

1. 环境安装配置

1.0 虚拟机安装

VirtualBox安装

从VirtualBox官网(<https://www.virtualbox.org/wiki/Downloads>)下载并依次安装VirtualBox安装包和扩展包。注意：提示是否安装Oracle Corporation通用串行总线控制器时选择 安装。



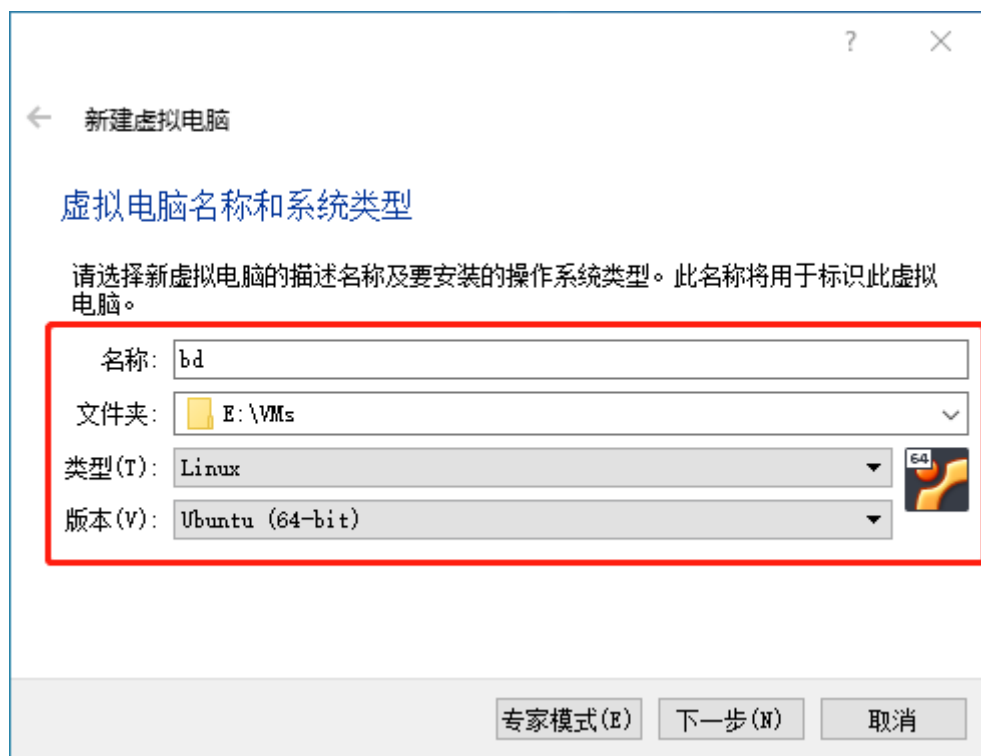
Ubuntu系统安装

从[中国科学技术大学开源软件镜像站](https://mirrors.ustc.edu.cn/ubuntu/)下载Ubuntu 18.04.5操作系统镜像 `ubuntu-18.04.5-live-server-amd64.iso`。

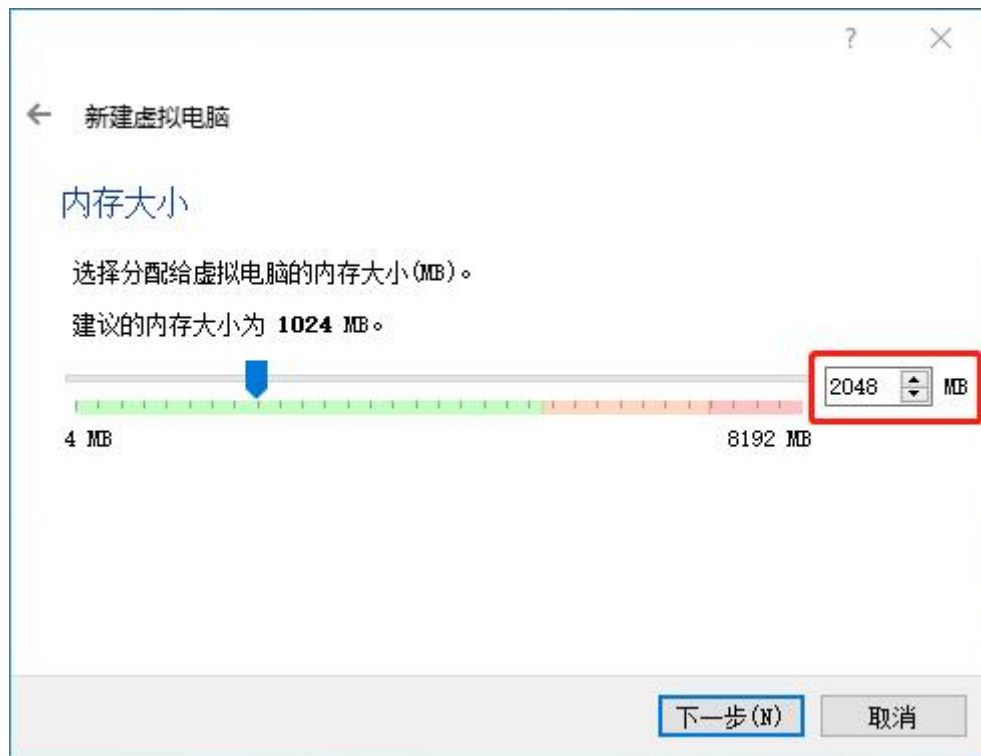
(1) 新建虚拟机



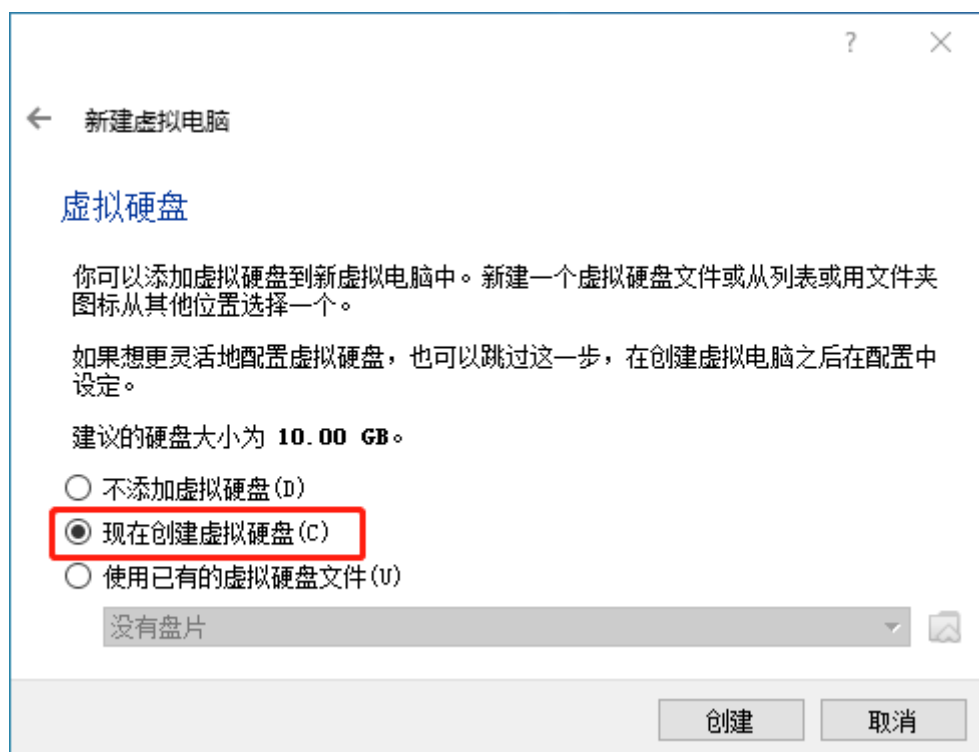
(2) 配置虚拟机名称、存储位置、操作系统类型和操作系统版本

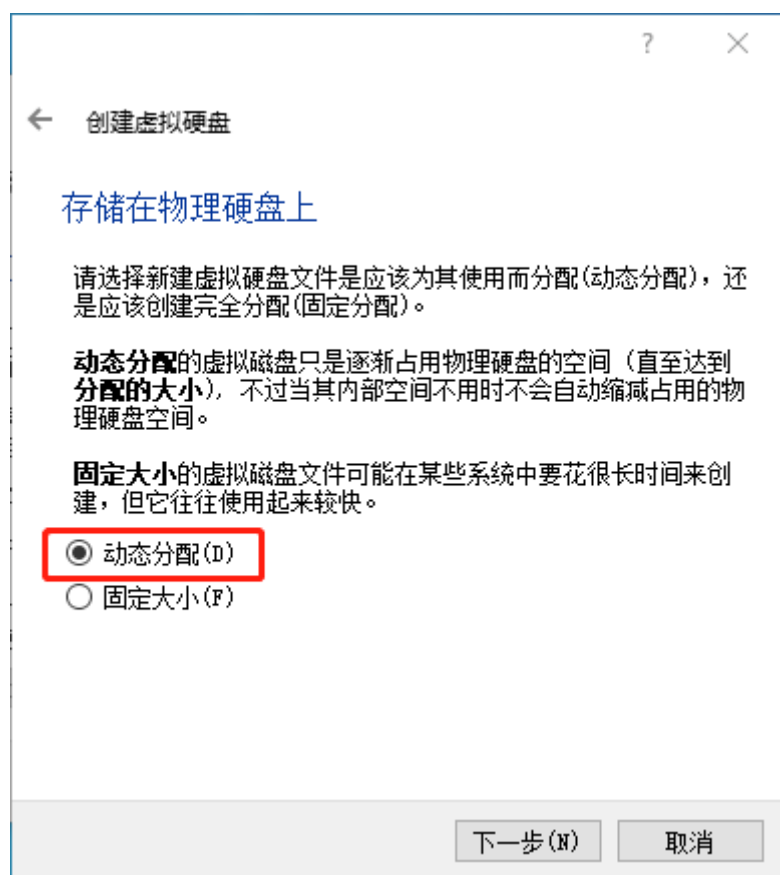
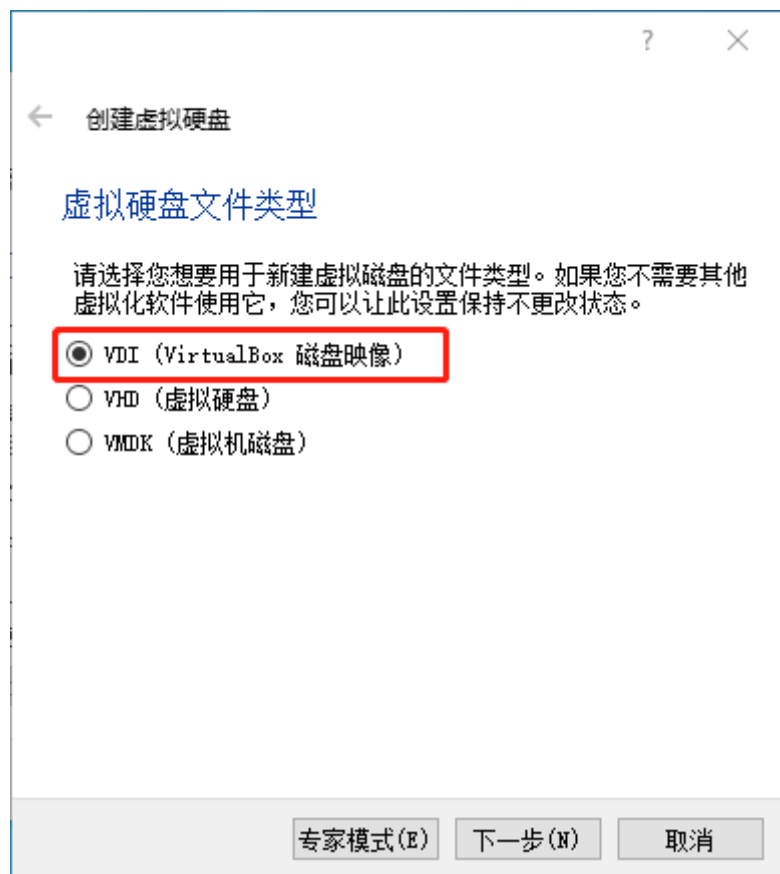


