

# Deep Learning(CS280) Tutorial:

## SIST AI-Cluster

---

2021/09/27

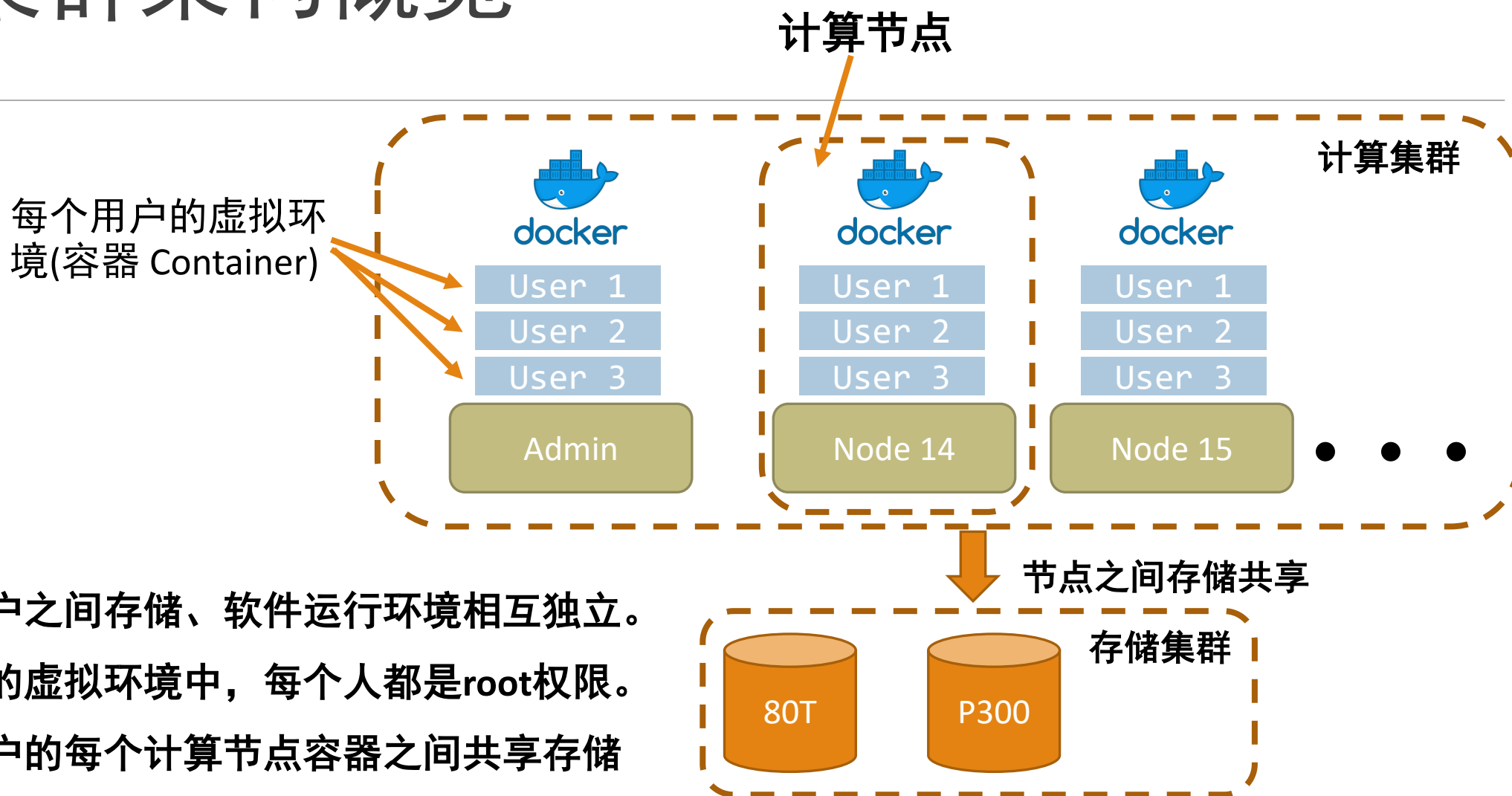
王立翹

# Overview

---

- 集群架构概览
- 登陆
- 远程调试代码

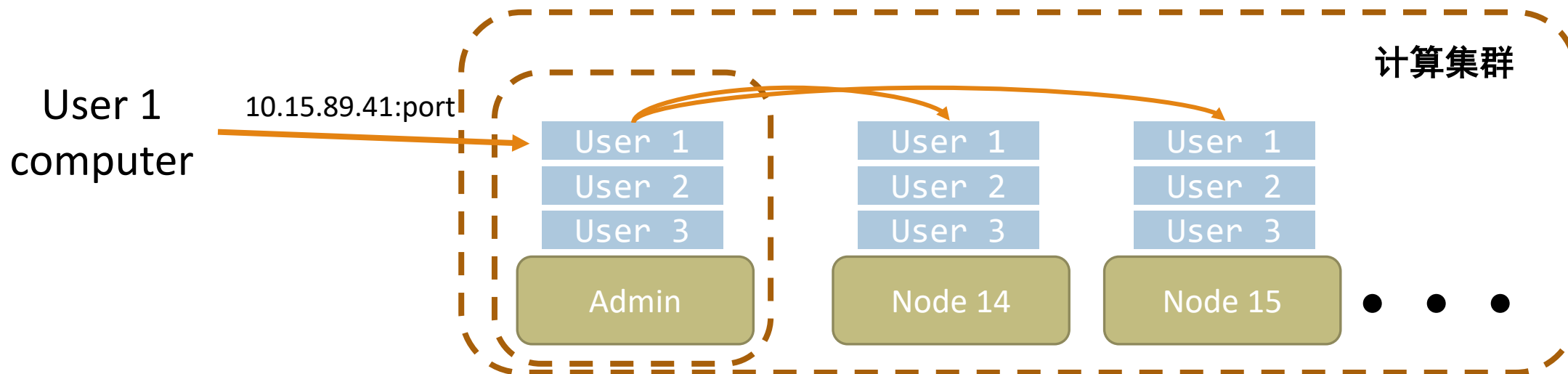
# 集群架构概览



- 每个用户之间存储、软件运行环境相互独立。
- 在自己的虚拟环境中，每个人都是root权限。
- 单个用户的每个计算节点容器之间共享存储

# 网络架构

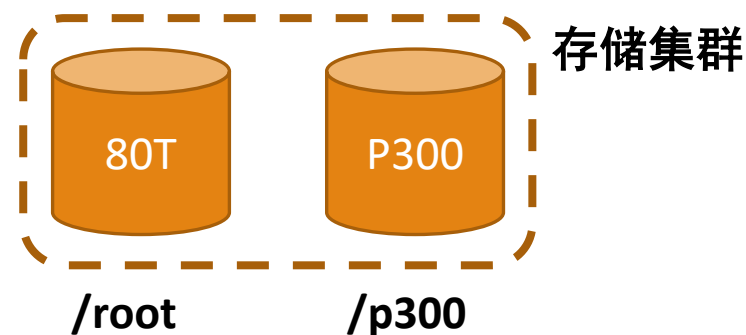
要访问其他节点，使用命令  
ssh nodeXX



- 在连接时不同用户的容器时，以端口进行区分。
- AI 集群所有节点仅有admin 节点与外网相连，访问其他节点需通过admin节点。
- 其他节点访问外网需使用代理(已配置好)

# 存储集群

- 80T:
  - 集群的虚拟环境、监控程序的数据
  - 用户的 /root 文件夹
- P300
  - 分布式存储集群
  - 用于存储高IO的数据
- 注意事项
  - 80T空间，带宽调度的策略都不如p300好。像模型的checkpoint, training data这些东西请放到P300中去，使用soft link将p300的文件挂载到root目录的文件夹内：
    - `ln -s /p300/<somefolder> <the_target_mount_point>`



# 计算资源--软件

---

- 集群已有环境(GPU 计算)
  - Ubuntu 16.04 , Nvidia Driver Version: 418.74 (不支持用户自定义)
  - CUDA Version: 9.0, gcc/g++ Version 5.4.0
  - shell: zsh /w Oh-My-ZSH plugin (因此shell的config file是~/.zshrc 而不是~/.bashrc )
- Python环境
  - 自带 python 2.7, python 3.6
  - 自行安装anaconda (推荐) - wget 安装脚本一键无脑安装即可
- 安装环境时...
  - 建议在有显卡的计算节点进行安装
  - apt 或者 自行编译安装
  - 不需要su/sudo, 每个用户都是root权限

