

云计算与大数据应用开发

实验一：虚拟化技术

丁烨

dingye@dgut.edu.cn

计算机科学与技术学院

2024-03-18



東莞理工學院
DONGGUAN UNIVERSITY OF TECHNOLOGY

❖ Docker

❖ <https://www.docker.com/>

❖ 开源免费，源代码：<https://github.com/docker/docker-ce/>



❖ Docker 是一个开源软件项目，提供基于容器（Container）的虚拟化技术

❖ Docker 利用 Linux 核心中的资源分离机制来创建独立的容器

❖ 容器使得 Docker 可以在单一 Linux 实体下运作

❖ 避免创建多个虚拟机造成的额外系统负担

❖ Docker 提倡的容器化技术是目前最主流的虚拟机解决方案

❖ 容器 (Container)

❖ 一般虚拟机的概念里，一个操作系统镜像 (Image) 可以用于创建多个**虚拟机 (VM)**

❖ 每个虚拟机拥有**独立**的虚拟内存、虚拟 CPU、虚拟硬盘，操作系统**完全隔离**

❖ 虚拟机 (虚拟硬盘) 可以被复制并重启，从而产生分支

❖ Docker 的概念里，一个硬盘镜像 (Image) 可以用于创建多个**容器 (Container)**

❖ Docker 一般不会直接使用操作系统镜像，而是包含需要使用的应用软件

❖ 容器**共享**宿主机的内存、CPU，但硬盘相对独立 (采用分块镜像模式)

❖ 容器共享宿主机的**部分操作系统指令**

❖ 容器可以重新打包成为新的硬盘镜像，镜像可以被用于创建新的容器，从而产生分支

❖ 容器内没有自己的内核，而且也没有进行硬件虚拟，因此容器要比传统虚拟机更为轻便

❖ Docker 的优点

- ❖ 不严格符合虚拟机标准，而是采用 Linux 核心资源分离技术实现虚拟化
- ❖ 性能卓越，接近物理机性能
- ❖ 部署虚拟机（容器）不需要部署整个操作系统，操作快速便捷
- ❖ 拥有庞大的社区，提供了大量开源免费（以及收费）的操作系统镜像

❖ Docker 的缺点

- ❖ 仅支持 Linux，其他操作系统需要借助其他虚拟化技术虚拟 Linux 环境
- ❖ MacOS: 借助 HyperKit（一般不用手动安装）
- ❖ Windows: 借助 Hyper-V