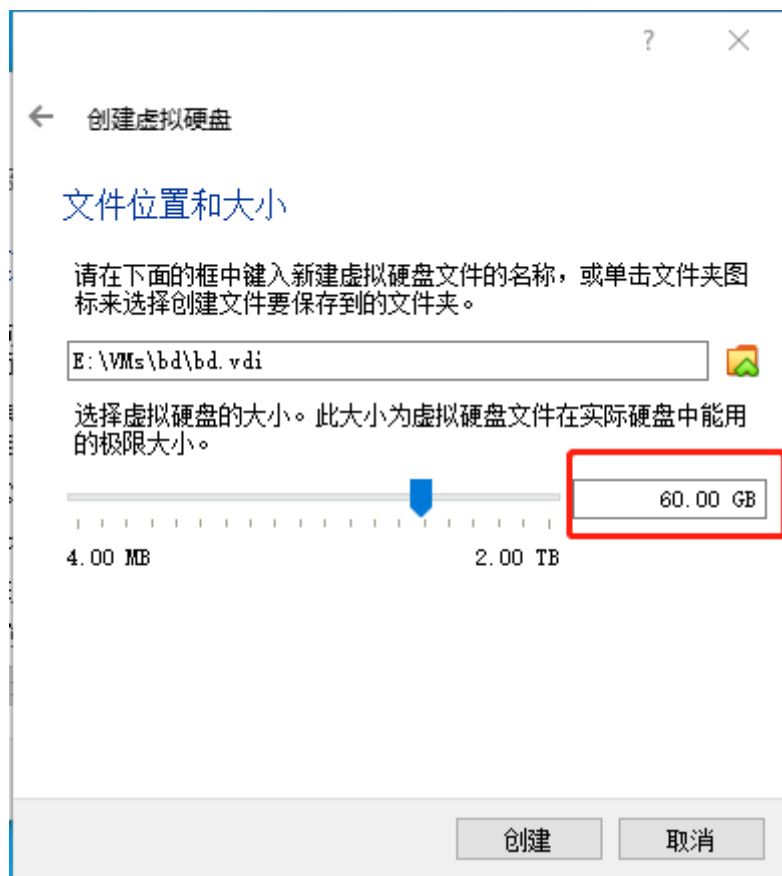
(3) 分配内存（内存大小后期可以进行调整）



(4) 分配硬盘 (注意: 硬盘大小后期不可调整)

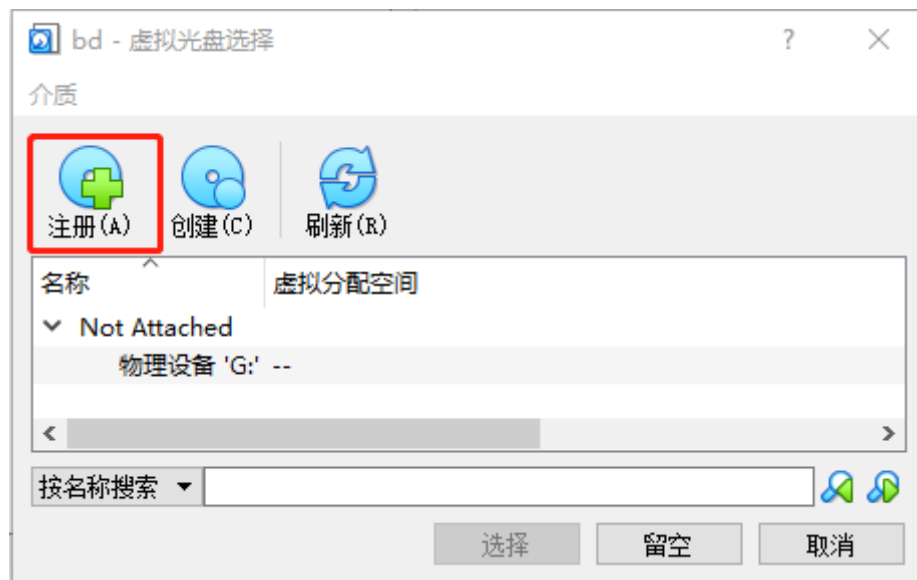
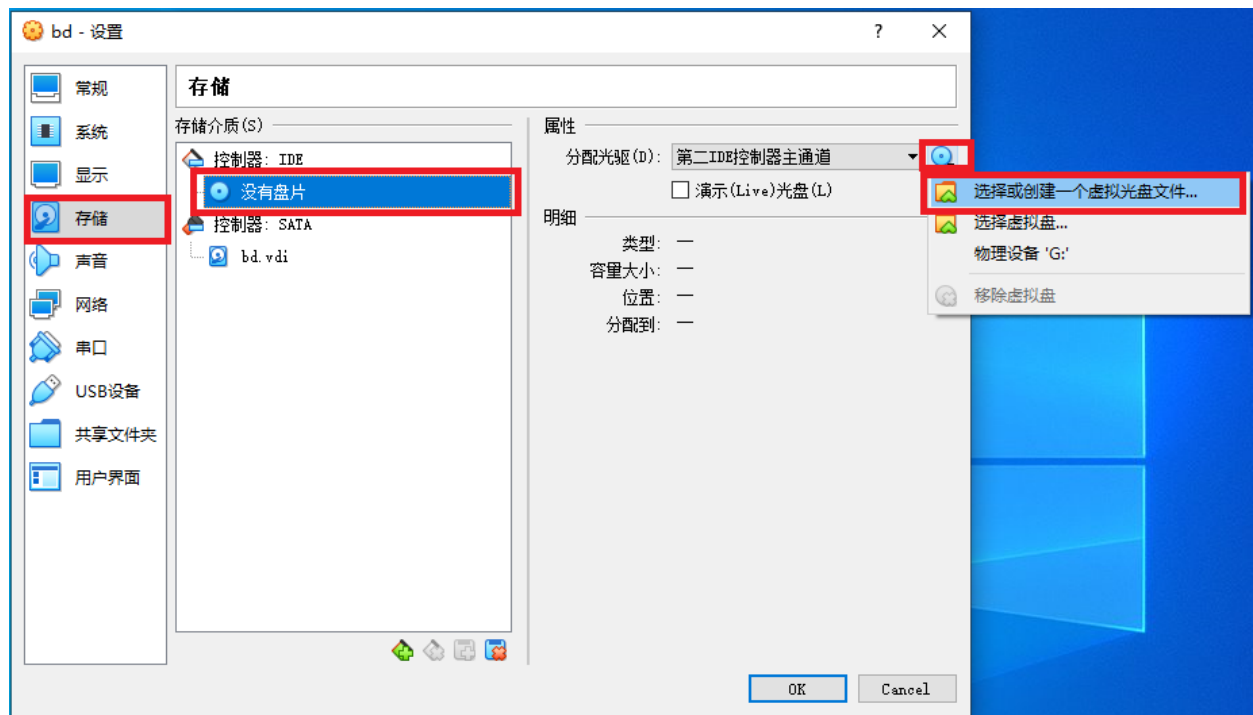


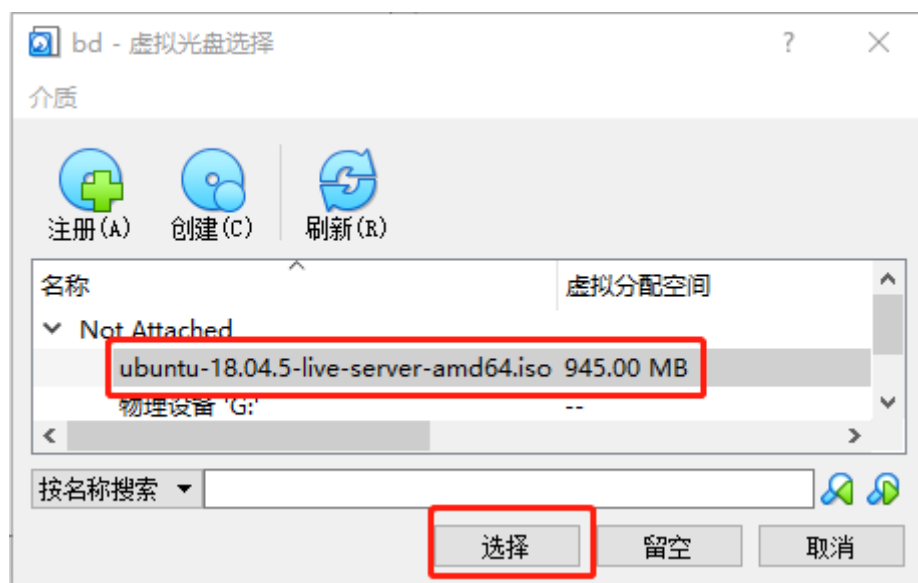
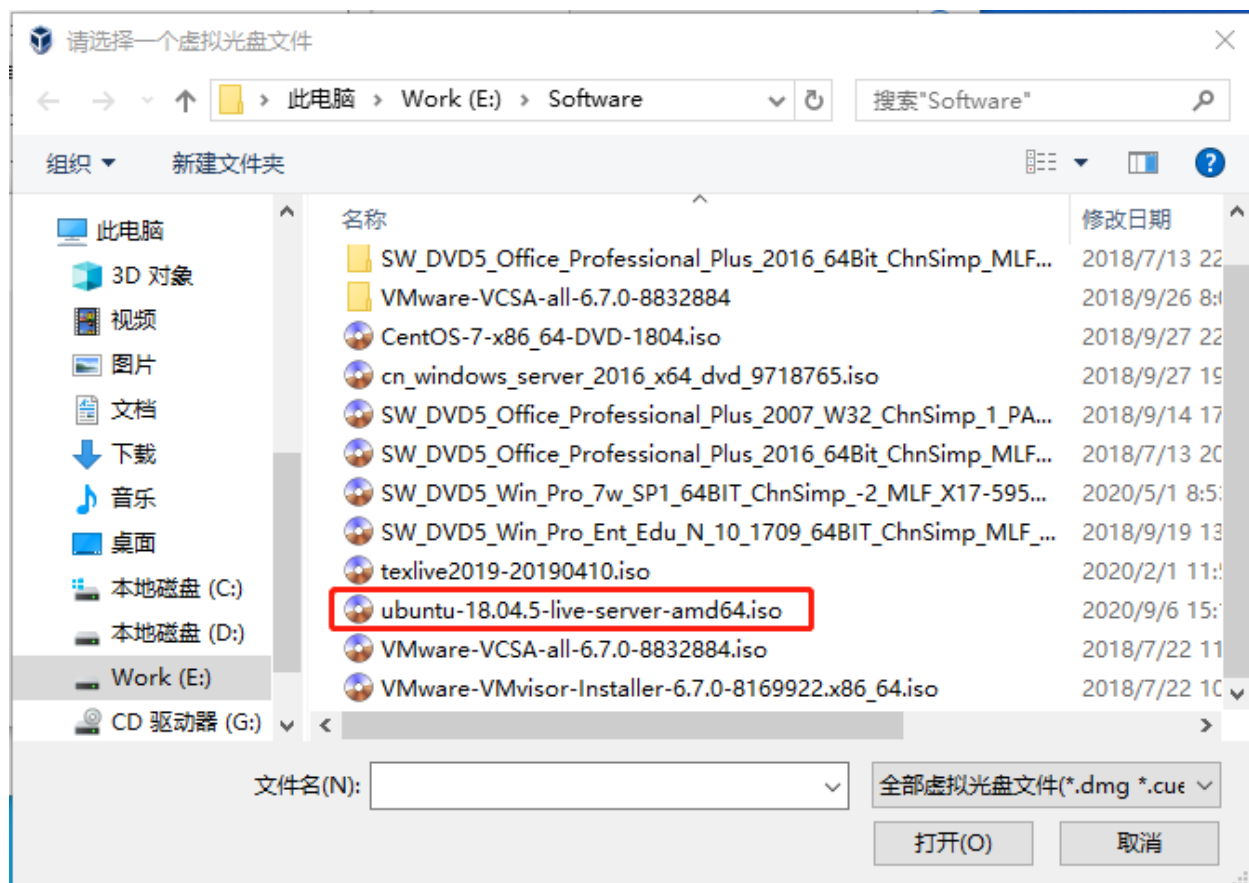