# 计算资源 – 硬件

---

- CPU-memory
  - Admin: 56 cores Intel Xeon E5-2690 CPU, 251GB
  - Node13,14: 28 cores Intel(R) Xeon(R) CPU E5-2690 v4, 251GB
  - Node25,26: 48 cores Intel(R) Xeon(R) CPU E5-2697 v2 , 377GB
- GPU:
  - Admin: 无
  - Node13,14: Tesla K80 /w 11GB x 8
  - Node25,26: Tesla K40M /w 11GB x 8
- 虽然Admin节点看起来很厉害，但绝对不要在admin节点跑任何计算程序！！

# Let's login!

如何获取自己的账号：

- [http://10.15.89.41:8899/cs280\\_permission](http://10.15.89.41:8899/cs280_permission)

- 重要信息：

- CONTAINER\_PORT：登录计算节点的ssh 端口
- ADMIN\_PORT：为每个用户分配的用于承载其他功能的端口
- PERMISSION NODES：可用节点

Home User Permission Discuss Doc AI Status P40 Status SVIP GPU Status PLUS GPU Status							
UID	USERNAME	CHINESE NAME	EMAIL	ADVISOR	CONTAINER_PORT	ADMIN_PORT	PERMISSION NODES
1033	liuyf	刘永飞	liuyf3@shanghaitech.edu.cn	何旭明	22033	30330-30339	<div>admin</div> <div>node03</div> <div>node04</div> <div>node15</div> <div>node16</div> <div>node17</div> <div>node28</div> <div>node31</div> <div>node32</div>
1061	yanshp	颜世鹏	yanshp@shanghaitech.edu.cn	何旭明	22061	30610-30619	<div>admin</div> <div>node03</div> <div>node04</div> <div>node17</div> <div>node21</div> <div>node22</div> <div>node34</div> <div>node35</div>



# Login

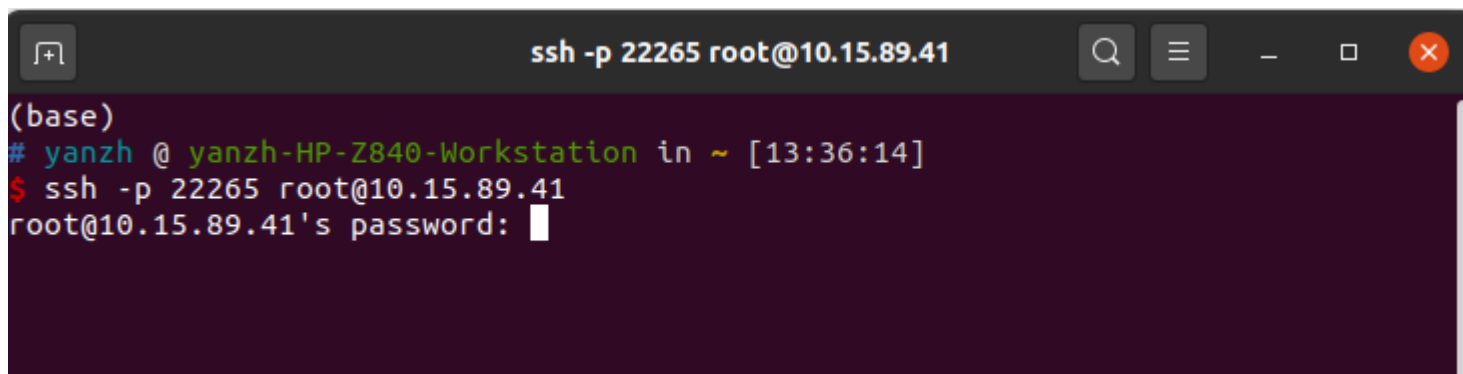
---

- 集群ip为校园网内网，登录时本机IP需要是学校IP.
- 用户名: root 初始密码: sist 请尽快使用passwd修改密码
- Login by ssh:
  - [Putty](#) , termius, MobaXterm (Windows)
  - 建议Connection – Seconds between keepalive (0 to turn off) 填上 60 之类的时间值，防止长时间不操作掉线

