使用个数和空闲个数，Gpu 平均利用率，Gpu 显存使用率。



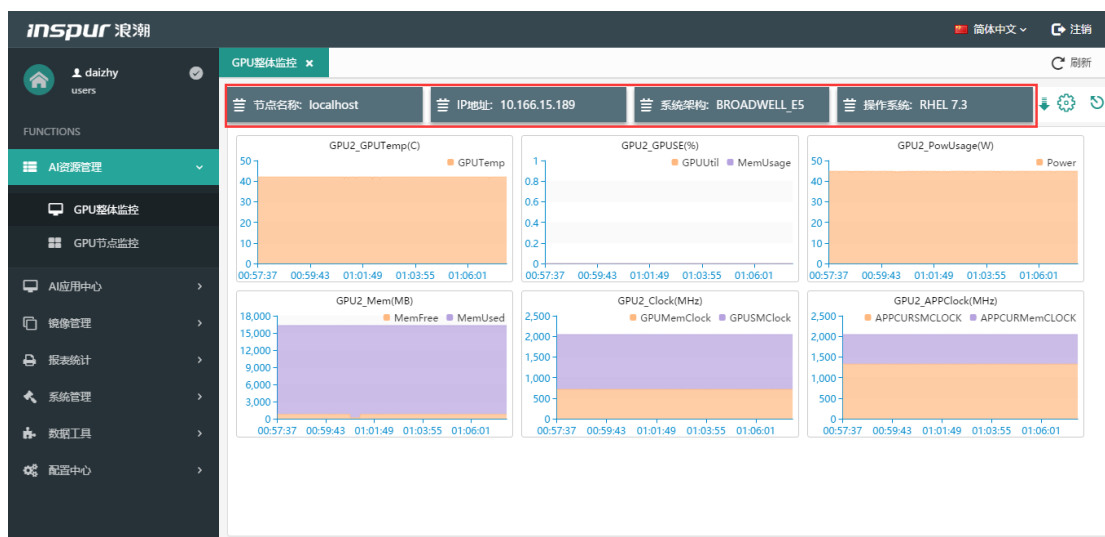
2)GPU 单卡实时运行监控

1 以下列表列出了节点对应的 gpu 的运行情况，例如：箭头所指含义为，localhost 节点对应 gpu_2 卡的运行情况，统计的是单卡的利用率和显存利用率。

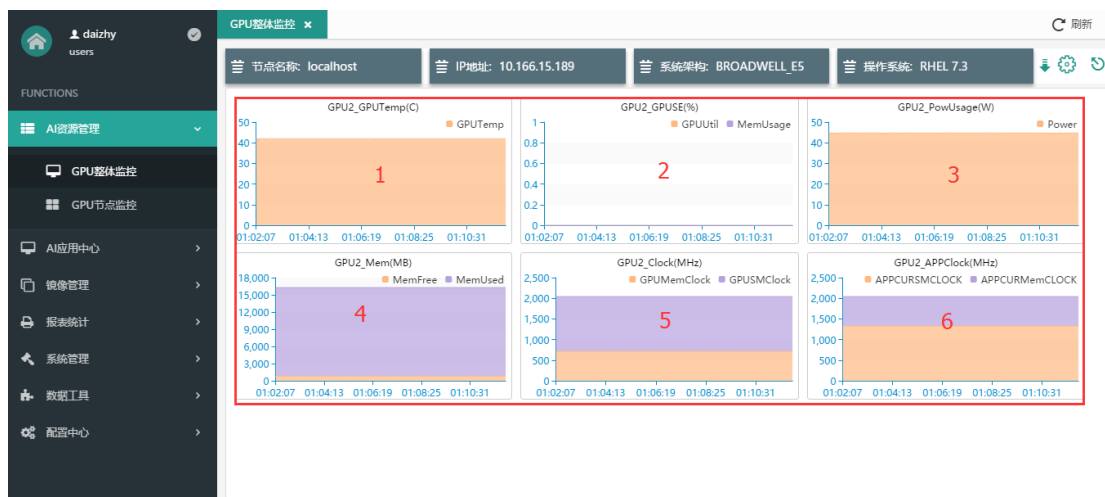


2 查看某单个 gpu 卡的实时运行情况，如查看 GPU2，左键单击即可进入此 GPU2 实时运行页面中。

最上一栏标出了节点名称，ip，操作系统等基本信息



下面的图形，横轴表示时间，纵轴表示（从左直右，从上到下）1、表示此时间段的温度；2、表示 GPU2 的使用率；3、GPU2 的功耗；4、GPU2 的内存情况；4、GPU2 的时钟频率；5、GPU2 的应用时钟频率。



第九章 GPU 资源申请

1)容器资源申请

1 点击 GPU 资源申请页面

2 资源设置方法:

- 申请名称: 申请容器的名称, 例如 RES_APPLY
- 资源框架: 选择容器的框架 (caffe、mxnet、tensorflow), 例如选择 caffe
- 资源镜像: 选择 docker 中已有的镜像, 选择 caffe: gpu 镜像
- 容器数量: 创建容器的数量, 例如创建 1 个
- GPU 数: 该容器独有的 GPU 数, 例如设置 4 个 GPU 卡
- CPU 数: 该容器独有的 CPU 数, 例如设置 1 个

g 内存(MB)：该容器申请的内存大小，例如设置 20000

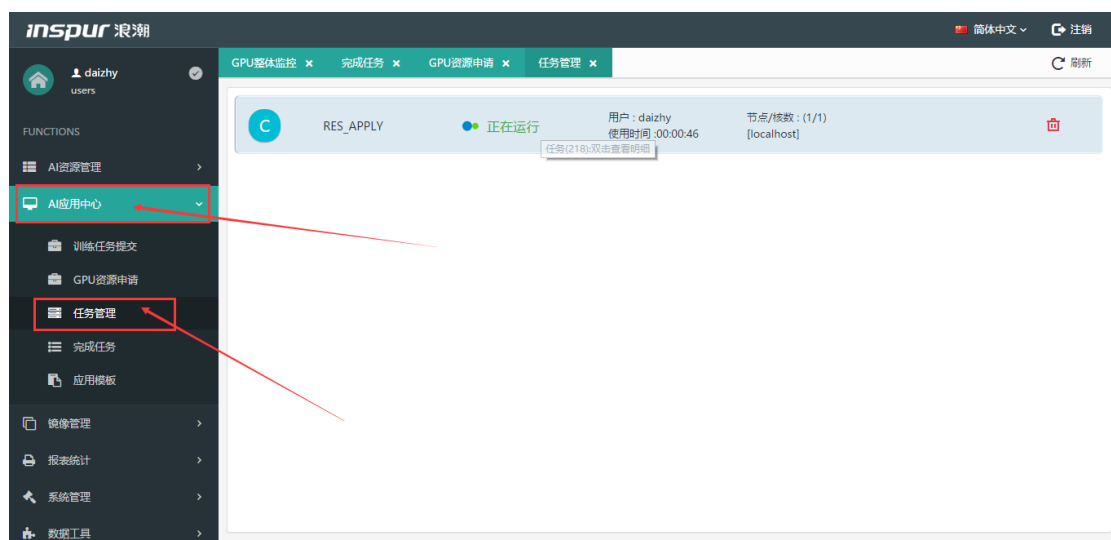
h 数据卷：

主机路径（要共享到容器中的主机路径）容器路径（共享主机路径映射到容器中的路径），例如主机路径是:/opt/docker/caffedata, 容器路径:/usr/local/data

