

环境配置

zsh

[zsh && oh my zsh安装](#)

[代理设置](#)

[powerline安装](#)

tmux

[安装](#)

[oh my tmux](#)

docker

[安装](#)

[docker原理](#)

[docker-compose](#)

kubernetes

Java

[安装](#)

[环境变量设置](#)

Redis

ElasticSearch

tips

[设置进入wsl时位于home目录](#)

[在wsl中打开pdf](#)

ref

[ubuntu 20.04 on WSL](#)

励志成为一个终端玩家，迟早有一天卸载vscode。

zsh

zsh && oh my zsh安装

首先需要安装zsh，由于我的ubuntu的用户hhubibi是后来加进去的，没有shell，所以直接指定了zsh作为我的shell。装好zsh之后从官方github上面把oh my zsh下载下来就行。

Plain Text | 复制代码

```
1 $ sudo apt install zsh
2 $ useradd -m hhubibi -s /bin/zsh
3 $ sh -c "$(wget -O-
  https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"
```

代理设置

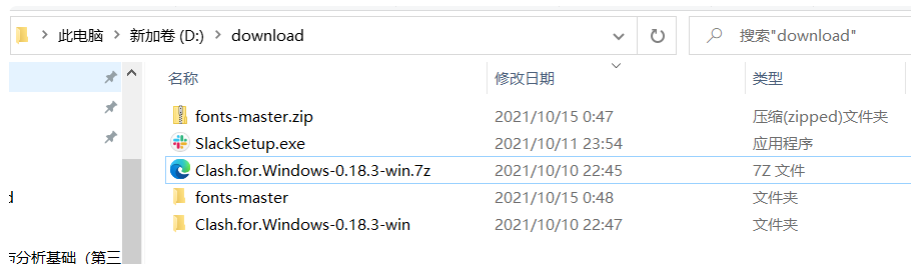
需要导出环境变量，为了方便起见直接在 `~/.zshrc` 里面配置，注意要开启clash的LAN还有代理。

Plain Text | 复制代码

```
1 export hostip=$(cat /etc/resolv.conf |grep -oP '(?<=nameserver\ ).*')
2 export https_proxy="http://${hostip}:7890";
3 export http_proxy="http://${hostip}:7890";
4 export all_proxy="socks5://${hostip}:7890";
```

powerline安装

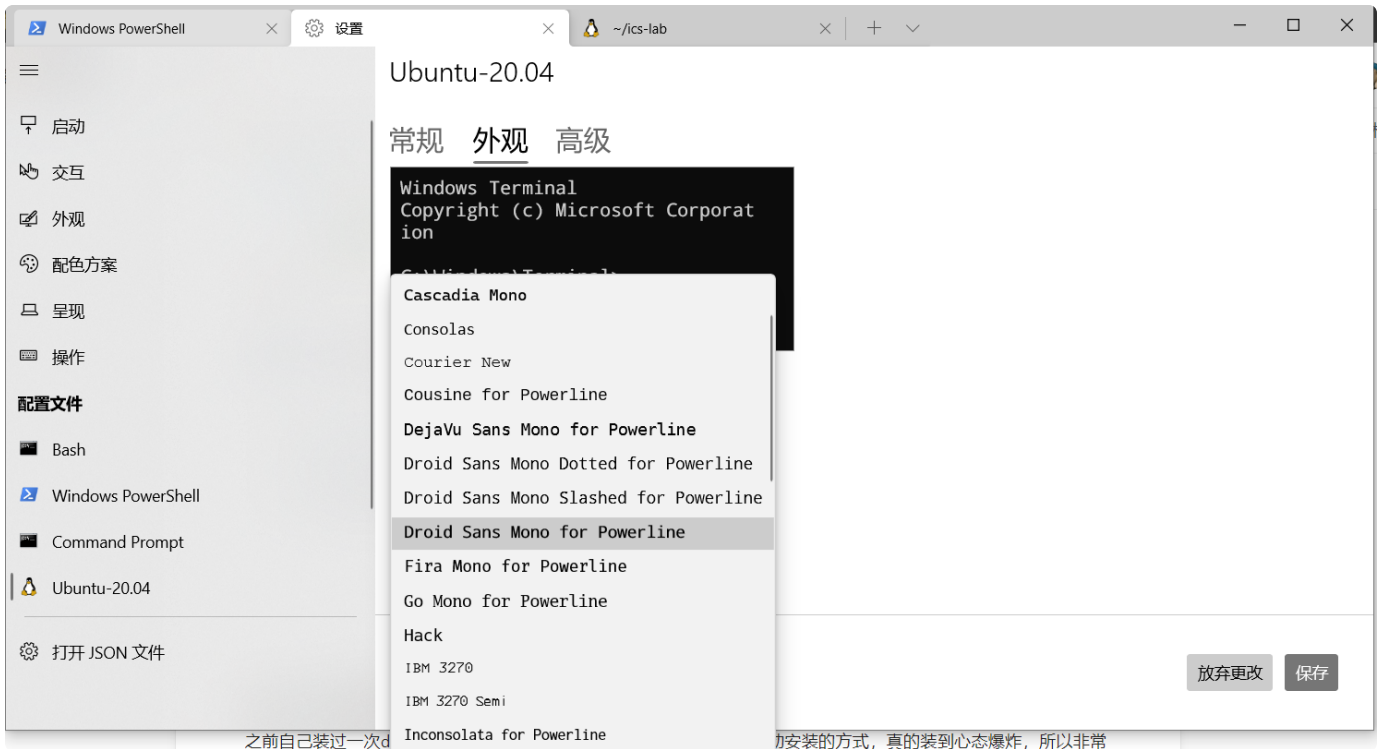
为了使用zsh好看的主题，有必要装一个powerline。从repo (<https://github.com/powerline/fonts>) 把东西下载到本地并解压。在fonts-master下有一个 `install.ps1` 文件，是PowerShell的脚本文件，可以通过运行这个脚本文件来装字体。