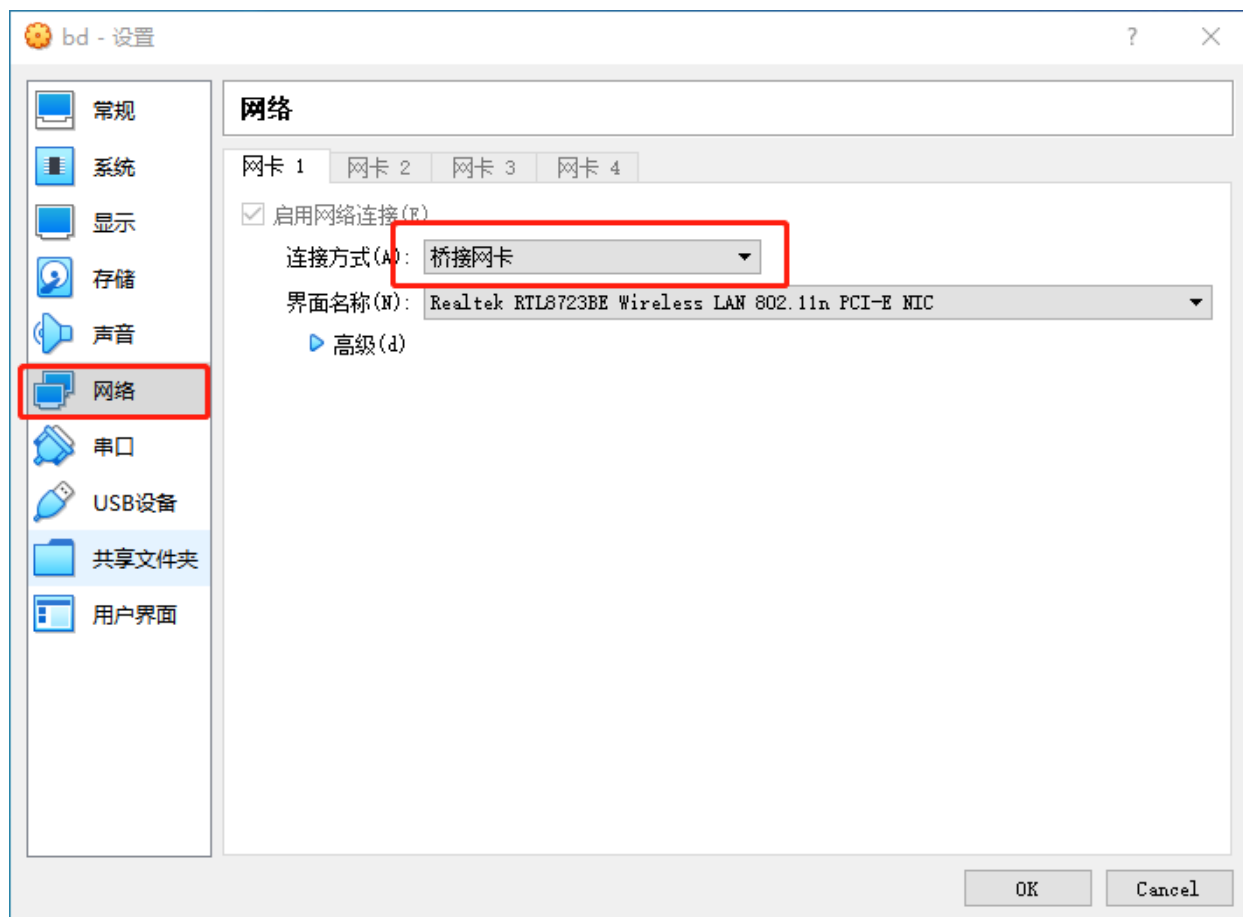
(5) 对虚拟机进行设置







设置网络连接方式为 桥接网卡



(6) 安装Ubuntu 18.04



使用键盘上下键选择语言，回车键确认

Willkommen! Bienvenue! Welcome! Добро пожаловать! Welkom!

[Help]

Use UP, DOWN and ENTER keys to select your language.

[Asturianu	▶]
[Bahasa Indonesia	▶]
[Català	▶]
[Deutsch	▶]
[English	▶]
[English (UK)	▶]
[Español	▶]
[Français	▶]
[Hrvatski	▶]
[Latviski	▶]
[Lietuviškai	▶]
[Magyar	▶]
[Nederlands	▶]
[Norsk bokmål	▶]
[Polski	▶]
[Suomi	▶]
[Svenska	▶]
[Čeština	▶]
[Ελληνικά	▶]
[Беларуская	▶]
[Русский	▶]
[Српски	▶]
[Українська	▶]

Installer update available

[Help]

Version 20.09.1 of the installer is now available (20.07.1+git2.5de9df3e is currently running).

You can read the release notes for each version at:

<https://github.com/CanonicalLtd/subiquity/releases>

If you choose to update, the update will be downloaded and the installation will continue from here.

[Update to the new installer]
[Continue without updating]
[Back]

Keyboard configuration

[Help]

Please select your keyboard layout below, or select "Identify keyboard" to detect your layout automatically.

Layout: [English (US) ▼]

Variant: [English (US) ▼]

[Identify keyboard]

[Done]
[Back]

Network connections

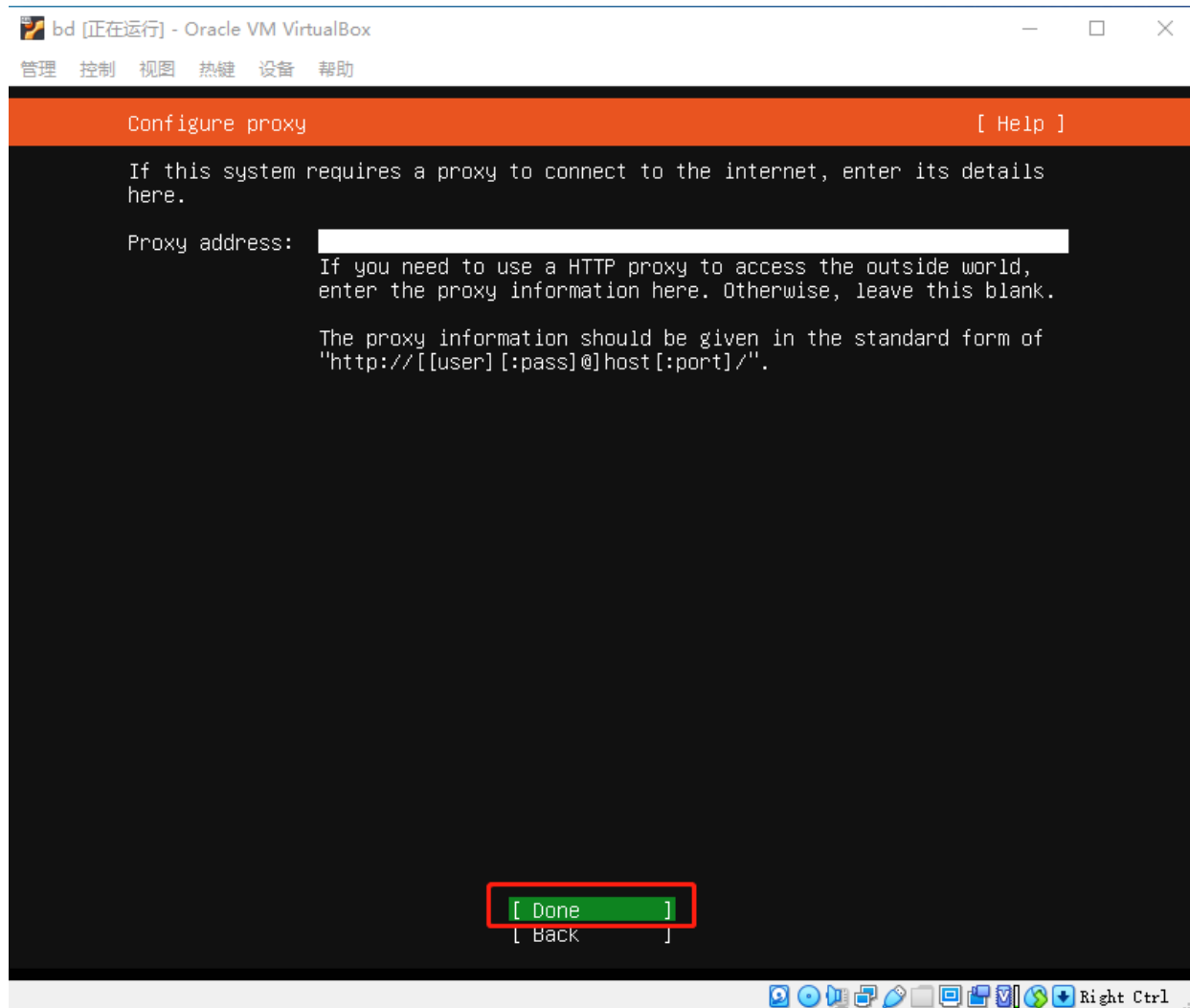
[Help]

Configure at least one interface this server can use to talk to other machines, and which preferably provides sufficient access for updates.

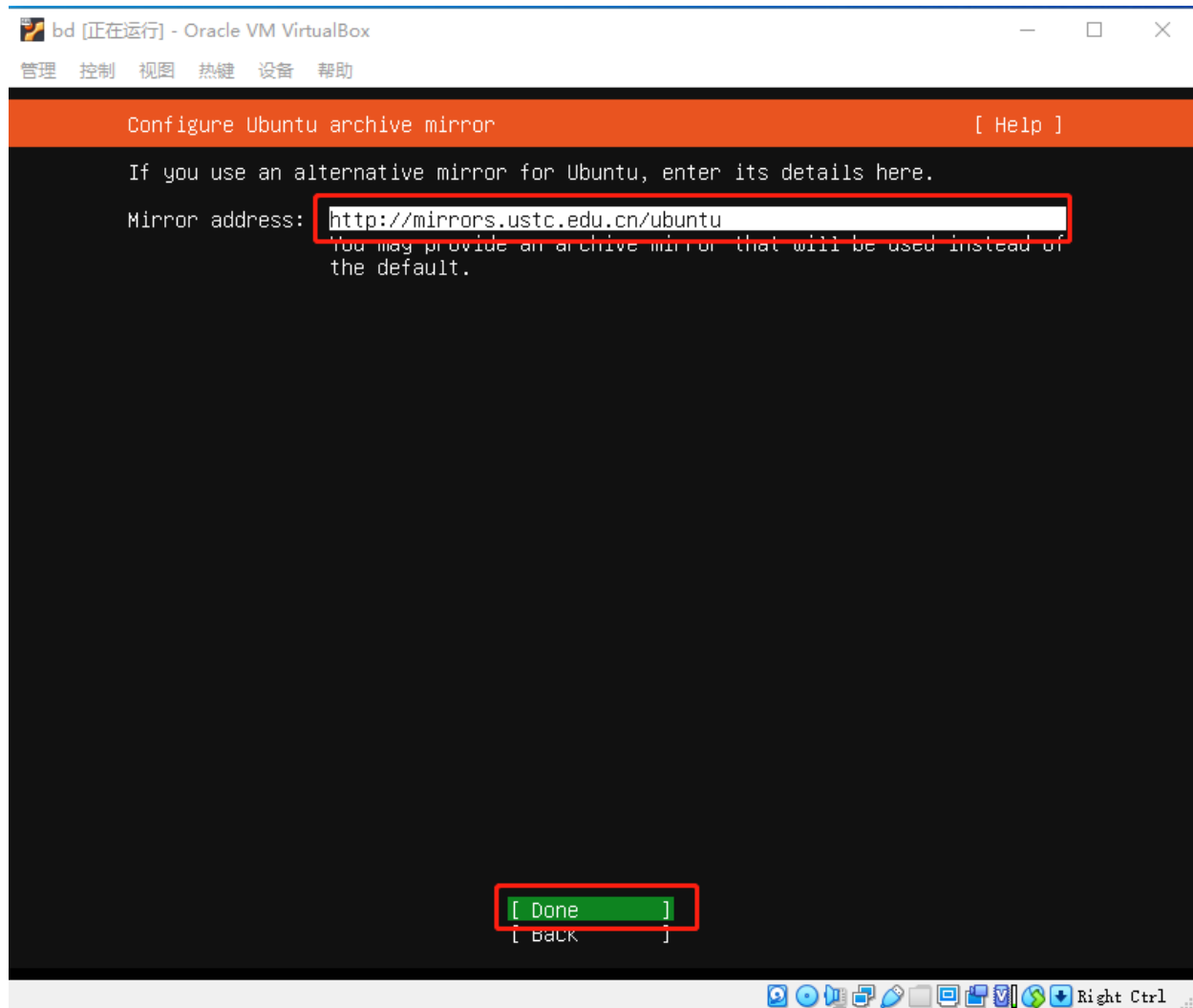
NAME	TYPE	NOTES
[enp0s3	eth	- ▶]
DHCPv4 10.0.2.15/24		
08:00:27:81:60:92 / Intel Corporation / 82540EM Gigabit Ethernet Controller (PRO/1000 MT Desktop Adapter)		
[Create bond ▶]		