i 最后点击创建，提示创建成功的任务号。例如 380

2)使用容器

1 在任务管理中查看已创建的容器。双击容器，查看详细信息



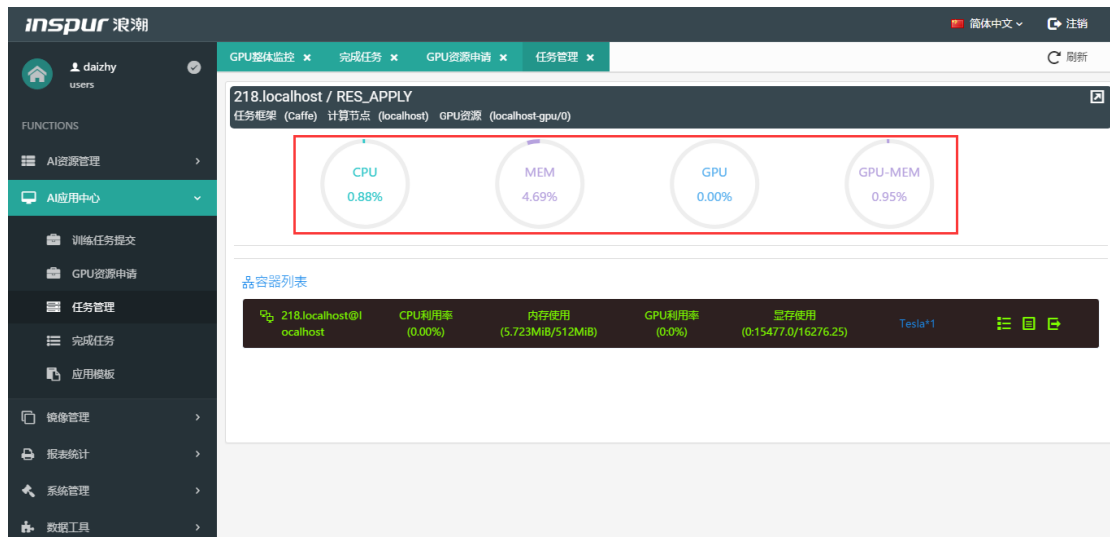
2 双击容器，查看详细信息



第一栏列出了节点的基本信息

点击右上小方格返回上级显示界面

3 中间一栏列出了当前节点的资源占用情况



4 最下一行显示了容器的具体资源占用情况。后面三个按钮分别为容器的详情，容器日志，远程 shell 登录容器。



5 点击 shell 登录打开网页版 shell 窗口如下图：在该 shell 中可以对容器进行命令行的操作，可以训练自己的任务和其他操作。

```

Login[root@10.166.15.189]:
The authenticity of host '10.166.15.189 (10.166.15.189)' can't be established.
ECDSA key fingerprint is 31:0c:b0:94:6e:45:c5:86:9b:6e:46:83:6b:bc:34.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/.ssh/known_hosts).
root@10.166.15.189's password:
Last login: Fri Jul 14 04:16:29 2017 from localhost
[root@localhost ~]# ls
1          dockercaffe.tar.gz  frame_acc.frm          maui-master          mauiScripts
backup.sh  dockermxnet.tar.gz          infinibandlib.tar.gz  maui-master.tar.gz  &?%??
[root@localhost ~]# cd maui-master
[root@localhost maui-master]# ls
CHANGELOG  contrib  include          Makefile.in          Patch-for-gpu-GRES.patch  src
configure  docs     LICENSE          maui.cfg.dist.in     README
configure.ac  etc     LICENSE.mcompat  maui-private.cfg.dist.in  README_patch
[root@localhost maui-master]# cd include/
[root@localhost include]# sl
-bash: sl: command not found
[root@localhost include]# ls
Makefile.inc.ll.in  maui.cfg.dist.inc.nm.in  mg2.h          mprof-proto.h          msched-version.h
Makefile.inc.lsf.in  maui.cfg.dist.inc.rm.in  mg2-proto.h    msched-common.h        msu.h
Makefile.inc.mx.in   mclient.h               moab.h         msched-common-proto.h  msu-proto.h
Makefile.inc.pbs.in  mclient-proto.h         moab-local.h.in  msched.h
Makefile.inc.pcre.in  mcom.h                 moab-proto.h    msched-llapi.h
Makefile.inc.sdr.in  mcompat-proto.h         moab-s_apps.h   msched-pbsapi.h
Makefile.inc.sge.in  mcom-proto.h           moab-wiki.h     msched-proto.h
[root@localhost include]#

```

第十章 应用模板

用户通过应用模板模块，可以针对各种应用自定义不同的作业提交页面。应用模板的设置包含两个部分：页面布局及作业脚本。用户通过页面布局确定该应用的作业提交选项及输入方式；通过作业脚本设置应用的运行及如何使用作业提交页面中输入的参数。新建成功的应用模板，将会显示在作业提交模块的作业列表中。用户在作业提交页面中，选择已设置好的作业模板，输入模板中设定的作业参数，即可将输入参数传递给对应的应用脚本，并提交该应用脚本。