然后 `win+x` 以管理员模式打开powershell，一步一步执行下面的指令。

```
1 $ cd D:\downloads\fonts-master # 这是我的install.psl在的目录文件
2 $ Set-ExecutionPolicy Bypass # 不这样脚本不能执行，接下来type选择Y
3 $ .\install.psl # 执行脚本
4 $ Set-ExecutionPolicy Default # 恢复默认
```

然后设置一下想要换的字体。改一下powershell的配色，我用了S



把zsh主题换了，配色也换了

```
Windows PowerShell  设置  ~/ics-lab
hhubibi@DESKTOP-SMVKGMO ~$ ls
code ics-lab
hhubibi@DESKTOP-SMVKGMO ~$ ls
code ics-lab
hhubibi@DESKTOP-SMVKGMO ~$ pwd
/home/hhubibi
hhubibi@DESKTOP-SMVKGMO ~$ ls
code ics-lab
hhubibi@DESKTOP-SMVKGMO ~$ cd ics-lab
hhubibi@DESKTOP-SMVKGMO ~/ics-lab$ ls
README.txt  ans.txt  attacklab.pdf  cookie.txt  ctarget  farm.c  hex2raw  rtarget
hhubibi@DESKTOP-SMVKGMO ~/ics-lab$ vim farm.c
hhubibi@DESKTOP-SMVKGMO ~/ics-lab$ ls
README.txt  ans.txt  attacklab.pdf  cookie.txt  ctarget  farm.c  hex2raw  rtarget
hhubibi@DESKTOP-SMVKGMO ~/ics-lab$ vim farm.c
hhubibi@DESKTOP-SMVKGMO ~/ics-lab$
```

tmux

安装

```
1 $ sudo apt-get install tmux
```

Plain Text | 复制代码

oh my tmux

又是一个生产力工具，在下面这个仓库里，至于如何使用，直接看markdown，非常详细。会用vim的人用tmux问题不大，甚至可以把tmux配的跟vim一样。下面是我配置好的。用了zsh比较火的一个主题。

<https://github.com/gpakosz/.tmux>

```
1 $ cd
2 $ git clone https://github.com/gpakosz/.tmux.git
3 $ ln -s -f .tmux/.tmux.conf
4 $ cp .tmux/.tmux.conf.local .
```

Plain Text | 复制代码

docker

安装

之前自己装过一次docker，在阿里云的服务器上，采用了查看CSDN手动安装的方式，真的装到心态爆炸，所以非常不推荐自己安装，直接用官方脚本安装好了（看看菜鸟教程，感觉还行），省去一堆麻烦事，最好能有美区ip的梯子。

注意docker不能采用直接apt install docker的方式。

Plain Text | [复制代码](#)

```
1 $ curl -fsSL https://get.docker.com | bash -s docker --mirror Aliyun
```

这之后会出现

Plain Text | [复制代码](#)

```
1 → ~ sudo docker version
2 Client: Docker Engine - Community
3 Version: 20.10.9
4 API version: 1.41
5 Go version: go1.16.8
6 Git commit: c2ea9bc
7 Built: Mon Oct 4 16:08:29 2021
8 OS/Arch: linux/amd64
9 Context: default
10 Experimental: true
11 Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the
    docker daemon running?
```