[Done]

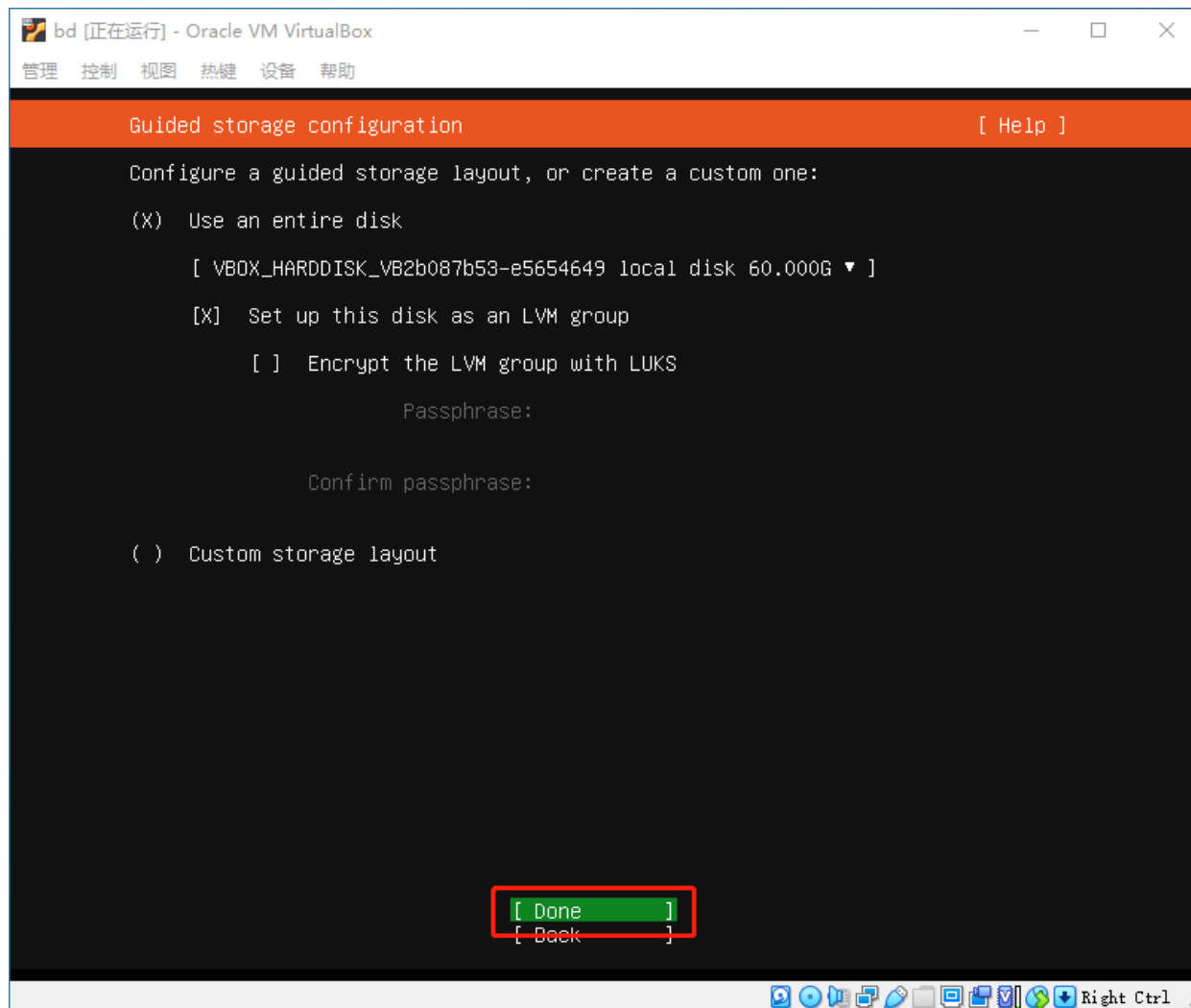
[Back]



配置中国科学技术大学软件源，加速软件包下载更新速度



磁盘分区，使用默认（直接键盘向下键一到 Done 并回车）



Storage configuration

[Help]

FILE SYSTEM SUMMARY

MOUNT POINT	SIZE	TYPE	DEVICE TYPE
[/	29.498G	new ext4	new LVM logical volume ▶]
[/boot	1.000G	new ext4	new partition of local disk ▶]

AVAILABLE DEVICES

DEVICE	TYPE	SIZE
[ubuntu-vg (new)	LVM volume group	58.996G ▶]
free space		29.498G

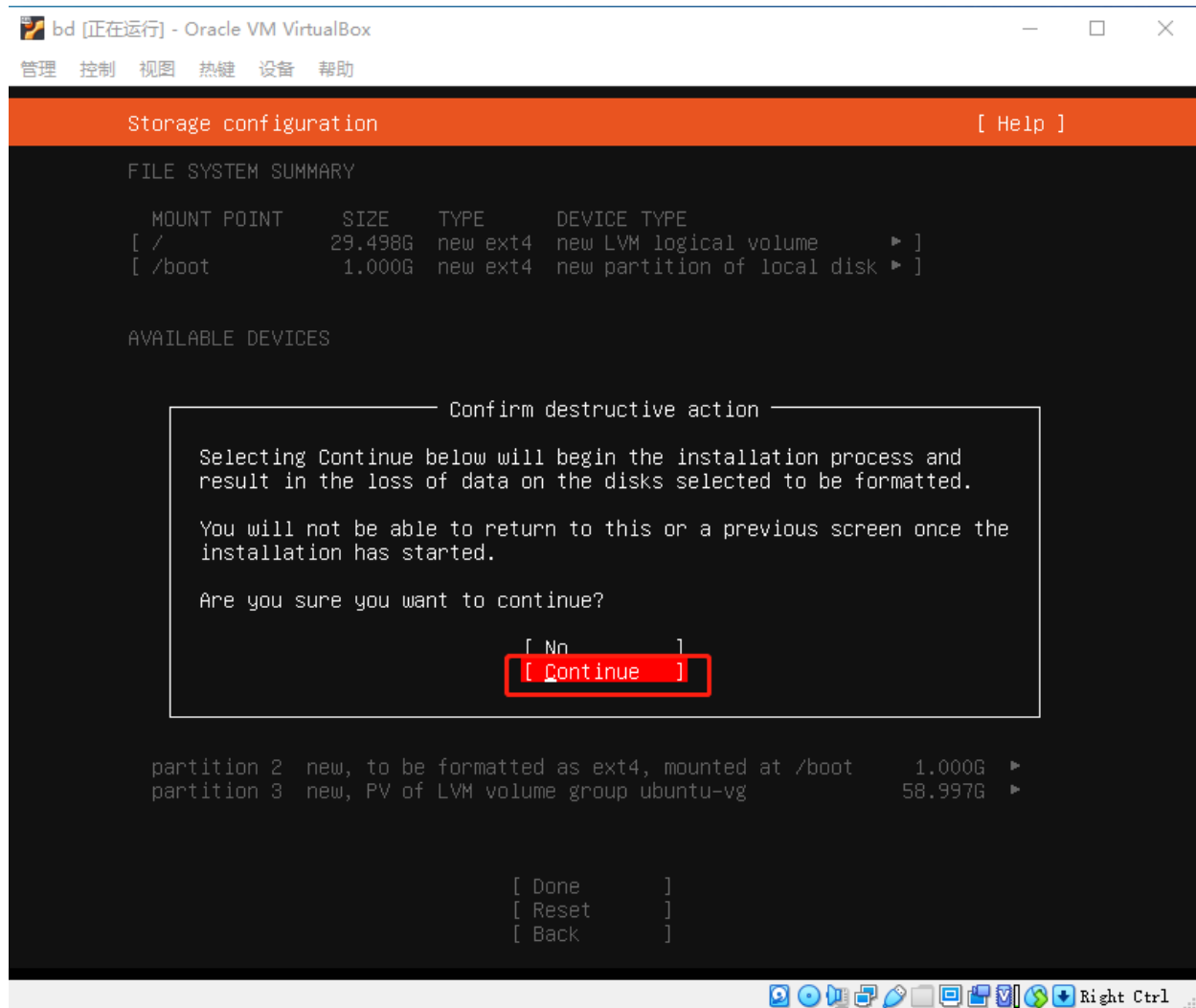
[Create software RAID (md) ▶]
[Create volume group (LVM) ▶]

USED DEVICES

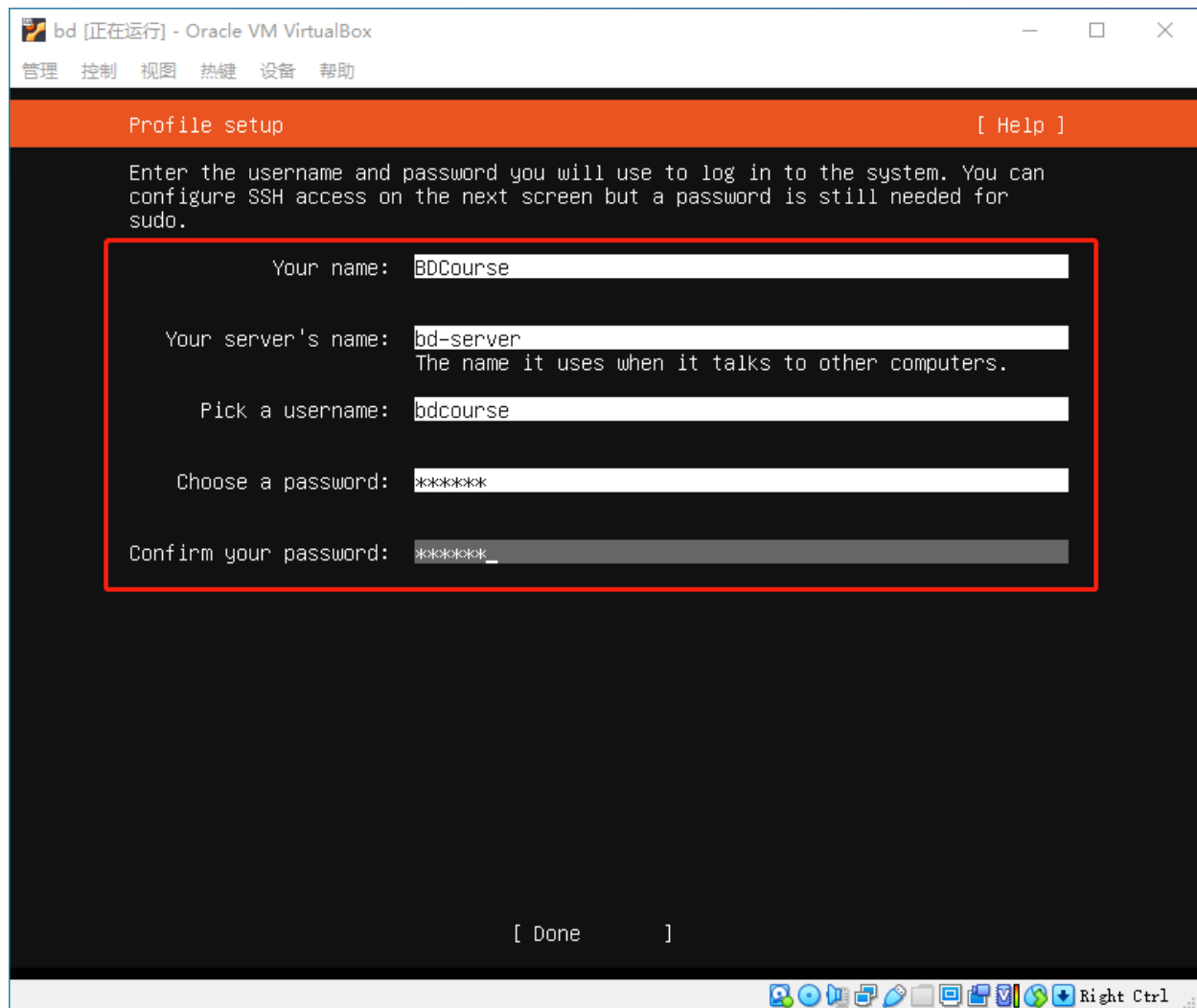
DEVICE	TYPE	SIZE
[ubuntu-vg (new)	LVM volume group	58.996G ▶]
ubuntu-lv new, to be formatted as ext4, mounted at /		29.498G ▶

[VBOX_HARDDISK_VB2b087b53-e5654649 local disk 60.000G ▶]
partition 1 new, bios_grub 1.000M ▶
partition 2 new, to be formatted as ext4, mounted at /boot 1.000G ▶
partition 3 new, PV of LVM volume group ubuntu-vg 58.997G ▶

[Done]
[Reset]
[Back]



配置用户名 (bdcourse)、密码 (Bd2021)、主机名 (bd-server) 等



安装OpenSSH Server (键盘光标移动到中括号中, 点击空格键选中)

SSH Setup

[Help]

You can choose to install the OpenSSH server package to enable secure remote access to your server.