10.1 新建作业模板

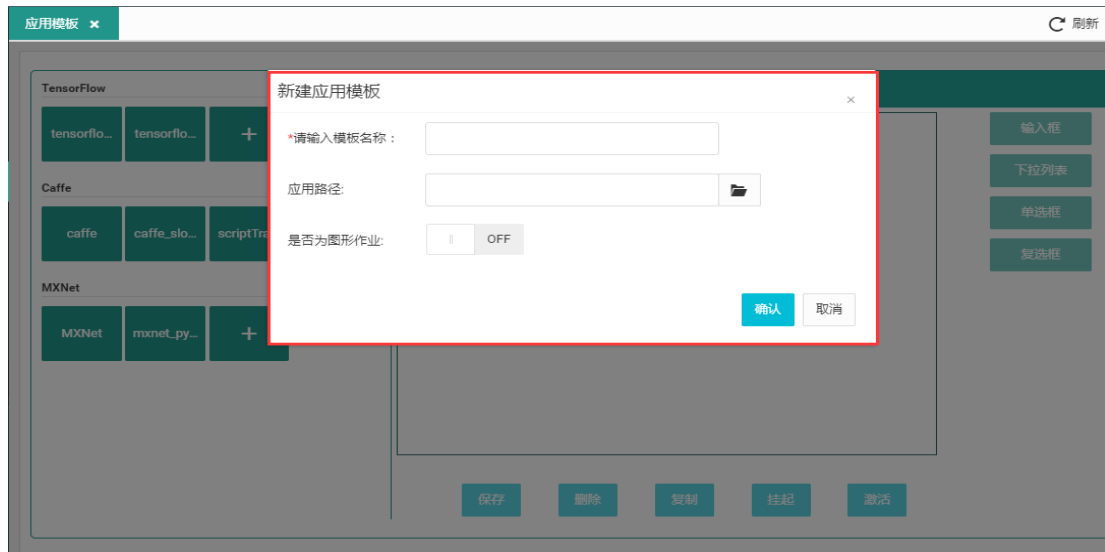
根据用户的应用需求可以灵活配置不同的应用提交模板，下面具体说明参数的作用。

应用名称：输入要新建的应用模板名称，只能是英文字母。

应用路径：输入要新建的模板的应用程序安装全路径。

图形作业：是否需要 VNC 查看作业图形界面功能。

在应用模板界面输入应用模板名称，对应应用程序的安装路径，选择是否是图形作业，点击“确定”按钮，即可新建一个应用模板。新建的应用模板将会显示在左侧的应用模板列表栏中。



新建的应用模板默认为被选择状态，接下来设置该应用模板的页面布局 and 作业脚本

1) 设置作业模板布局-添加一个输入框

如下图，参数说明：

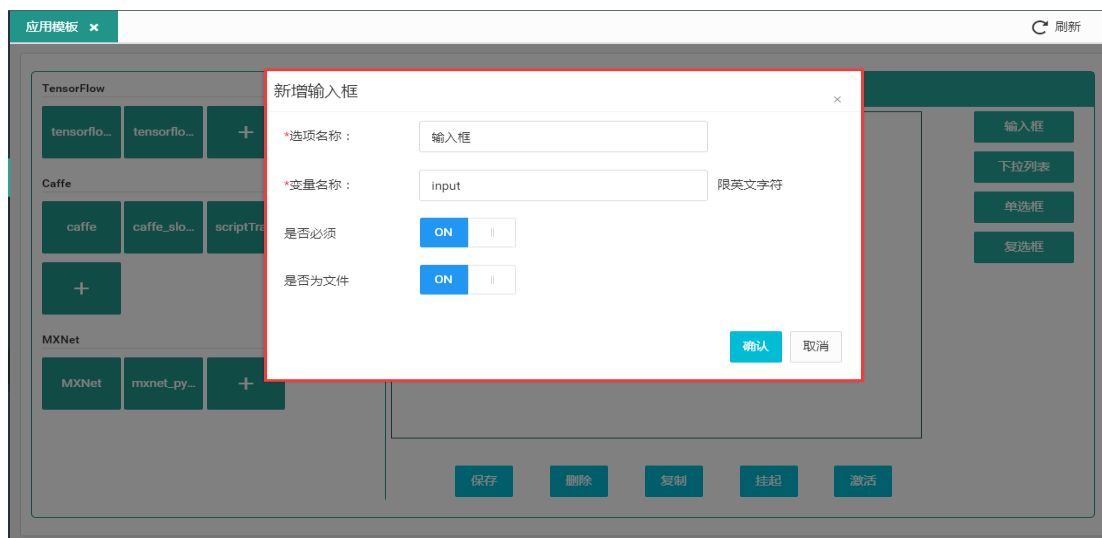
选项名称：作业提交选项的名称

变量名称：作业提交选项对应的变量，该变量将接受用户输入传递给作业脚本

是否必须：该选项是否是作业提交的必填选项

文件选择：该选项是否需要文件选择器

在应用模板页面右侧操作栏中点击“输入框”，将弹出添加输入框页面，在该页面中输入选项名称，变量名称，选择是否是必须选项，是否需要文件选择功能，点击确定按钮，将给指定应用模板添加一个输入框选项