需要设置开机自启动并启动 Docker-ce

Plain Text | [复制代码](#)

```
1 # 下面两行may work, may fail
2 $ sudo systemctl enable docker
3 $ sudo systemctl start docker
4
5 # 如果上面的fail了，就是wsl的问题，ref2说的很清楚，下面这行可以的
6 $ sudo service docker start
```

到目前位置我们的docker都需要sudo权限，如何使docker在普通用户下也能工作（不加前缀sudo），需要把当前用户加入docker组，并更新这个组，docker组在docker安装的时候就已经创建了，没有的

话自己创建一个

Plain Text | [复制代码](#)

```
1 $ sudo groupadd docker # 可选
2 $ sudo usermod -aG docker $USER
3 $ newgrp docker
```

下面是结果，可以看到不用再加sudo了

Plain Text | [复制代码](#)

```
1 → ~ docker version
2 Client: Docker Engine - Community
3   Version:           20.10.9
4   API version:       1.41
5   Go version:        go1.16.8
6   Git commit:        c2ea9bc
7   Built:             Mon Oct  4 16:08:29 2021
8   OS/Arch:           linux/amd64
9   Context:           default
10  Experimental:      true
11
12 Server: Docker Engine - Community
13  Engine:
14    Version:           20.10.9
15    API version:       1.41 (minimum version 1.12)
16    Go version:        go1.16.8
17    Git commit:        79ea9d3
18    Built:             Mon Oct  4 16:06:37 2021
19    OS/Arch:           linux/amd64
20    Experimental:     false
21  containerd:
22    Version:           1.4.11
23    GitCommit:        5b46e404f6b9f661a205e28d59c982d3634148f8
24  runc:
25    Version:           1.0.2
26    GitCommit:        v1.0.2-0-g52b36a2
27  docker-init:
28    Version:           0.19.0
29    GitCommit:        de40ad0
30 → ~ docker images      # 由于还没有装镜像，所以是空的
31 REPOSITORY    TAG        IMAGE ID      CREATED      SIZE
32 → ~
```

这次安装docker非常快，十几分钟吧，第一次装了几个小时...

docker原理

学习docker需要了解linux下的namespace和cgroups机制，这两者为容器提供了隔离和资源管理需要的东西。具体的原理可以阅读：

- docker官方文档推荐的容器原理系列文章（kernel space, container runtimes, container images, container security）：<https://github.com/saschagrunert/demystifying-containers#part-iv-container-security>
- segmentfault上的docker系列：<https://segmentfault.com/a/1190000009309276>

docker-compose

官方安装教程：<https://docs.docker.com/compose/install/>

在wsl下挂梯子也装不成功，一度以为是网络问题，其实是root的问题。对官方命令修改如下：

```
1  $ wget https://github.com/docker/compose/releases/download/1.29.2/docker-  
    compose-$(uname -s)-$(uname -m)  
2  $ sudo mv docker-compose-Linux-x86_64 /usr/local/bin/docker-compose  
3  $ sudo chmod +x /usr/local/bin/docker-compose
```

JSON | [复制代码](#)

之后进行验证：

```
1  $ docker-compose --version  
2  docker-compose version 1.29.2, build 5becea4c
```

JSON | [复制代码](#)

kubernetes

Java

安装

Plain Text | [复制代码](#)

```
1 $ sudo apt-get update
2 $ sudo apt-get install default-jre
3 $ sudo apt-get install default-jdk
```

Plain Text | [复制代码](#)

```
1 → ~ java --version
2 openjdk 11.0.11 2021-04-20
3 OpenJDK Runtime Environment (build 11.0.11+9-Ubuntu-0ubuntu2.20.04)
4 OpenJDK 64-Bit Server VM (build 11.0.11+9-Ubuntu-0ubuntu2.20.04, mixed mode,
  sharing)
5
6 → ~ javac --version
7 javac 11.0.11
```

```
→ code ls
HelloWorld.java
→ code javac HelloWorld.java
→ code java HelloWorld
Hello World!
→ code |
```

环境变量设置

java安装路径

Plain Text | [复制代码](#)

```
1 → ~ sudo update-alternatives --config java
2 There is only one alternative in link group java (providing /usr/bin/java):
  /usr/lib/jvm/java-11-openjdk-amd64/bin/java
3 Nothing to configure.
```

导出环境变量，打开 `~/.zshrc`，加入两行

Plain Text | [复制代码](#)

```
1 export JDK_HOME="/usr/lib/jvm/java-11-openjdk-amd64"
2 export PATH="$JDK_HOME:$PATH"
```