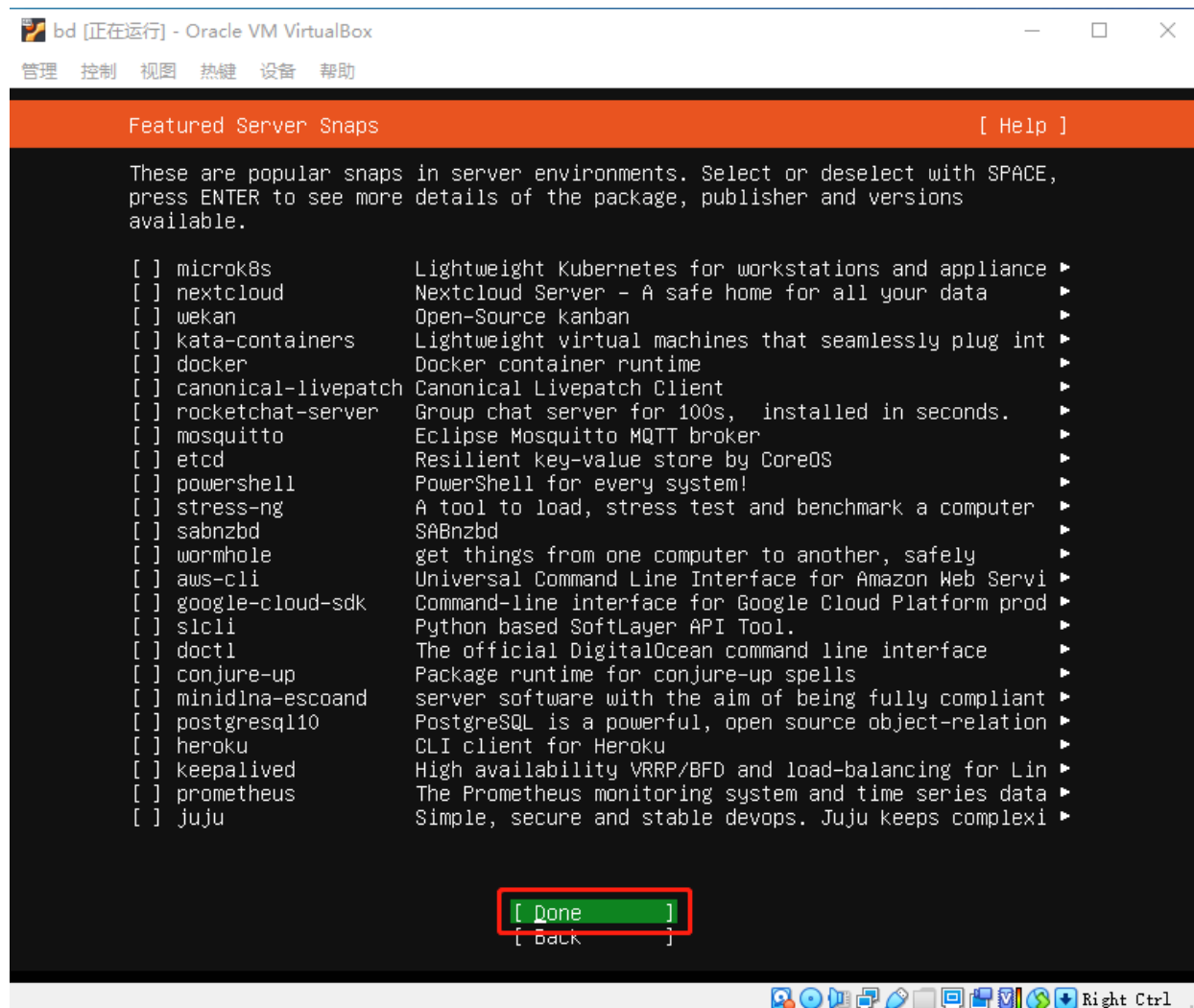
☒ Install OpenSSH server

Import SSH identity: [No ▼]
You can import your SSH keys from Github or Launchpad.

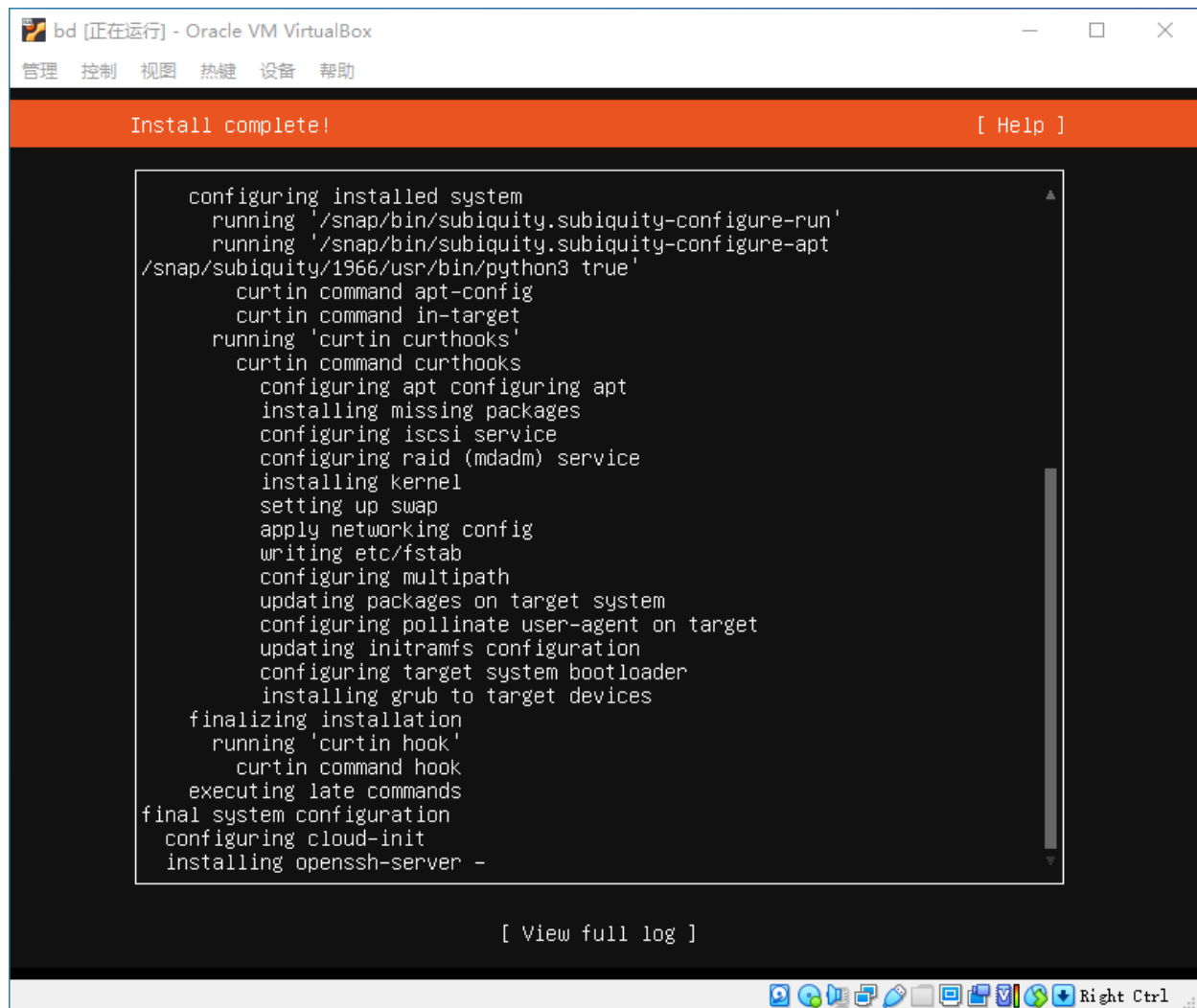
Import Username:

☐ Allow password authentication over SSH

[Done]
[Back]



开始安装系统



安装完成，重启系统

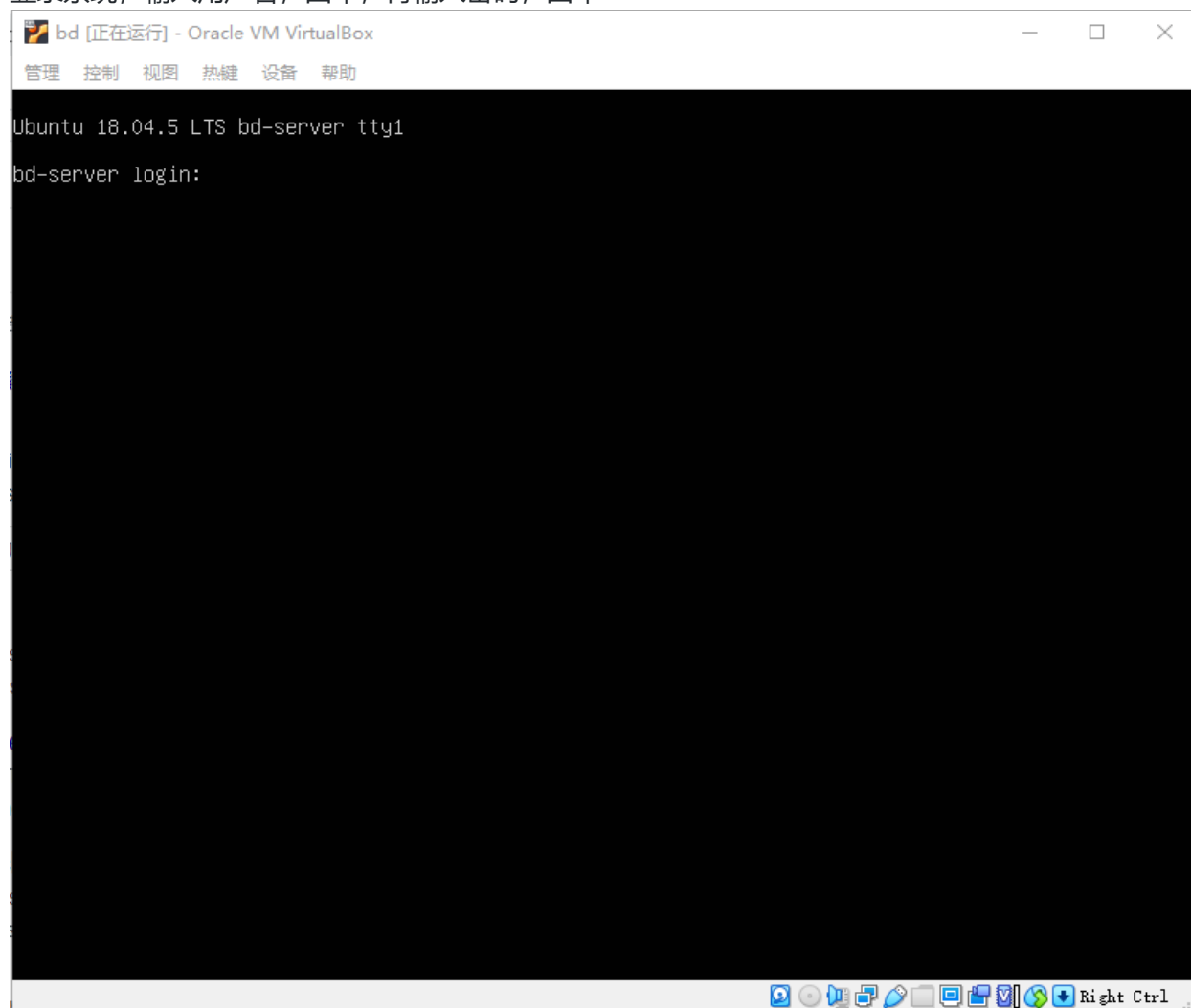
Installation complete!