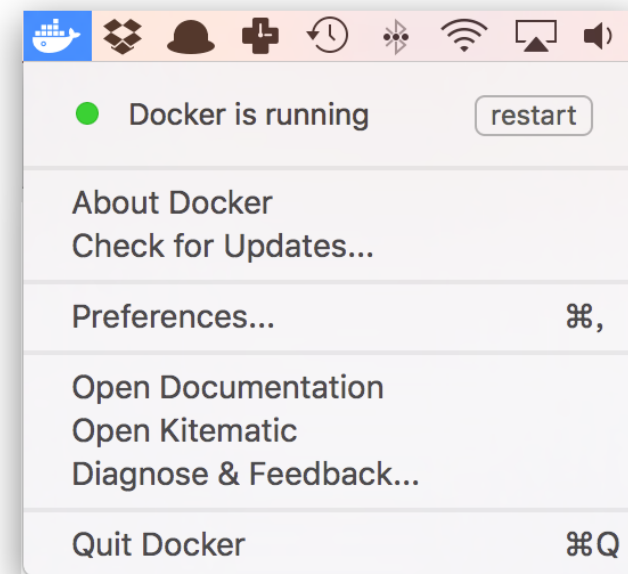
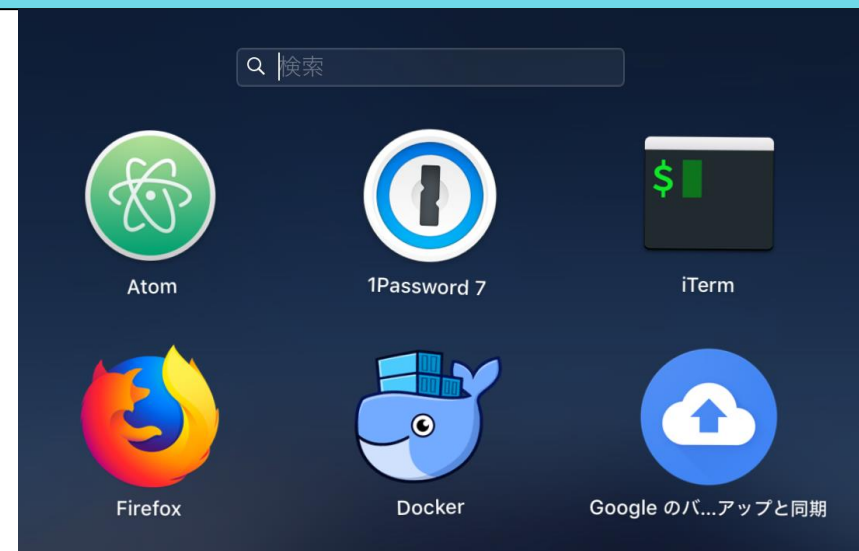
- ❖ 我们选择以下四种操作系统作为示例：
 - ❖ Ubuntu 22.04
 - ❖ macOS 14
 - ❖ Windows 11
-
- ❖ 如果你使用云计算服务提供的 Linux，请尽量选择 Ubuntu 22.04 作为操作系统
 - ❖ 由于实验室环境配置有限，**建议大家自带设备完成实验**

- ❖ 在 Ubuntu 22.04 下安装 Docker
- ❖ <https://docs.docker.com/engine/install/ubuntu/>
- ❖ `curl -fsSL get.docker.com -o get-docker.sh`
- ❖ `sudo sh get-docker.sh --mirror Aliyun`
- ❖ 安装完毕后，如果希望以非管理员身份访问 Docker:
- ❖ `sudo usermod -aG docker <username>`

安装 Docker

macOS 14

- ❖ 在 macOS 14 下安装 Docker
- ❖ <https://docs.docker.com/desktop/mac/install/>
- ❖ `brew cask install docker`
- ❖ 安装完毕后，从应用中找到 Docker 图标并点击运行
- ❖ 运行之后，会在右上角菜单栏看到多了一个鲸鱼图标，这个图标表明了 Docker 的运行状态
- ❖ 每次点击鲸鱼图标会弹出操作菜单



- ❖ 在 Windows 11 下安装 Docker
- ❖ 在 Windows 上安装 Docker 之前，需要确认 BIOS 的虚拟化功能已经成功开启
- ❖ 可以使用 Windows 任务管理器的性能标签查看虚拟化是否已开启

The screenshot shows the Windows Task Manager Performance tab. The 'Performance' tab is selected and circled in red. The main display shows CPU performance for an Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz. A line graph shows CPU utilization over 60 seconds, with a current value of 73%. Below the graph, system information is displayed in a table format.

Utilization	Speed	Base speed:	2.59 GHz	
73%	2.98 GHz	Sockets:	1	
Processes	Threads	Handles	Cores:	2
161	1936	54996	Logical processors:	4
Up time	0:00:12:42			
		Virtualization:	Enabled	
		L1 cache:	128 KB	
		L2 cache:	512 KB	
		L3 cache:	4.0 MB	

At the bottom of the window, there are buttons for 'Fewer details' and 'Open Resource Monitor'.

