HTTP 请求与响应

从入门到工作：HTTP全解

版权声明

联系方式

如果你想要购买本课程
请微信联系 xiedaimala02 或 xiedaimala03

如果你发现有人盗用本课程
请微信联系 xiedaimala02 或 xiedaimala03

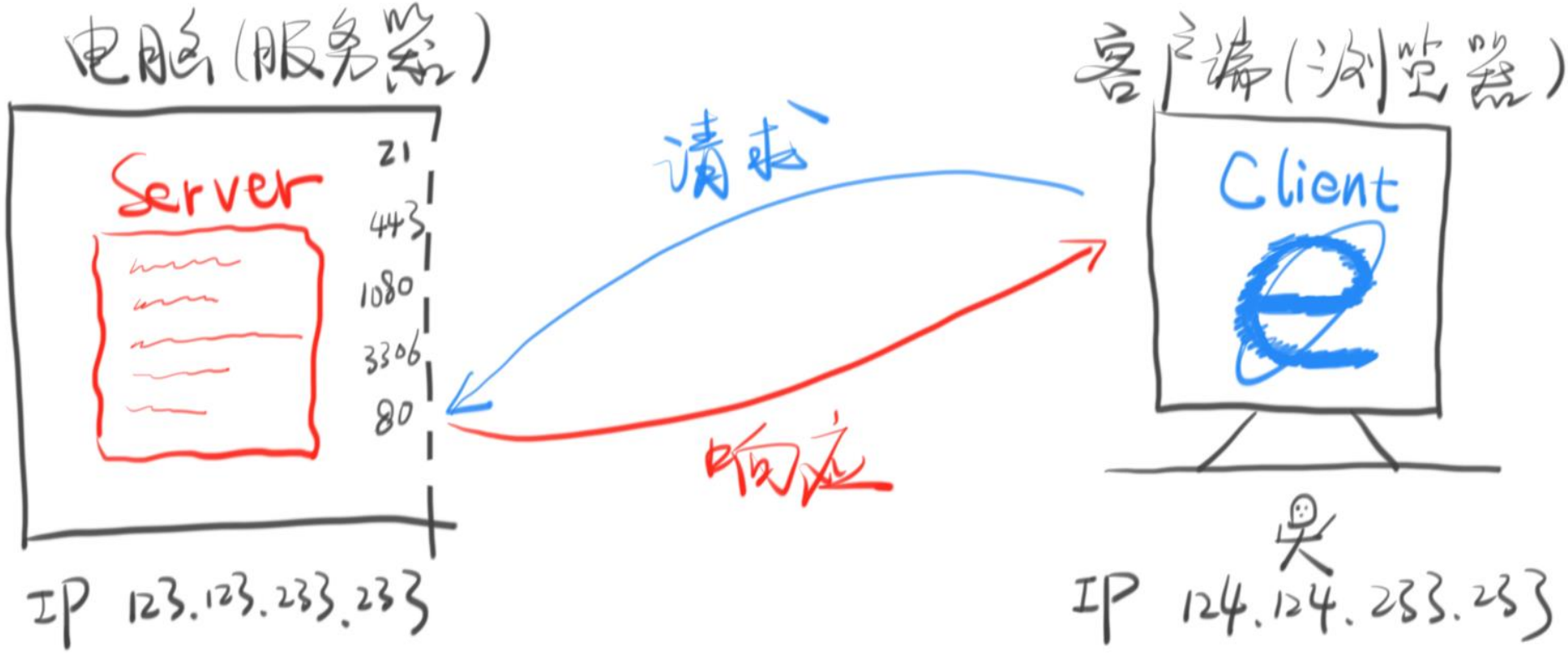
前置条件

- 安装 Node.js 8+
- 理解 IP 和端口
- 理解 URL 路径和查询参数
- 不需要会 JavaScript
- 本节课所有代码建议复制使用

本节课所有代码
都在课程简介有备份

方便大家复制粘贴

请求与响应模型


电脑(服务器)
Server
21
443
1080
3306
80
IP 123.123.233.233
请求
响应
客户端(浏览器)
Client
IP 124.124.233.233

如何发请求

- 方法
  - ✓ 用 Chrome 地址栏
  - ✓ 用 curl 命令
- 概念
  - ✓ 帮你发请求的工具叫做「用户代理」
  - ✓ 英文名 User Agent

如果用chrome，chrome就是用户代理。如果用curl，curl就是用户代理

如何做出一个响应

- 需用编程
  - Node.js 有一个 http 模块可以做到
  - 初始代码已写好，直接用
- 注意事项
  - 这些代码就是服务器代码，一般放在服务器上
  - path 是不带查询参数的路径 /x
  - query 是查询参数的对象形式 {a:'1'}
  - queryString 是查询参数的字符串形式 ?a=1
  - pathWithQuery 是带查询参数的路径，一般不用
  - request 是请求对象
  - response 是响应对象