添加成功后，如下图所示：



2) 设置作业模板布局-添加一个下拉列表

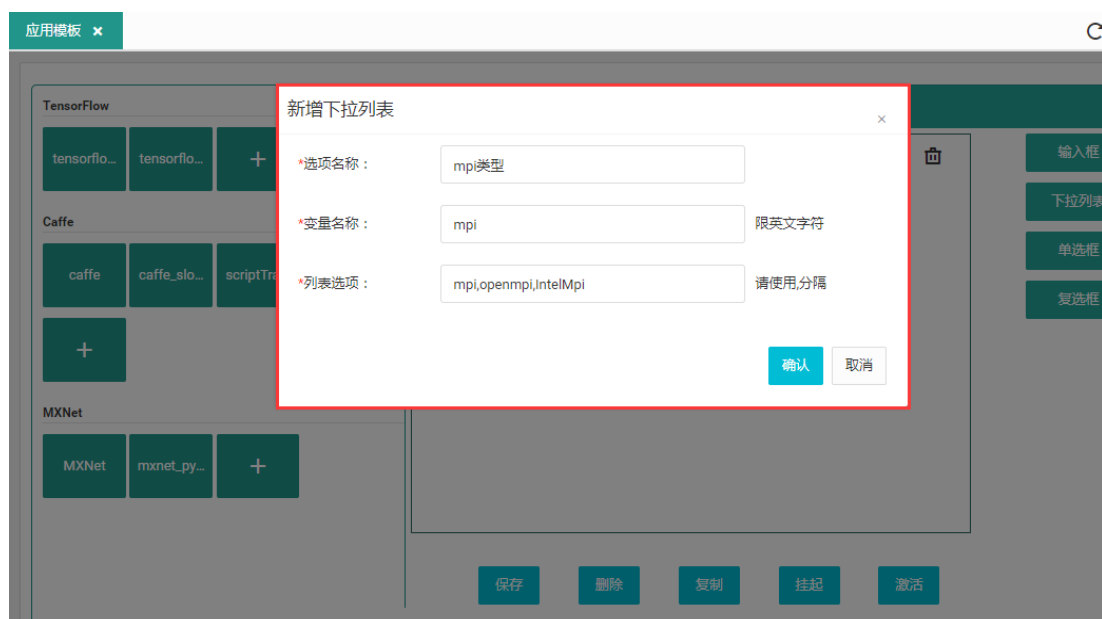
如下图，参数说明：

选项名称：作业提交选项的名称

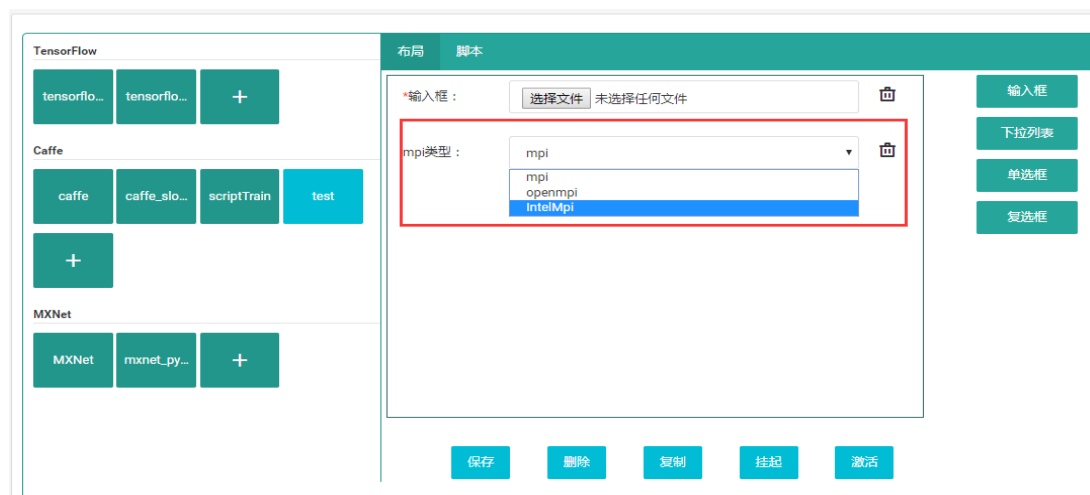
变量名称：作业提交选项对应的变量，该变量将接受用户输入传递给作业脚本

列表选项：下拉列表提供的下拉选项，使用英文逗号分开”，”（注意，不识别中文逗号”，”）

在应用模板页面右侧操作栏中点击“下拉列表”，将弹出添加输入框页面，在该页面中输入选项名称，变量名称，在列表选项中输入所要提供给用户选择的下拉列表选项（建议将默认下拉选项写在最前面），点击确定按钮，给指定应用模板添加一个下拉框选项



添加下拉列表后，页面布局显示如下图：



3) 设置作业模板布局-添加一个单选框

如下图，参数说明：

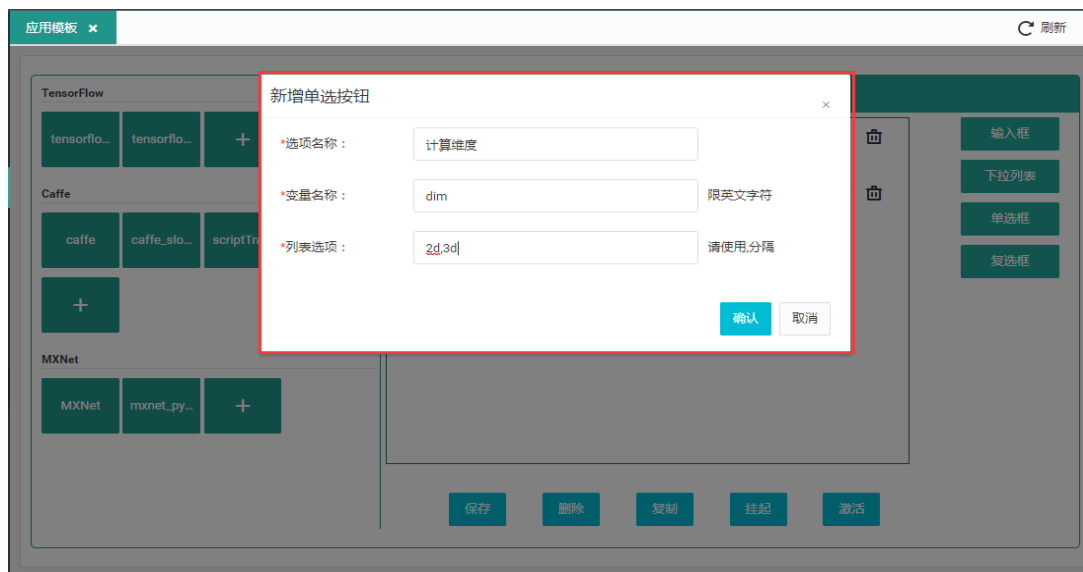
选项名称：作业提交选项的名称

变量名称：作业提交选项对应的变量，该变量将接受用户输入传递给作业脚本

列表选项：单选框列表提供的单选选项，使用英文逗号分开”，”（注意，不识别中文逗号”，”）

在应用模板页面右侧操作栏中点击“下拉列表”，将弹出添加输入框页面，在该页面中输入选项名称，变量名称，在列表选项中输入所要提供给用户选择的下拉列表选项，点击确定按钮，给指定应用模板添加一个下