[Help]

```
Finished install!  
/snap/subiquity/1966/usr/bin/python3 true'  
  curtin command apt-config  
  curtin command in-target  
  running 'curtin curthooks'  
  curtin command curthooks  
    configuring apt configuring apt  
    installing missing packages  
    configuring iscsi service  
    configuring raid (mdadm) service  
    installing kernel  
    setting up swap  
    apply networking config  
    writing etc/fstab  
    configuring multipath  
    updating packages on target system  
    configuring pollinate user-agent on target  
    updating initramfs configuration  
    configuring target system bootloader  
    installing grub to target devices  
  finalizing installation  
    running 'curtin hook'  
    curtin command hook  
  executing late commands  
final system configuration  
  configuring cloud-init  
  installing openssh-server  
  restoring apt configuration  
  downloading and installing security updates
```

[\[View full log \]](#)[\[Reboot \]](#)

登录系统，输入用户名，回车，再输入密码，回车



输入 `ifconfig` 命令查看系统IP

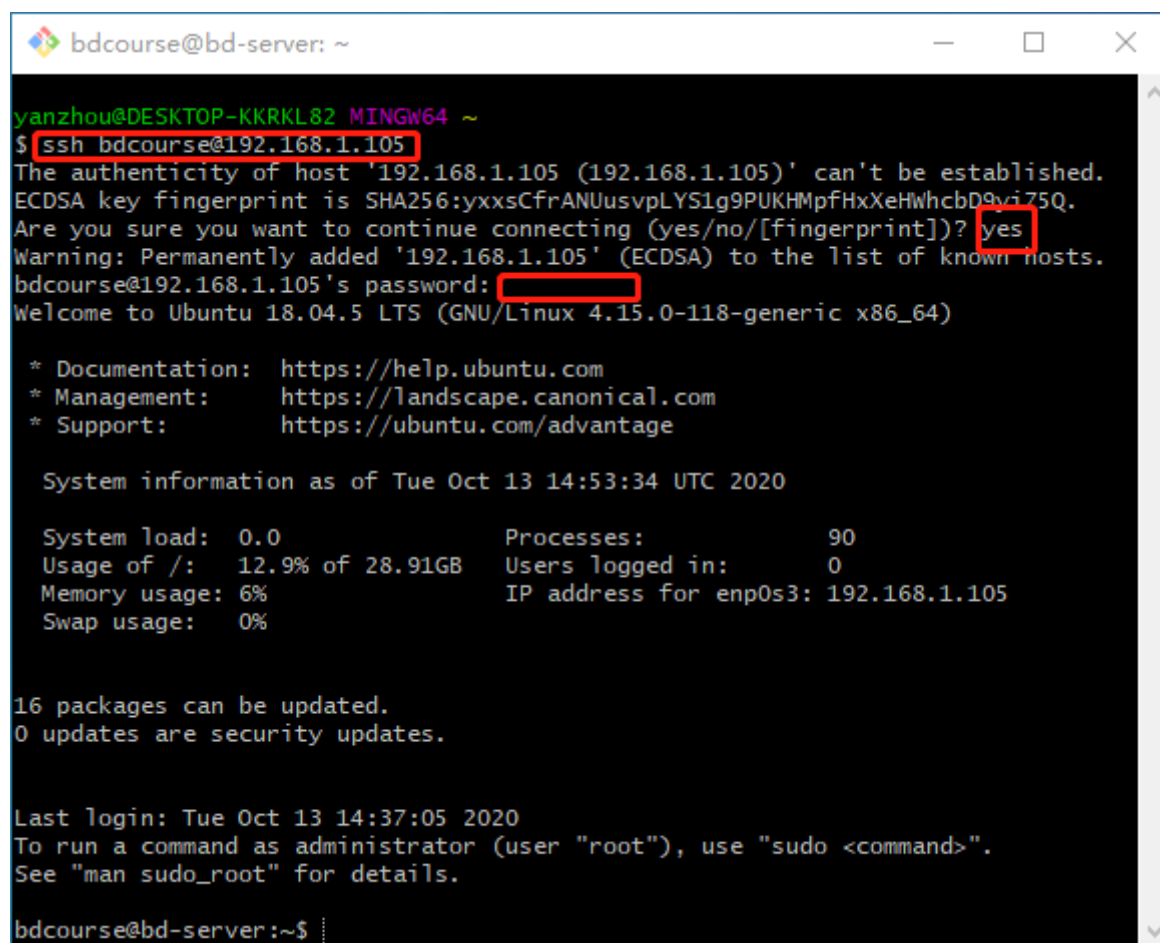
```
bdcourse@bd-server:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.105 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::a00:27ff:fe81:6092 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:81:60:92 txqueuelen 1000 (Ethernet)
    RX packets 24 bytes 6387 (6.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 25 bytes 3220 (3.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 84 bytes 6308 (6.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 84 bytes 6308 (6.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Bash工具安装

安装[git for windows](#)或[PuTTY](#)或[Cygwin](#)

SSH (Git Bash) 远程登录虚拟机，登录命令格式 `ssh <username>@<host>`，其中 `<username>` 替换为具体用户名，`<host>` 替换为远程主机IP地址。



```
bdcourse@bd-server: ~
yanzhou@DESKTOP-KKRKL82 MINGW64 ~
$ ssh bdcourse@192.168.1.105
The authenticity of host '192.168.1.105 (192.168.1.105)' can't be established.
ECDSA key fingerprint is SHA256:yxxsCfrANUusvpLYS1g9PUKHmpfHxXeHWhcbD9vi75Q.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.105' (ECDSA) to the list of known hosts.
bdcourse@192.168.1.105's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-118-generic x86_64)

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

System information as of Tue Oct 13 14:53:34 UTC 2020

System load:  0.0               Processes:    90
Usage of /:   12.9% of 28.91GB   Users logged in: 0
Memory usage: 6%               IP address for enp0s3: 192.168.1.105
Swap usage:  0%

16 packages can be updated.
0 updates are security updates.

Last login: Tue Oct 13 14:37:05 2020
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

bdcourse@bd-server:~$
```

1.1 安装Anaconda

说明：本课程操作手册中所有 开头的表示shell命令，# 开头的表示注释说明，需要仔细阅读。

```
# wget为Linux的下载命令
$ wget https://repo.anaconda.com/archive/Anaconda3-2021.05-Linux-x86_64.sh

# 安装Anaconda, bash执行shell脚本, shell脚本可以看作是多条shell命令集合在一起的一个文件, 当然
shell脚本也可以包含控制语句
# 使用默认安装位置 (PREFIX=/home/bdcourse/anaconda3)
# 选择初始化Anaconda3 (Do you wish the installer to initialize Anaconda3 by running conda
init? yes)
$ bash Anaconda3-2021.05-Linux-x86_64.sh

# 退出重新登录, 环境生效
$exit
```

1.2 安装scrapy和pymysql

```
# 首先配置pypi源为清华大学源
$ pip install -i https://pypi.tuna.tsinghua.edu.cn/simple pip -U
$ pip config set global.index-url https://pypi.tuna.tsinghua.edu.cn/simple
```



```
# 安装gcc
$ sudo apt-get install gcc

$ pip install scrapy pymysql
```

1.3 安装Chrome Headless (选作)

```
$ sudo apt-get install libxss1 libappindicator1 libindicator7
$ wget https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb
# 或者从百度网盘下载之后复制到虚拟机中
# scp google-chrome-stable_current_amd64.deb bdcourse@192.168.1.105:~/
# 下一行命令会报错, 使用最后一行修复
$ sudo dpkg -i google-chrome*.deb
$ sudo apt-get install -f
```

1.4 安装Chrome Driver (选做)

```
$ sudo apt-get install unzip
$ wget https://chromedriver.storage.googleapis.com/86.0.4240.22/chromedriver_linux64.zip
# 或者从百度网盘下载之后复制到虚拟机中
# scp chromedriver_linux64.zip bdcourse@192.168.1.105:~/
$ unzip chromedriver_linux64.zip
$ mkdir /home/bdcourse/opt
$ sudo mv chromedriver /home/bdcourse/opt/

# 配置环境变量
$ sudo vi /etc/profile
# 修改PATH, 在PATH中添加/home/bdcourse/opt/chromedriver, 例如
# 注意, 需要根据自己的PATH在最前面添加/home/bdcourse/opt/chromedriver, 而不是完全照搬下面的配置
export PATH=/home/bdcourse/opt/chromedriver:$PATH
$ source /etc/profile
```

1.5 安装Selenium (选作)

```
$ pip install selenium
```

参考文档:

- <https://www.selenium.dev/documentation/en/>
- <https://wangxin1248.github.io/linux/2018/09/ubuntu18.04-install-chrome-headless.html>

2. 创建Scrapy项目

2.0 爬取策略

从SegmentFault用户排行榜首页（如下图所示）中的用户作为初始节点，从每个用户从个人主页（如 <https://segmentfault.com/u/evilboy>）爬取用户基本信息、内容信息和活动信息，顺着用户的粉丝和关注关系使用广度优先搜索方式爬取更多用户信息和用户好友关系。

segmentfault

首页 问答 专栏 资讯 课程 活动 发现

搜索问题或关键字

立即登录 免费注册

用户排行榜

这里是活跃用户们的贡献排行榜，他们为社区贡献了不可磨灭的力量，让更多人得到了成长。

今天 还剩01时31分00秒

1. 疯狂的技术宅 +150	11. Java技术栈 +40	1. justjavac 46.6k	11. weakish 24k
2. 徐九 +98	12. 宗恩 +38	2. 前端小智 42.5k	12. hsfzxjy 22k
3. 前端小智 +77	13. 渔台宇鹏 +38	3. 边城 39.2k	13. 王下邀月... 21.9k
4. 民工哥 +73	14. justjavac +33	4. 公子 36.1k	14. mcfog 21.8k
5. 老徐不二 +61	15. yzlee +31	5. nightire 30.6k	15. 沙渺 21.6k
6. 高阳Sunny +60	16. xialeistudio +30	6. trigkit4 29.8k	16. xialeistudio 20.7k
7. 然后去远足 +56	17. Roger李 +26	7. 疯狂的技术宅 28k	17. 苏生不惑 18.6k
8. 治王治治 +55	18. 李 linong +26	8. 有明 26.5k	18. kikong 18.3k
9. 超神经Hyp... +55	19. xuriliang +25	9. leftstick 26.5k	19. 浪里行舟 18.3k
10. Peter谭金杰 +51	20. LeanCloud +25	10. 依云 24.9k	20. 守候 17.9k

综合 活跃度 + 声望

本周 还剩2天

1. 疯狂的技术宅 +361	1. 民工哥 +1961	1. 民工哥 +1961	1. 前端小智 +4189
2. 民工哥 +310	2. 疯狂的技术宅 +1227	2. 疯狂的技术宅 +1227	2. 疯狂的技术宅 +3199
3. 前端小智 +270	3. 前端小智 +1010	3. 前端小智 +1010	3. 民工哥 +3192
4. 高阳Sunny +230	4. 徐九 +972	4. 徐九 +972	4. Eno_Yao +2887
5. 然后去远足 +179	5. Java技术栈 +944	5. Java技术栈 +944	5. fefe +2779
6. 徐九 +128	6. 然后去远足 +850	6. 然后去远足 +850	6. 然后去远足 +2761
7. Eno_Yao +122	7. Choerodon... +568	7. Choerodon... +568	7. 徐九 +2688
8. 浪里行舟 +121	8. Eno_Yao +530	8. Eno_Yao +530	8. Meathill +2123
9. LeanCloud +112	9. wale +524	9. wale +524	9. 李 linong +1942
10. SegmentFault +106	10. 木夕木夕 +515	10. 木夕木夕 +515	10. 阿山 +1917

4月份 还剩6天

第2季度 还剩67天

2020年度 还剩251天

2.1 新建项目

```
# segmentfaultspider为项目的名称，可以自定义
$ scrapy startproject segmentfaultspider
```

项目 segmentfaultspider 目录结构为：