验证一下

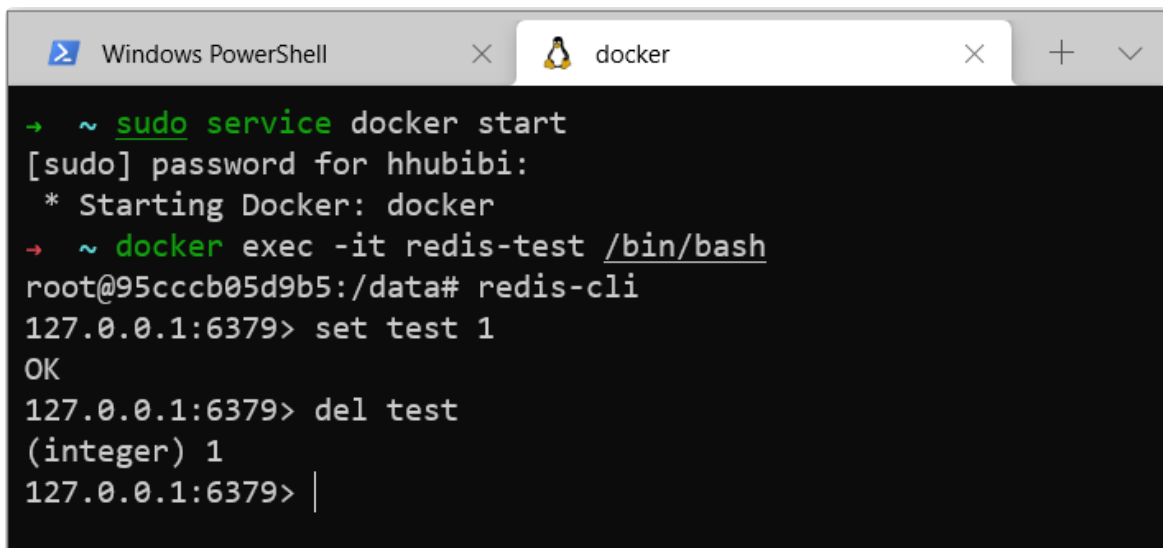
Plain Text | [复制代码](#)

```
1 → ~ echo $JDK_HOME
2 /usr/lib/jvm/java-11-openjdk-amd64
3 → ~
```

Redis

Plain Text | [复制代码](#)

```
1 $ docker pull redis
2 $ docker run -itd -p 6379:6379 --name=redis-test redis
3 $ docker exec -it redis-test /bin/bash
```



The screenshot shows a Windows PowerShell terminal window with two tabs: 'Windows PowerShell' and 'docker'. The 'docker' tab is active, displaying the following commands and output:

```
→ ~ sudo service docker start
[sudo] password for hhubibi:
* Starting Docker: docker
→ ~ docker exec -it redis-test /bin/bash
root@95cccb05d9b5:/data# redis-cli
127.0.0.1:6379> set test 1
OK
127.0.0.1:6379> del test
(integer) 1
127.0.0.1:6379> |
```

ElasticSearch

```

1 $ docker network create elastic
2 $ docker pull docker.elastic.co/elasticsearch/elasticsearch:7.15.1
3 $ docker run --name es01-test --net elastic -p 9200:9200 -p 9300:9300 -e
  "discovery.type=single-node" -d
  docker.elastic.co/elasticsearch/elasticsearch:7.15.1
4
5 $ docker pull docker.elastic.co/kibana/kibana:7.15.1
6 $ docker run --name kib01-test --net elastic -p 5601:5601 -e
  "ELASTICSEARCH_HOSTS=http://es01-test:9200" -d
  docker.elastic.co/kibana/kibana:7.15.1