拉框选项。



成功添加单选框后，将在布局页面显示



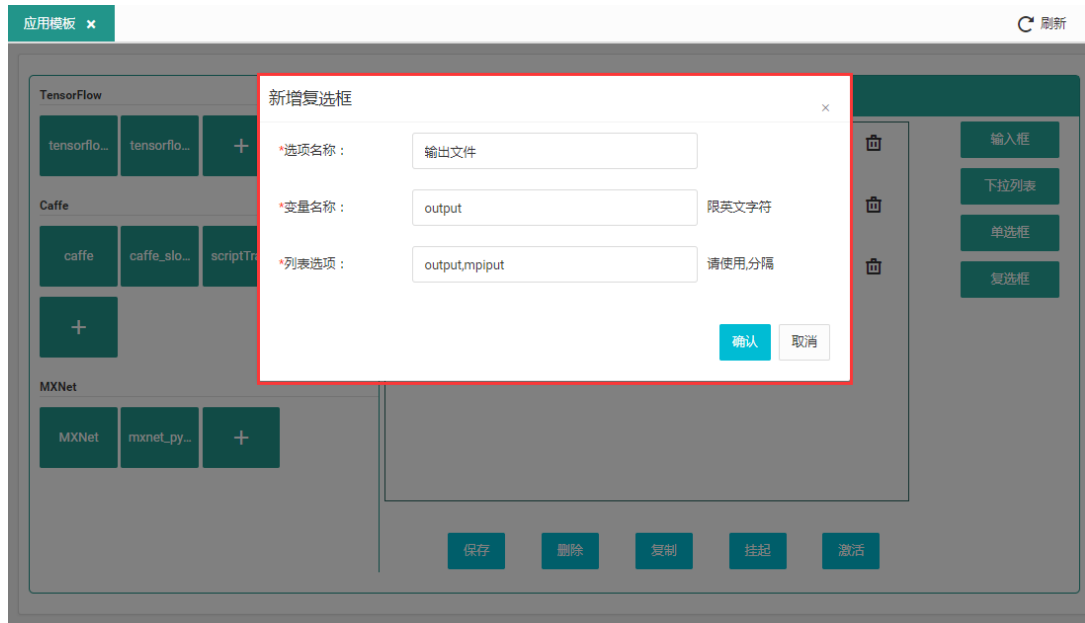
4) 设置作业模板布局-添加一个复选框

如下图，参数说明：

选项名称：作业提交选项的名称

变量名称：作业提交选项对应的变量，该变量将接受用户输入传递给作业脚本

列表选项：复选框列表提供的复选选项，使用英文逗号分开”，”（注意，不识别中文逗号”，”）



添加复选框列表成功后，布局页面如下图：



5) 设置作业模板脚本

在应用模板页面切换至作业脚本视图，如下：



前面布局页面中添加的变量都会以#参数名称: xxx 变量名称\$xxx 形式列出。

用户在页面（如输入文件，MPI 类型）中输入或选择的参数，将会以\$xxx（对应的为\$input,\$mpi）传递给脚本。然后，脚本中可以直接将这些参数传递给应用程序

例如：

```
/opt/fluent -i $input -mpi $mpi $dim
```

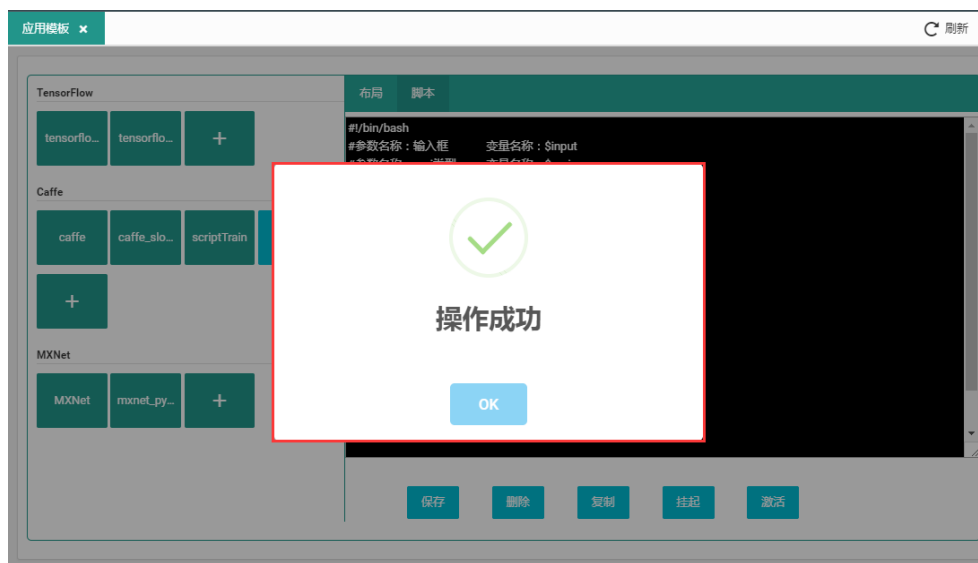
对于复选框的参数使用需要稍作处理，用户在页面选择的复选框将以（output-errput）形式传递给应用脚本的对应参数。如下：

```
output=`echo $output | cut -d '-' -f1`
```

```
errput=`echo $output | cut -d '-' -f2`
```

10.2 保存作业模板

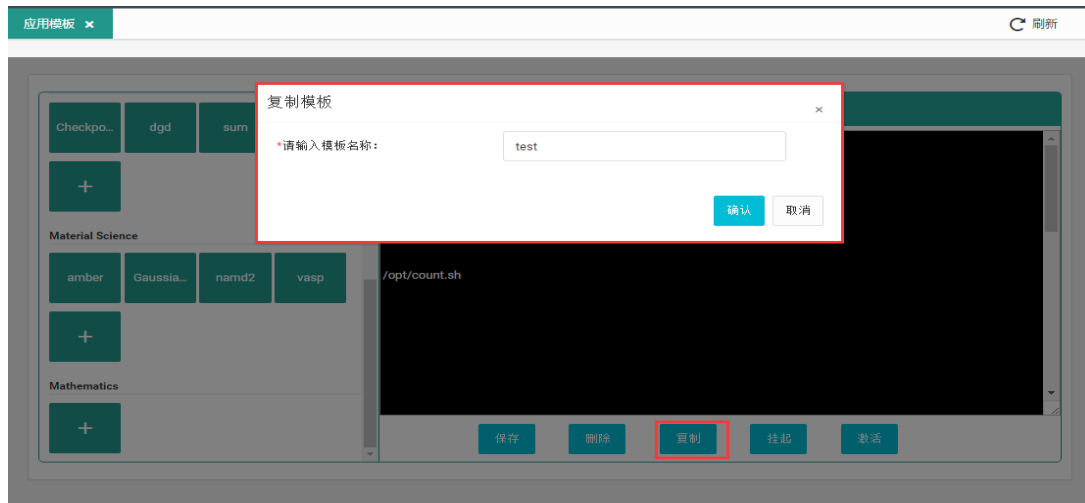
应用模板的布局及脚本设置好之后，点击保存按钮，应用模板完成创建，如下图



10.3 复制应用模板

在应用模板页面，选择一个应用模板，点击“复制”按钮，输入要复制的新模板名称，可以将所选应用模板，复制一个新的应用模板。新的应用模板将于所选应用模板有同样的布局 and 脚本设置。对于功能相近的应用模板，可以采用此操