```
| - scrapy.cfg           # 部署相关配置
| - segmentfaultspider/  # 项目Python模块
|   | - __init__.py
|   | - items.py         # item对象定义文件
|   | - middlewares.py
|   | - pipelines.py
|   | - settings.py      # 项目配置
```

```
| | - spiders/          # 爬虫文件夹
| | | - __init__.py
```

2.2 定义item

Item是爬取信息的存储数据结构，下面是SegmentFault用户信息item示例，实际爬取的时候根据需要可以更加定义更多的Item，更多的字段。

```
# file: segmentfaultspider/segmentfaultspider/items.py

# vi是Linux的编辑器，可用于编辑文本文件
$ vi segmentfaultspider/segmentfaultspider/items.py

# -*- coding: utf-8 -*-
import scrapy

class UserItem(scrapy.Item):
    username = scrapy.Field()
    nickname = scrapy.Field()
```

2.3 定义Spider

下面是SegmentFault网站用户信息爬虫示例，根据实际需要可以新建更多的爬虫，解析更多的字段。

```
# file: segmentfaultspider/segmentfaultspider/spiders/userspider.py

# -*- coding: utf-8 -*-

import scrapy

# 导入UserItem
from segmentfaultspider.items import UserItem

class UserSpider(scrapy.Spider):
    # 爬虫名称，用于执行爬虫时命令 scrapy crawl <spidername> 中的<spidername>
    name = 'users'

    # 爬虫启动后首先执行的函数
    def start_requests(self):
        top10_url = 'https://segmentfault.com/users'
        # 发送Request请求，其中url参数为请求的URL地址，callback为处理返回响应内容的函数名称（自定义，需要实现）
        yield scrapy.Request(url=top10_url, callback=self.parse_top10)

    # 处理响应数据，response为响应对象
    def parse_top10(self, response):
        # response对象有xpath函数，使用XPath语法对相应的HTML标签进行搜索
        # 这里用户列表在class为widget-top10的列表中
        users = response.xpath('//ol[contains(@class, "widget-top10")]/li')

        for user in users:
            user_item = UserItem()
            # TODO 考虑健壮性
            # 每个user对象上也有xpath方法，可以继续进行搜索
            user_item['username'] = user.xpath('a/@href').get().split('/')[2]
            user_item['nickname'] = user.xpath('a/span/text()').get()

            yield user_item

        # TODO 抓取详情页
        profile_url = 'https://segmentfault.com/u/' + user_item['username']
        yield scrapy.Request(url=profile_url, callback=self.parse_profile)

    def parse_profile(self, response):
        # TODO 实现
        pass
```

此时在**项目根目录**执行 `scrapy crawl users`，则会将爬取的结果打印到屏幕，如下图：

```
2020-04-24 14:23:52 [scrapy.core.scraper] DEBUG: Scraped from <200 https://segmentfault.com/users>
{'nickname': '疯狂的技术宅', 'username': 'evilboy'}
2020-04-24 14:23:52 [scrapy.core.scraper] DEBUG: Scraped from <200 https://segmentfault.com/users>
{'nickname': '徐九', 'username': 'weepie'}
2020-04-24 14:23:52 [scrapy.core.scraper] DEBUG: Scraped from <200 https://segmentfault.com/users>
{'nickname': '前端小智', 'username': 'minnanitkong'}
2020-04-24 14:23:52 [scrapy.core.scraper] DEBUG: Scraped from <200 https://segmentfault.com/users>
{'nickname': '民工哥', 'username': 'jishuroad'}
2020-04-24 14:23:52 [scrapy.core.scraper] DEBUG: Scraped from <200 https://segmentfault.com/users>
{'nickname': '老徐不二', 'username': 'laoxubuer'}
2020-04-24 14:23:52 [scrapy.core.scraper] DEBUG: Scraped from <200 https://segmentfault.com/users>
{'nickname': '高阳Sunny', 'username': 'sunny'}
2020-04-24 14:23:52 [scrapy.core.scraper] DEBUG: Scraped from <200 https://segmentfault.com/users>
{'nickname': '然后去远足', 'username': 'rhqyz'}
2020-04-24 14:23:52 [scrapy.core.scraper] DEBUG: Scraped from <200 https://segmentfault.com/users>
{'nickname': '王治治', 'username': 'wangdazhi_sifou'}
```

2.4 数据存储到MySQL

安装MySQL Server

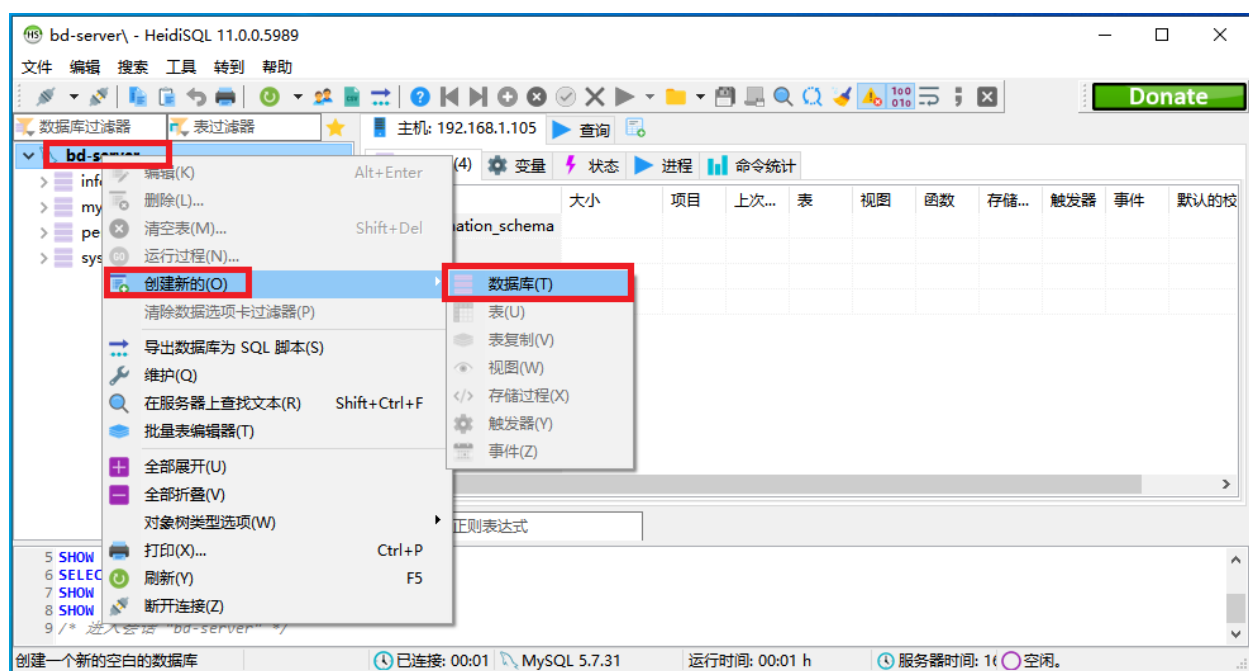
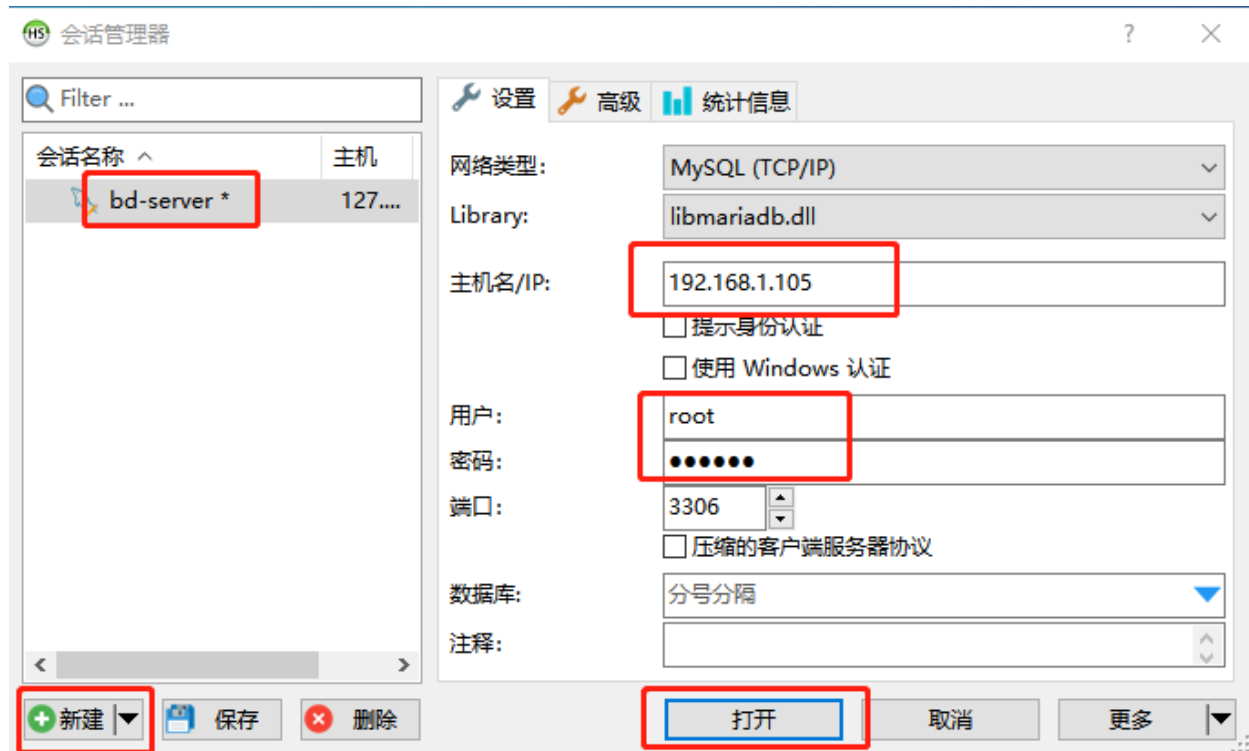
```
$ sudo apt update
$ sudo apt install mysql-server
$ sudo mysql_secure_installation
# root密码Bd2021
# 其中VALIDATE PASSWORD PLUGIN选择n
# 其中 Disallow root login remotely? (Press y/Y for Yes, any other key for No) : n

# 注意，如果上面的VALIDATE PASSWORD PLUGIN没有选择n将会出现密码强度验证问题，可以通过下面的方法解决
# 解决MySQL问题
$ sudo mysql -u root
mysql> update mysql.user set authentication_string=PASSWORD('Bd2021'),
plugin='mysql_native_password' where user='root';
mysql> flush privileges;
mysql> exit;
$ sudo service mysql restart