# login

---

- ssh [root@10.15.89.41](#) -p CONTAINER\_PORT (UNIX)
- 用户名: root 初始密码: sist 请尽快使用passwd修改密码

A terminal window with a dark background. The title bar shows 'ssh -p 22265 root@10.15.89.41'. The prompt is '(base)'. The user 'yanzh' is logged in at 'yanzh-HP-Z840-Workstation'. The command 'ssh -p 22265 root@10.15.89.41' has been entered, and the prompt is now 'root@10.15.89.41's password:'.

```
(base)
# yanzh @ yanzh-HP-Z840-Workstation in ~ [13:36:14]
$ ssh -p 22265 root@10.15.89.41
root@10.15.89.41's password: 
```

# 计算资源--GPU

---

- GPU 资源的监控
  - nvidia-smi (第三方的 glances, gpustat)
  - 集群dashboard 网页监控 <http://10.15.89.41:8899/gpu>
  - fuser -v /dev/nvidia\* 命令(可能需要先安装psmisc: apt install psmisc)
- 指定GPU的使用
  - export CUDA\_VISIBLE\_DEVICES="gpu\_id"
  - 一个自己定义的函数: set\_gpu x
  - 如果不指定, 默认用0~N号卡
  - 由于硬件是大家共享的, 最好手动设置为一个空闲的卡
  - 如果你使用tensorflow, 一定要设置这个variable
- 结束程序
  - pkill -f <进程名>
  - kill -9 <pid>
  - 用可视化htop 监控工具手动终止进程

# AI Cluster上运行Jupyter notebook

---

使用场景：

- 在node7 的端口32658上运行jupyter notebook，需要在浏览器打开jupyter窗口

Step1: 在Admin上开启正向代理

- 命令格式：ssh -NL 10.15.89.41:<Port>:0.0.0.0:<Port> node15  
    <port> 为Admin-port中任意一个未使用的端口号
- 'ssh -NL 10.15.89.41:32658:0.0.0.0:32658 node07'

Step2: 在node7上运行 jupyter notebook

- 'jupyter notebook --allow-root --ip=0.0.0.0 --port=32658'

Step3: 在浏览器访问： '10.15.89.41:32658'

# 计算资源 – 远程调试代码

---

## Visual Studio Code

- 安装remote-ssh 插件
- 添加远程服务器信息
- 连接到服务器，远程修改服务器代码

```
home > yanzh > .ssh > ≡ config
1  # Read more about SSH config f
2  Host jzx_AI_cluster
3      HostName 10.15.89.41
4      User root
5      Port 22265
6
```

## Pycharm远程调试

- <https://cloud.tencent.com/developer/news/221060>

# 计算资源-远程连接

- ssh 连接设置— ssh keep alive

```
vim /etc/ssh/sshd_config  
ClientAliveInterval 30  
ClientAliveCountMax 6
```

- 后台运行程序

- tmux后台运行程序, 关掉当前ssh session的窗口也能保持后台程序的运行, 可以使用鼠标 相关快捷键见教程 <https://www.hamvocke.com/blog/a-quick-and-easy-guide-to-tmux/>
- 使用nohup命令

- 本地与远程文件共享

- Use sshfs(Unix):

```
sshfs -p port root@10.15.89.41:<target-dir> <local_mount_point> -o reconnection -o compression=no -o follow-symlinks -o sync_read -o sshfs_sync
```

- [SFTP Net Drive 2017](#) (Windows include windows sub linux)
- [Pycharm remote Deploy](#)
- [大文件传输使用 scp 命令](#)

# 计算资源-注意事项

---

- 终止程序后要记得检查相关资源是否已经释放
- 绝对禁止admin节点跑程序！！！！
- 禁止恶意占卡

# Q&A

---

有问题欢迎发邮件或者发在piazz上

wangla@shanghaitech.edu.cn