作快速在已有模板基础上复制一个，然后根据模板需要进行修改。

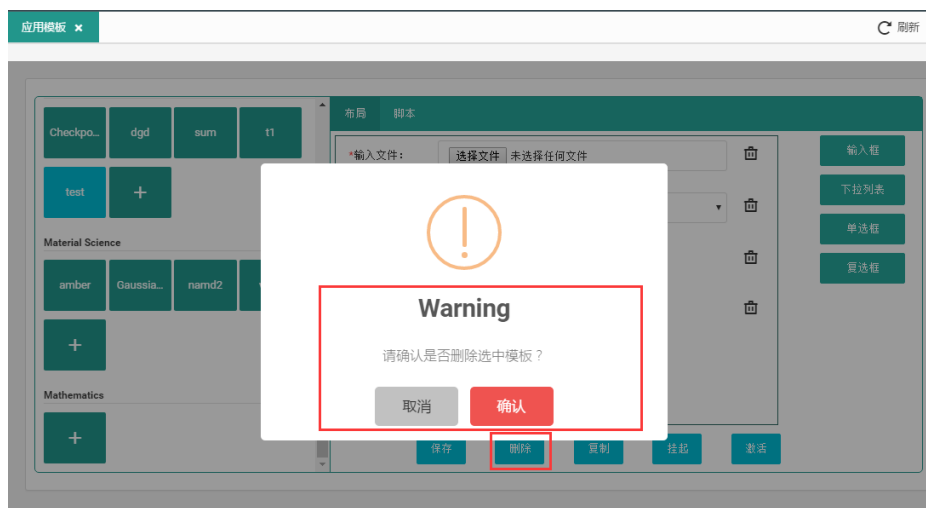


复制完成后，在左侧应用模板列表中，显示新复制的应用模板，如下：



10.4 删除应用模板

在应用模板页面，点击应用模板，点击“删除”按钮，确认删除选择”yes”，可以删除一个应用模板

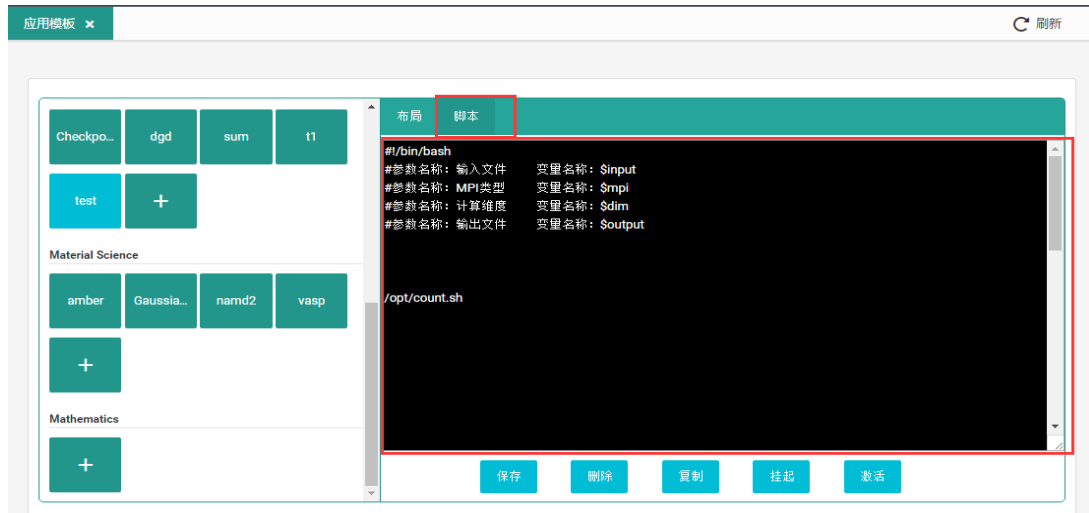


10.5 编辑应用模板

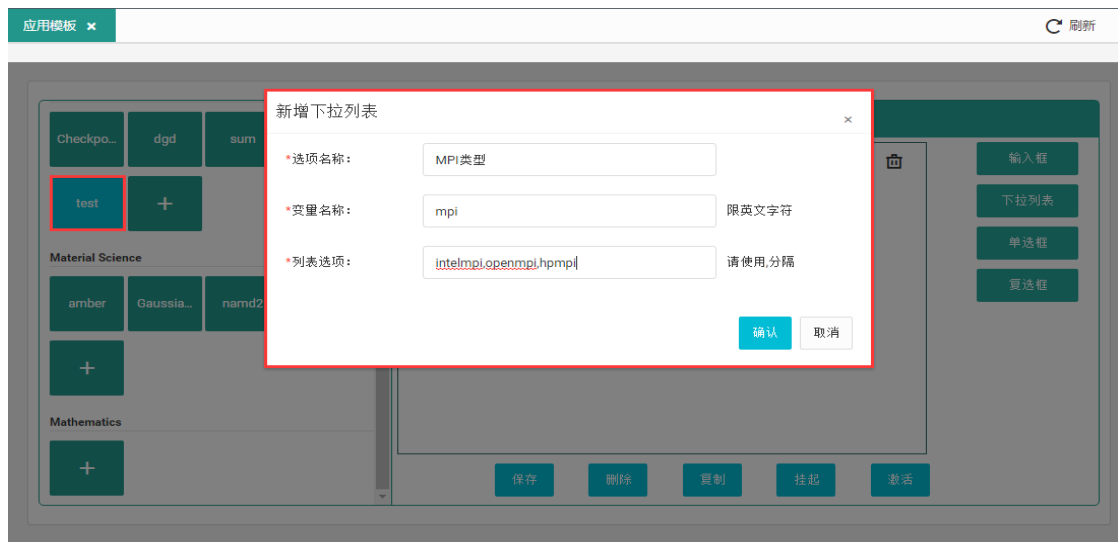
进入应用模板后，在左侧应用模板栏中选择已设置的应用模板，在右侧布局栏中将出现该应用模板的布局设置；如下图：