```

注意一定要加上版本号（直接docker hub上面看最新版本），官方镜像没有latest的tag。

浏览器中输入 <http://localhost:9200>，可以看到

```

1 {
2   "name" : "678b3a1d9b45",
3   "cluster_name" : "docker-cluster",
4   "cluster_uuid" : "rjEDX4bhRqCux69U5enMlQ",
5   "version" : {
6     "number" : "7.14.1",
7     "build_flavor" : "default",
8     "build_type" : "docker",
9     "build_hash" : "66b55ebfa59c92c15db3f69a335d500018b3331e",
10    "build_date" : "2021-08-26T09:01:05.390870785Z",
11    "build_snapshot" : false,
12    "lucene_version" : "8.9.0",
13    "minimum_wire_compatibility_version" : "6.8.0",
14    "minimum_index_compatibility_version" : "6.0.0-beta1"
15  },
16   "tagline" : "You Know, for Search"
17 }

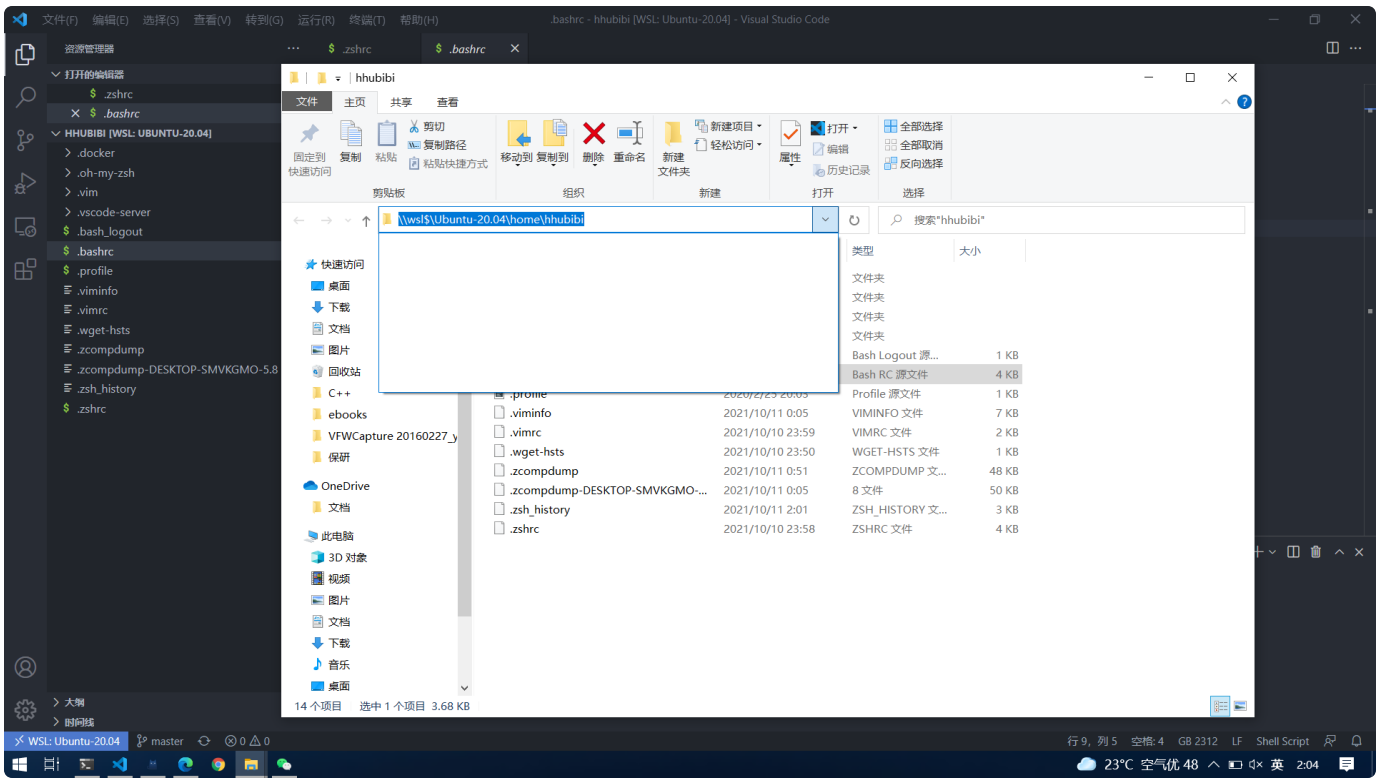
```

在浏览器中输入 <http://localhost:5601>，就可以看到可视化界面了。点击 **Dev Tools -> Console** 就可以输入请求了。

tips

设置进入wsl时位于home目录

可以通过资源管理器查看 `~` 所在的位置，把路径加到 **启动目录** 中即可。



在wsl中打开pdf

对于习惯了终端的用户而言，*keyboard is all you need*. 我们一般习惯把东西放在home目录下，这时候在本地一般找不到文件，所以打开pdf比较头疼，用vscode去连接pdf太麻烦了。所以直接在终端打开pdf就行了。

Plain Text | 复制代码

```
1 $ wslview a.pdf
```

ref

1. <https://www.jianshu.com/p/5bfeb5920fb1>
2. https://blog.csdn.net/qq_43685040/article/details/112056242
3. <https://stackoverflow.com/questions/63148517/how-to-install-powerline-fonts-on-wsl>
4. apt-get彻底卸载软件包: https://blog.csdn.net/get_set/article/details/51276609
5. Docker下安装ElasticSearch和Kibana: <https://segmentfault.com/a/1190000020140461>