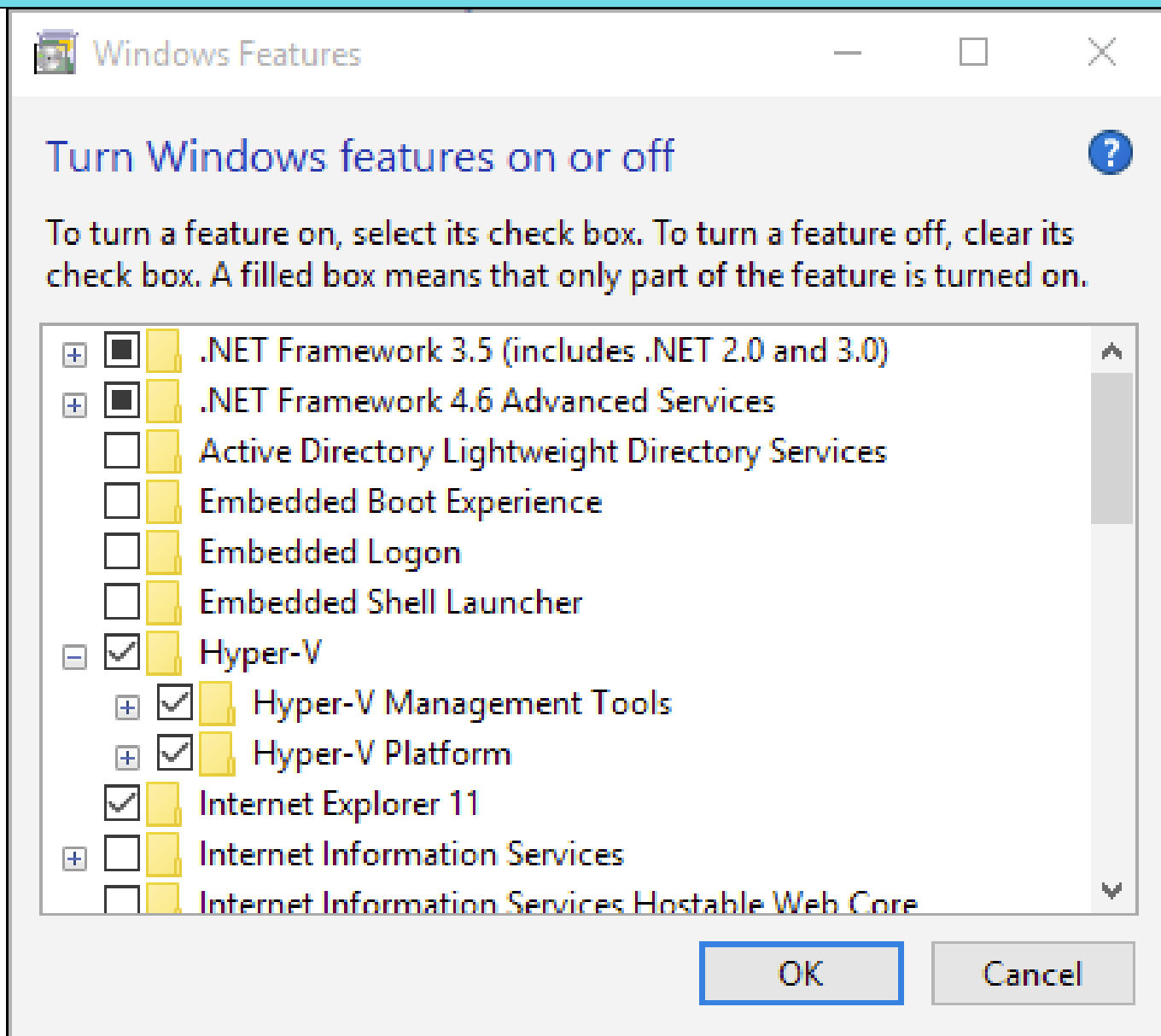
安装 Docker

Windows 11

- ❖ 如未开启，则需要进入 BIOS 开启
- ❖ 开机时按下 DEL 键进入 BIOS
- ❖ 进入 CPU 设置或安全设置
- ❖ 开启虚拟化功能



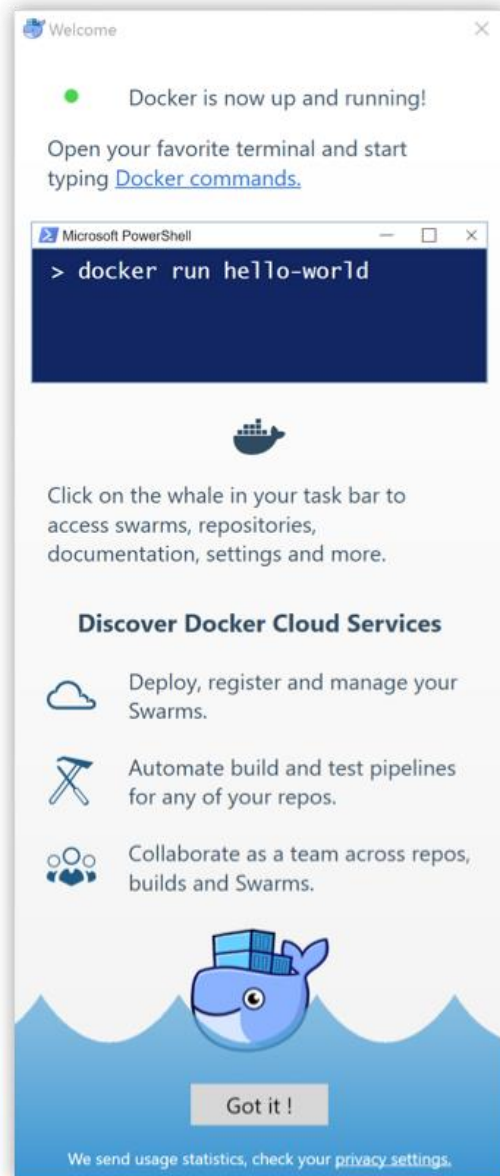