点击切换到脚本栏，将显示所选应用的脚本配置，如下图：



可以对已设置的应用模板进行编辑，包括添加新参数，或者删除已有参数。如下图，给 test 应用模板添加一个下拉列表。



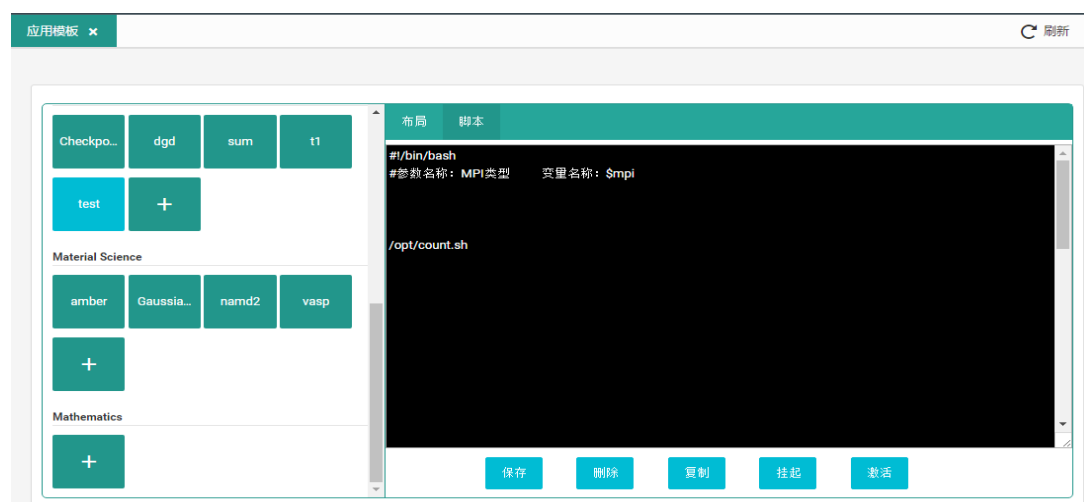
另外，可以将已添加参数删除。只需点击参数后面对应的删除图标即可，如下图：



删除后布局页面如下图所示：

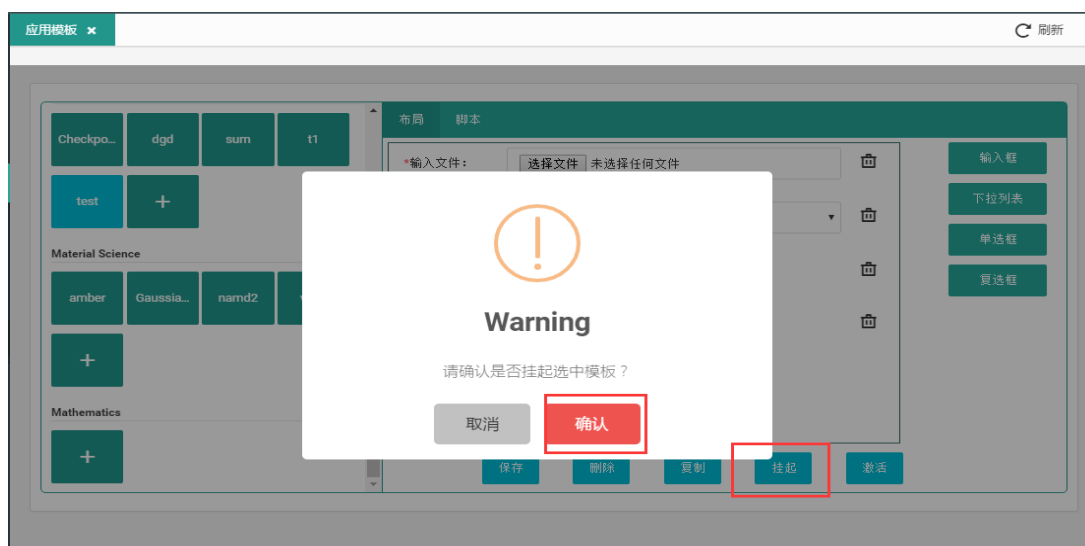


同时，对应脚本中的参数也被删除：



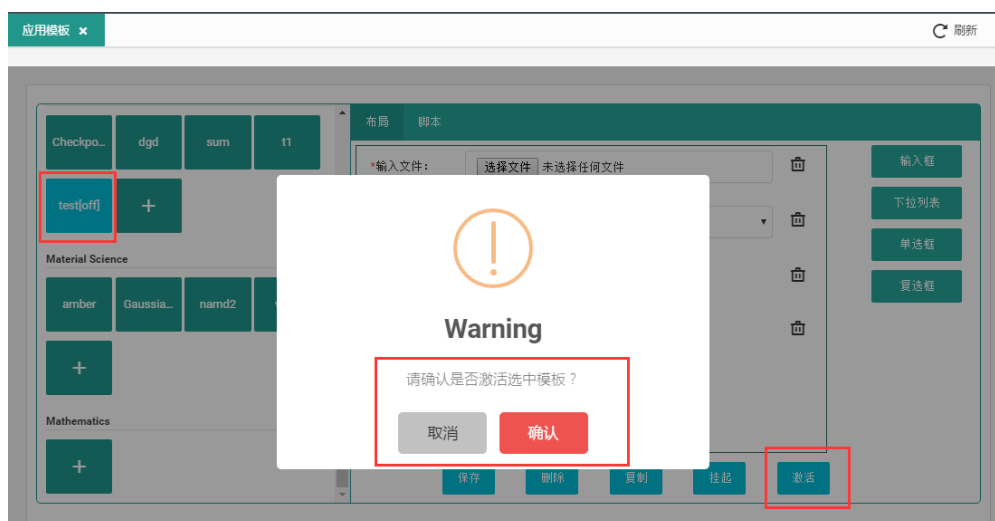
10.6 挂起应用模板

在应用模板页面，点击应用模板，点击“挂起”按钮，，确认挂起选择”yes”，可以挂起一个应用模板。



10.7 激活应用模板

在应用模板页面，点击应用模板，点击“激活”按钮，，确认激活选择”yes”，可以激活一个应用模板

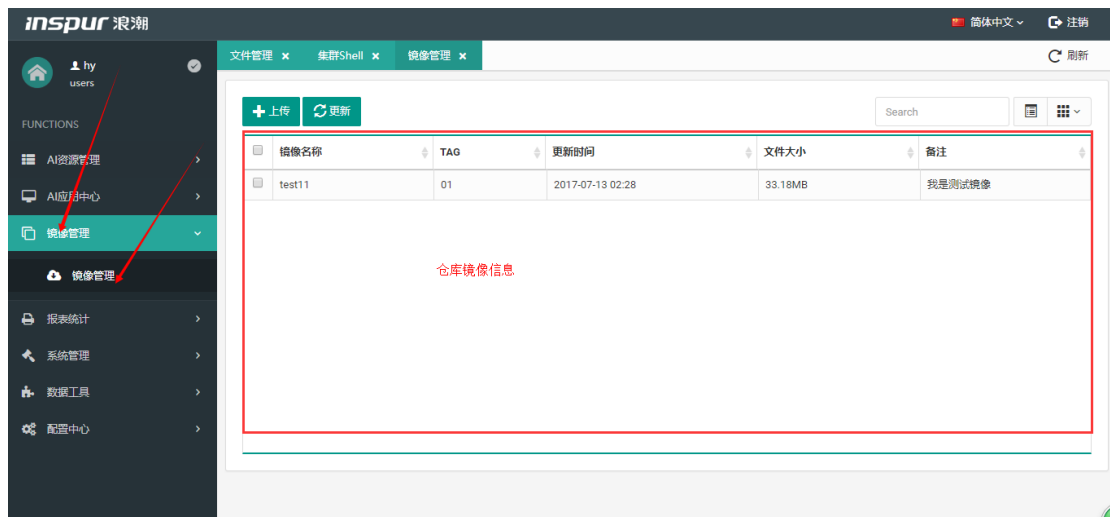


第十一章 镜像管理

镜像管理模块可以查看私有镜像仓库的镜像信息。可以将集群本地某个节点镜像放在私有仓库进行保存，可以在集群中下载镜像仓库中的镜像。方便快捷的管理集群中的镜像。