# 修改mysql配置
$ sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf
# 注释掉bind-address = 127.0.0.1
$ sudo systemctl restart mysql
```

安装HeidiSQL，官网地址：<https://www.heidisql.com/download.php>

使用HeidiSQL连接到数据库，地址：`192.168.1.105:3306`，新建数据库 `segmentfault` 和表 `users`（创建 `username` 和 `nickname` 等字段），如下图所示。



创建数据库...

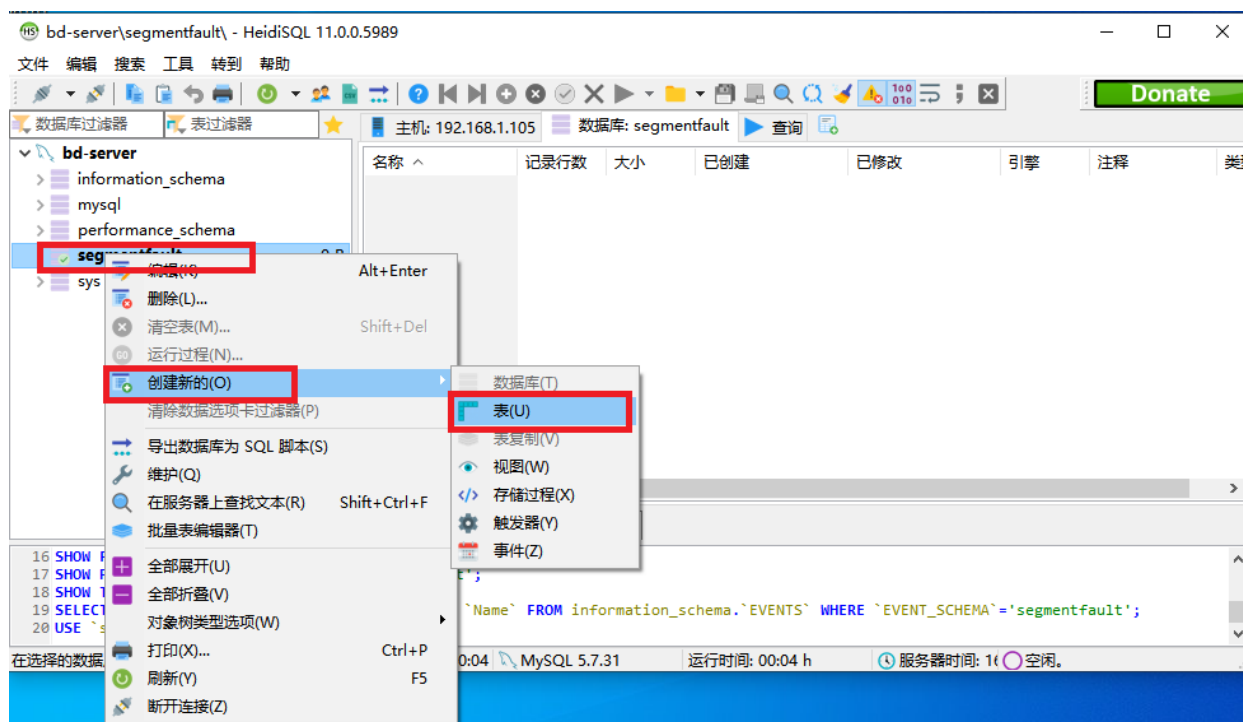
名称(N):

字符校对(O):

服务器默认: latin1_swedish_ci

CREATE 代码:

```
CREATE DATABASE `segmentfault` /*!40100 CO
```



基本 选项 索引 外键 分区 CREATE 代码

名称:

注释:

字段:

#	名称	数据类型	长度/集合	无符号的	允许 N...	填零	默认
1	username	VARCHAR	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
2	nickname	VARCHAR	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL

修改项目配置文件，配置ITEM PIPELINES

```
# file: segmentfaultspider/segmentfaultspider/settings.py

ITEM_PIPELINES = {
    'segmentfaultspider.pipelines.UserItemPipeline': 300,
}
```

最后，编写 `UserItemPipeline`，实现抓取的用户数据（`UserItem`）写入到数据库中。

```
# file: segmentfaultspider/segmentfaultspider/pipelines.py

# -*- coding: utf-8 -*-

import pymysql
from segmentfaultspider.items import UserItem

class UserItemPipeline(object):
    def __init__(self):
        # 配置MySQL数据库连接
        self.connection = pymysql.connect(host='192.168.1.199',
                                          port=3306,
                                          user='<your user name>',
                                          password='<your password>',
                                          db='<your database>',
                                          charset='utf8mb4')

    def process_item(self, item, spider):
        # 判断是否是UserItem对象，因为不同的Item对象都会经过同一个pipeline
        if isinstance(item, UserItem):
            cursor = self.connection.cursor()
            cursor.execute('INSERT INTO `users` (`username`, `nickname`) VALUES (%s, %s)', (item['username'], item['nickname']))
            # TODO 健壮性，捕获异常，处理主键重复等异常
            # 注意，下面这一行必须有，cursor.execute之后需要执行下面的commit才会提交给MySQL服务器执行
            self.connection.commit()
```

3. Scrapy shell使用

Scrapy提供一种交互式shell，用于调试Scrapy爬虫代码，**主要用于测试XPath或CSS表达式。**

启动Scrapy Shell

```
# <url>为要爬取页面的URL 地址
$ scrapy shell <url>
# 例如
$ scrapy shell https://segmentfault.com/users
```

Scrapy Shell打开ipython进行交互，并自动创建相关变量和函数，如下图红框标识。


```

2020-04-24 15:22:08 [asyncio] DEBUG: Using selector: EpollSelector
[s] Available Scrapy objects:
[s] scrapy      scrapy module (contains scrapy.Request, scrapy.Selector, etc)
[s] crawler     <scrapy.crawler.Crawler object at 0x7ff890f90bd0>
[s] item        {}
[s] request     <GET https://segmentfault.com/users>
[s] response    <200 https://segmentfault.com/users>
[s] settings     <scrapy.settings.Settings object at 0x7ff890f2e350>
[s] spider      <DefaultSpider 'default' at 0x7ff890acf0d0>
[s] Useful shortcuts:
[s] fetch(url[, redirect=True]) Fetch URL and update local objects (by default
, redirects are followed)
[s] fetch(req)                  Fetch a scrapy.Request and update local object
s
[s] shelp()                    Shell help (print this help)
[s] view(response)            View response in a browser
2020-04-24 15:22:08 [asyncio] DEBUG: Using selector: EpollSelector
In [1]: |

```

可以在Scrapy shell中对UserSpider中的XPath或CSS表达式进行测试，如下面一组图所示

```

In [1]: users = response.xpath('//ol[contains(@class, "widget-top10")]/li')
In [2]: users
Out[2]:
[<Selector xpath="//ol[contains(@class, "widget-top10")]/li" data='<li>\n
href="/u/evilboy">\n ... '>,
  <Selector xpath="//ol[contains(@class, "widget-top10")]/li" data='<li>\n
href="/u/weepie">\n ... '>,
  <Selector xpath="//ol[contains(@class, "widget-top10")]/li" data='<li>\n
href="/u/minnanitkong... '>,
  <Selector xpath="//ol[contains(@class, "widget-top10")]/li" data='<li>\n
href="/u/jishuroad">\n... '>,
  <Selector xpath="//ol[contains(@class, "widget-top10")]/li" data='<li>\n
href="/u/laoxubuer">\n... '>,
  <Selector xpath="//ol[contains(@class, "widget-top10")]/li" data='<li>\n
href="/u/sunny">\n ... '>,
  <Selector xpath="//ol[contains(@class, "widget-top10")]/li" data='<li>\n
href="/u/rhgyz">\n ... '>,
In [3]: users[0].xpath('a[@href]').get()
Out[3]: '/u/evilboy'

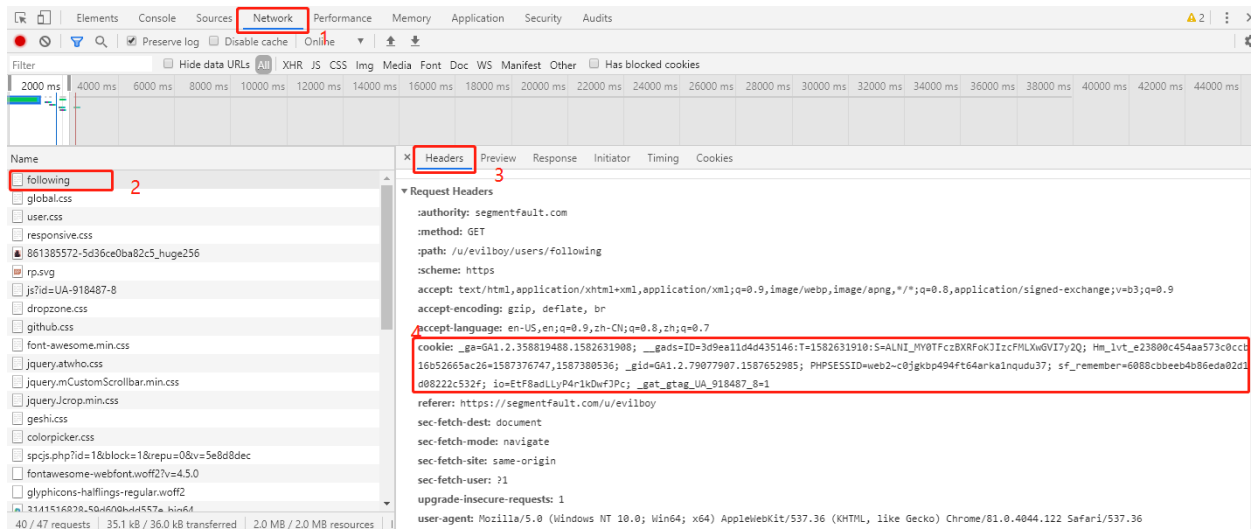
```

4. 需登录才能访问页面的爬取（选作）

4.1 方案1 - 从浏览器获取登录后的cookie

HTTP是无状态协议，浏览器和服务端之间的状态保持是通过cookie/session机制实现的，在浏览器端通过cookie存储状态信息，在服务端通过session存储状态信息。在每次请求时浏览器会将cookie信息以HTTP头的信息发送给服务端，表明自己的状态（例如登录状态和登录账户信息）。Scrapy下载网页的时候也可以随带HTTP头信息，因此可以将浏览器登录后的cookie信息直接复制到Scrapy请求的HTTP头配置中。

HTTP请求可以通过Chrome的调试工具进行查看，如下图所示为SegmentFault网站登录后，访问关注列表页时发送的HTTP请求的请求头。



将上图所示HTTP头信息中的部分信息复制出来，构造Scrapy请求头信息和cookie信息。

注意其中的:path和referer根据实际爬取的用户进行修改

```
headers = {
    ':authority': 'segmentfault.com',
    ':method': 'GET',
    ':path': '/u/evilboy/users/following',
    ':scheme': 'https',
    'referer': 'https://segmentfault.com/u/evilboy',
    'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
    like Gecko) Chrome/81.0.4044.122 Safari/537.36'
}

cookies = {
    'PHPSESSID': 'web2~c0jgkbp494ft64arka1nqudu37'
}
```

然后在Scrapy Shell中进行调试，输入输出如下所示：

```
In [1]: headers = {
...:     ':authority': 'segmentfault.com',
...:     ':method': 'GET',
...:     ':path': '/u/evilboy/users/following',
...:     ':scheme': 'https',
...:     'referer': 'https://segmentfault.com/u/evilboy',
...:     'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537
...: .36 (KHTML, like Gecko) Chrome/81.0.4044.122 Safari/537.36'
...: }

In [2]: cookies = {
...:     'PHPSESSID': 'web2~c0jgkbp494ft64arka1nqudu37'
...: }

In [3]: req = scrapy.Request('https://segmentfault.com/u/evilboy/users/following',
headers=headers, cookies=cookies)

In [4]: fetch(req)
2020-04-25 07:51:38 [scrapy.core.engine] DEBUG: Crawled (200) <GET
```

```
https://segmentfault.com/u/evilboy/users/following> (referer:
https://segmentfault.com/u/evilboy)

In [5]: response.xpath('//title/text()').get()
Out[5]: '疯狂的技术宅 关注的人 - SegmentFault 思否'

In [6]: users = response.xpath('//ul[contains(@class, "profile-following__users")]/li')

In [7]: users[0].xpath('div/div/div/a/@href').get()
Out[7]: '/u/liketree'

In [8]: users[0].xpath('div/div/div/a/text()').get()
Out[8]: '喻木同學'
```

4.2 方案2 - Selenium模拟登录，获取session ID

```
from selenium import webdriver
from selenium.webdriver.common.action_chains import ActionChains

options = webdriver.ChromeOptions()
driver =
webdriver.Chrome(options=options,executable_path='/home/bdcourse/opt/chromedriver')
driver.get('https://segmentfault.com/user/Login')

# 定位密码登录表单链接
pwd_login_form = driver.find_element_by_xpath('(//a[contains(@class, "login-mode")])[2]')
# 执行点击密码登录表单链接操作，直接pwd_login_form.click()不行，原因未知
ActionChains(driver).move_to_element(pwd_login_form).click(pwd_login_form).perform()
# 获取用户名输入框、密码输入框和登录按钮
usernameinput = driver.find_element_by_xpath('//form[contains(@class, "password-
form")]/input[@name="username"]')
passwordinput = driver.find_element_by_xpath('//form[contains(@class, "password-
form")]/input[@name="password"]')
submitbtn = driver.find_element_by_xpath('//form/button[contains(@class, "sf_do")]')

# 发送表单信息到表单文本框
usernameinput.send_keys('<your username>')
passwordinput.send_keys('<your password>')
submitbtn.click()

# 获取session ID
driver.get_cookie('PHPSESSID')

# 关闭Selenium
driver.quit()

# 获取session ID之后就可以按照方案4.1继续构造请求
```

参考：

- [Selenium Python API 文档](#)