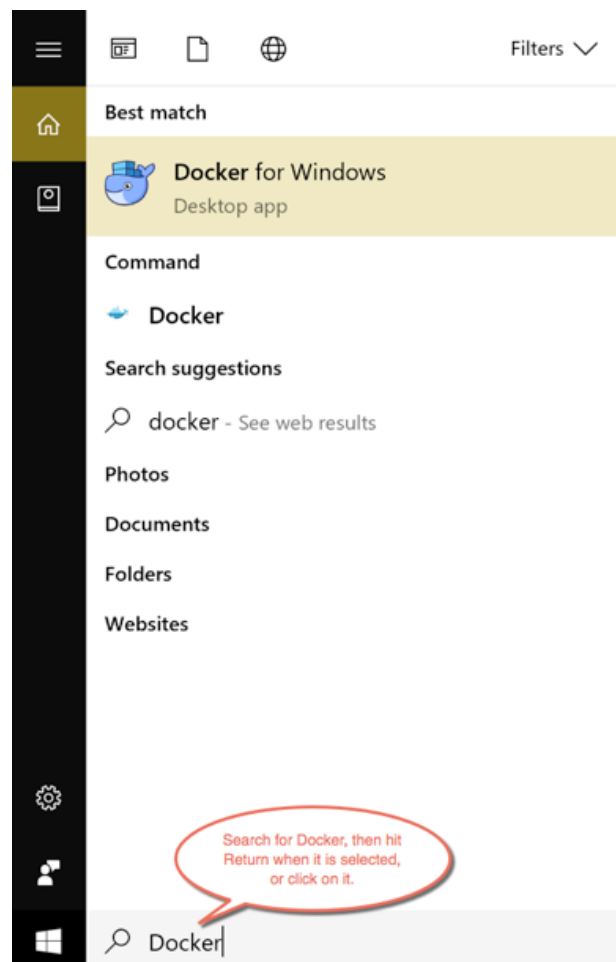
- ❖ 在 Windows 11 下安装 Docker
- ❖ 开启 Hyper-V:
- ❖ 在控制面板“添加 / 删除程序”中“开启 / 关闭 Windows 功能”选项里, 开启 Hyper-V 相关功能
- ❖ 由于需要使用 Hyper-V, Docker 仅支持 64 位的 Windows (Windows 11 只有 64 位版本)