11.1 查看私有仓库镜像

打开镜像管理中的镜像管理模块，查看仓库中的镜像信息。包括镜像名称，tag，更新时间，文件大小，备注。

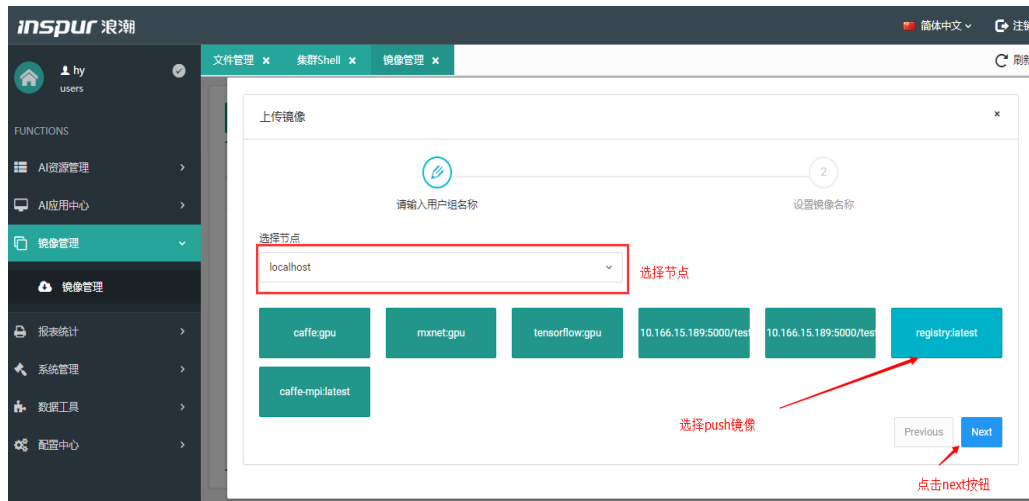


11.2 上传镜像到私有仓库

1) 点击上传按钮



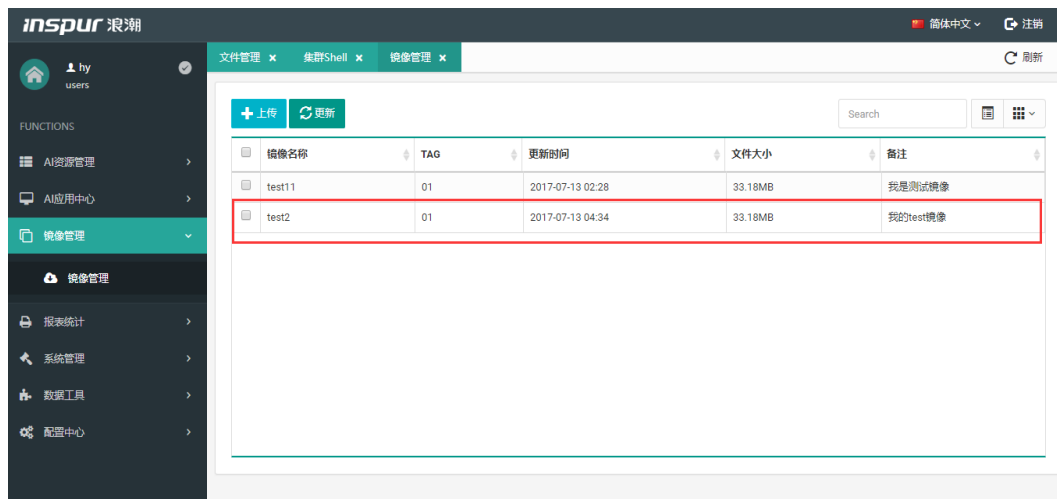
2) 选择节点信息，镜像名称，选择结束后，点击 next 按钮



3) Enter the warehouse information, and the image warehouse information includes: Private warehouse address, port number, upload image name, image tag, image remarks.



4) Image successfully uploaded to private warehouse

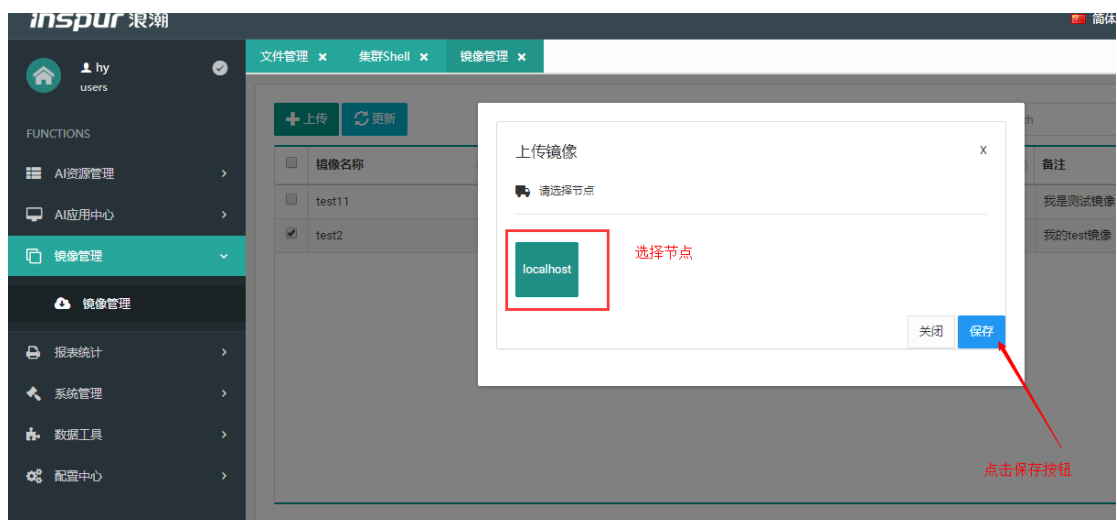


11.3 从私有仓库 pull 镜像

1) 选择某个镜像，点击更新按钮



2) 选择某个节点，点击更新保存



3) 镜像成功下载到本地