安装 Docker

Windows 11

- ❖ Hyper-V 配置成功后，下载 Docker 安装包安装即可：
- ❖ <https://docs.docker.com/desktop/windows/install/>
- ❖ 安装完毕后，从开始菜单中找到 Docker 图标并点击运行
- ❖ 运行之后，会在任务栏看到多了一个鲸鱼图标，这个图标表明了 Docker 的运行状态
- ❖ 每次点击鲸鱼图标会弹出操作菜单



- ❖ 国内从 Docker Hub 拉取镜像有时会遇到困难，此时可以配置**镜像加速器**
- ❖ 国内很多云服务商都提供了国内加速器服务，例如：
- ❖ Azure 中国镜像：<https://dockerhub.azk8s.cn>
- ❖ 阿里云加速器：<https://cr.console.aliyun.com/cn-hangzhou/mirrors>
- ❖ 七牛云加速器：<https://reg-mirror.qiniu.com>

❖ Ubuntu 22.04

❖ 在 `/etc/docker/daemon.json` 中写入如下内容：

```
{
  "registry-mirrors": [
    "https://dockerhub.azk8s.cn",
    "https://reg-mirror.qiniu.com"
  ]
}
```

❖ 注意，一定要保证该文件符合 JSON 规范，否则 Docker 将不能启动

❖ 然后重新启动服务：

❖ `sudo systemctl daemon-reload`

❖ `sudo systemctl restart docker`

- ❖ macOS 14 / Windows 11
- ❖ 点击 Docker 应用图标
- ❖ 依次选择：Preferences → Daemon → Registry Mirrors
- ❖ 在列表中填写加速器地址，例如：
<https://dockerhub.azk8s.cn>
- ❖ 修改完成之后，点击 Apply & Restart 按钮，Docker 就会重启并应用配置的镜像地址了



- ❖ 检查加速器是否生效
- ❖ 运行指令：
- ❖ `docker info`
- ❖ 如果从结果中看到了类似以下的内容，说明配置成功：

```
Registry Mirrors:  
https://dockerhub.azk8s.cn/
```

❖ docker run hello-world

```
~ » docker run hello-world
```

```
130 ↵ valency@Sakura
```

```
Hello from Docker!
```

```
This message shows that your installation appears to be working correctly.
```

```
To generate this message, Docker took the following steps:
```

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
(amd64)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

```
To try something more ambitious, you can run an Ubuntu container with:
```

```
$ docker run -it ubuntu bash
```

```
Share images, automate workflows, and more with a free Docker ID:
```

```
https://hub.docker.com/
```

```
For more examples and ideas, visit:
```

```
https://docs.docker.com/get-started/
```


❖ docker images

```
~ » docker images
```

```
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
hello-world         latest      bf756fb1ae65     8 months ago    13.3kB
```

```
~ » |
```

- ❖ `docker run -dti -p 22 rastasheep/ubuntu-sshd:18.04`
- ❖ 这行命令会启用一个在后台运行的 Ubuntu 18.04 容器，内置 SSHD

```
~ » docker run -dti -p 22 rastasheep/ubuntu-sshd:18.04 125 ↵ valency@Sakura
Unable to find image 'rastasheep/ubuntu-sshd:18.04' locally
18.04: Pulling from rastasheep/ubuntu-sshd
a48c500ed24e: Pull complete
1e1de00ff7e1: Pull complete
0330ca45a200: Pull complete
471db38bcfbf: Pull complete
0b4aba487617: Pull complete
33aa9e7ef666: Pull complete
39c956d3d54b: Pull complete
40462533cfca: Pull complete
2b1920b96f11: Pull complete
Digest: sha256:1df808ccf95c13b8e62548ad434829bc28ee701a65624e5c0440fc24482e4a62
Status: Downloaded newer image for rastasheep/ubuntu-sshd:18.04
3bf2cb6bb47619d7fb98e31a727e70bcdc568a584ea77dabb335498fa9c36d78
```

❖ docker ps -a

```
~ » docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
cc5da9143794	rastasheep/ubuntu-sshd:18.04	"/usr/sbin/sshd -D"	3 minutes ago	Up 3 minutes
de6a9f7e11f8	hello-world	"/hello"	45 minutes ago	Exited (0) 45 minutes ago

```
STATUS
```

```
Up 3 minutes
```

```
Exited (0) 45 minutes ago
```

```
PORTS
```

```
0.0.0.0:32769→22/tcp
```

```
NAMES
```

```
goofy_meitner
```

```
dreamy_jang
```

```
valency@Sakura
```



- ❖ 我们可以通过 SSH 远程连接到这个容器（密码是“root”）：
- ❖ `ssh -p 32769 root@127.0.0.1`
- ❖ 注意：端口是按照一定规律生成的，每次都不一样，请仔细查看

```
~ » ssh -p 32769 root@127.0.0.1
The authenticity of host '[127.0.0.1]:32769 ([127.0.0.1]:32769)' can't be established.
ECDSA key fingerprint is SHA256:Rath9QRSP1hKeFkIGwL1c1WUV+haHdJrxTyilRrRNnE.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[127.0.0.1]:32769' (ECDSA) to the list of known hosts.
root@127.0.0.1's password:
root@cc5da9143794:~# ls
root@cc5da9143794:~#
```

❖ `docker stop <container-id> && docker rm <container-id>`

```
~ » docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED          STATUS          PORTS                               NAMES
cc5da9143794   rastasheep/ubuntu-sshd:18.04       "/usr/sbin/sshd -D"     15 minutes ago  Up 15 minutes  0.0.0.0:32769->22/tcp              goofy_meitner
de6a9f7e11f8   hello-world                          "/hello"                57 minutes ago  Exited (0) 57 minutes ago                                     dreamy_jang
```

```
~ » docker stop cc5da
cc5da
```

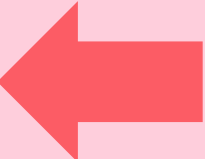

```
~ » docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED          STATUS          PORTS                               NAMES
cc5da9143794   rastasheep/ubuntu-sshd:18.04       "/usr/sbin/sshd -D"     15 minutes ago  Exited (0) 2 seconds ago                                     goofy_meitner
de6a9f7e11f8   hello-world                          "/hello"                57 minutes ago  Exited (0) 57 minutes ago                                     dreamy_jang
```

```
~ » docker rm cc5da
cc5da
```

```
~ » docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED          STATUS          PORTS                               NAMES
de6a9f7e11f8   hello-world                          "/hello"                57 minutes ago  Exited (0) 57 minutes ago                                     dreamy_jang
```

- ❖ 成功创建并远程连接进入课程镜像的容器后，记录容器 ID
- ❖ 打印当前日期和时间：date

```
~ » ssh -p 32770 root@127.0.0.1
root@127.0.0.1's password:
Last login: Sun Sep 27 06:13:03 2020 from 172.17.0.1
root@b744bbf39ed0:~# date
Sun Sep 27 06:15:37 UTC 2020
root@b744bbf39ed0:~#
```



❖ 在作业系统中下载并完成本实验课对应实验报告

❖ <https://hw.dgut.edu.cn/>

❖ **注意：**所有标识为 * 的地方都需要填写

❖ **截止日期：**2024-04-03 23:59:59

课程名称：云计算与大数据应用开发

学期：2023 年春季

实验名称	虚拟化技术			实验序号	1
姓名	***	学号	***	班级	***
实验地点	***	实验日期	***	指导老师	丁焯
教师评语	-			实验成绩	-
				分制	100
同组同学	无				

四、实验作业及分析

4.1 实验过程

1) *** 请将详细实验过程填写在此处 ***

4.2 实验结果

*** 请将实验结果截图填写在此处 ***

五、实验总结

*** 请撰写一段 200 字左右的实验总结 ***