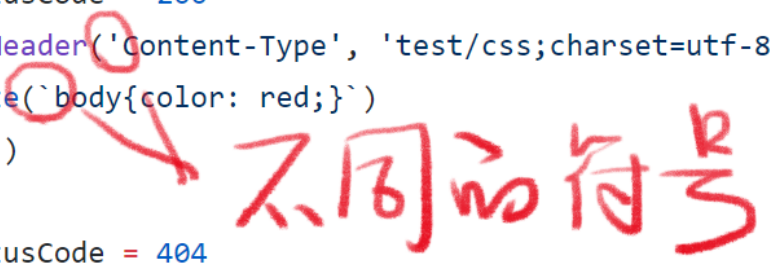
代码

```
/******** 从这里开始看，上面不要看 ************/

console.log('有个傻子发请求过来啦！路径（带查询参数）为：' + pathWithQuery)

if(path === '/'){
  response.statusCode = 200
  response.setHeader('Content-Type', 'text/html;charset=utf-8')
  response.write(`哈哈哈`)
  response.end()
} else if(path === '/x'){
  response.statusCode = 200
  response.setHeader('Content-Type', 'test/css;charset=utf-8')
  response.write(`body{color: red;}`)
  response.end()
} else {
  response.statusCode = 404
  response.setHeader('Content-Type', 'text/html;charset=utf-8')
  response.write(`呜呜呜`)
  response.end()
}

/******** 代码结束，下面不要看 ************/
```



代码逻辑

- 语法
  - ✓ `这种字符串`里面可以回车
  - ✓ '这种字符串' 里面要回车只能用 \n 表示
- 逻辑
  - ✓ 每次收到请求都会把中间的代码执行一遍
  - ✓ 用 if else 判断路径，并返回响应
  - ✓ 如果是已知路径，一律返回 200
  - ✓ 如果是未知路径，一律返回 404
  - ✓ Content-Type 表示内容的「类型/语法」
  - ✓ response.write() 可以填写返回的内容
  - ✓ response.end() 表示响应可以发给用户了

遥想当年李爵士

- 世界上第一个服务器程序
  - 我们也写一个服务器程序
- 世界上第一个网页
  - 我们在 / 路径返回一个 HTML 内容
  - 然后在 /x 路径返回一个 CSS 内容
  - 然后再 /y 路径返回一个 JS 内容
- 注意事项
  - URL 里的后缀卵用没有，/y.css 不一定是 CSS 内容
  - Content-Type 才是决定文件类型的关键

系统学习 HTTP

HTTP 到底是啥

体系化学习

- 必须学会什么
  - ✓ 基础概念（有哪些是必会的）
  - ✓ 如何调试（用的是 Node.js，可以用 log / debugger）
  - ✓ 在哪查资料（用的是 Node.js，看 Node.js 文档）
  - ✓ 标准制定者是谁（HTTP 规格文档：RFC 2612 等）
- 如何学
  - ✓ Copy - 抄文档、抄老师
  - ✓ Run - 放在自己的机器上运行成功
  - ✓ Modify - 加入一点自己的想法，然后重新运行

HTTP 基础概念

- 请求
  - ✓ 请求动词 路径加查询参数 协议名/版本 如：`GET /?wd=hi HTTP/1.1` `GET / HTTP/1.1`
  - ✓ `Host:` 域名或IP
  - ✓ `Accept: text/html` 告诉服务器想接收的内容
  - ✓ `Content-Type:` 请求体的格式
  - ✓ 回车
  - ✓ 请求体（也就是上传内容）
- 细节
  - ✓ 三部分：请求行、请求头、请求体
  - ✓ 请求动词有 GET / POST / PUT / PATCH / DELETE 等 get获取，post上传（面试一定会考）
  - ✓ 请求体在 GET 请求中一般为空
  - ✓ 文档位于 RFC 2612 第五章
  - ✓ 大小写不敏感（随意），最好照着我的写 js敏感，http本身不敏感

HTTP 基础概念

- 响应
  - 协议名/版本 状态码 状态字符串
  - Content-Type: 响应体的格式
  - 回车
  - 响应体（也就是下载内容）
- 细节
  - 三部分：状态行、响应头、响应体
  - 常见的状态码是考点
  - 文档位于 RFC 2612 第六章

用 curl 构造请求

```
curl -v -X POST -H 'Accept:text/html' -d '请求内容'  http://localhost:8888/xxx?wd=hi
```

- ✓ curl -v http://127.0.0.1:8888

- 设置请求动词
- ✓ -X POST
- ✓ 注意大小写

- 设置路径和查询参数
- ✓ 直接在 url 后面加

- 设置请求头
- ✓ -H 'Name: Value' 或者 --header 'Name: Value'

- 设置请求体
- ✓ -d '内容' 或者 --data '内容'

用 Node.js 读取请求

- 读取请求动词
  - request.method
- 读取路径
  - request.url 路径，带查询参数
  - path 纯路径，不带查询参数
  - query 只有查询参数
- 读取请求头
  - request.headers['Accept']
- 读取请求体
  - 比较复杂，先不讲

用 Node.js 设置响应

- 设置响应状态码
  - response.statusCode = 200
- 设置响应头
  - response.setHeader('Content-Type', 'text/html');
- 设置响应体
  - response.write('内容')
  - 可追加内容

代码细节先不管

直接抄，就是干

如何调试

怎么知道自己写错了还是写对了

console.log
调试大法

我不要你觉得，我要~~我觉得~~

我要 console.log 觉得

debug 就是不断质疑自己的过程

下面代码哪里有问题

```
if(path === '/'){
  response.statusCode = 200
  response.setHeader('Content-Type', 'text/html;charset=utf-8')
  response.write(`
    <link rel="stylesheet" href="./style.css">
    <h1>你好</h1>
  `)
  response.end()
} else if(path === './style.css'){
  response.statusCode = 200
  response.setHeader('Content-Type', 'text/css;charset=utf-8')
  response.write(`h1{color: red;}`)
  response.end()
} else {
  response.statusCode = 404
  response.end()
}
```

出 bug 就是因为你太自信了

找到 bug 之时，就是你发现自己时傻 X 之日

console.log 可以验证对错

不要相信自己，要相信 console.log

购买一台服务器

阿里云按量付费，即用即停

为什么要用收费的服务器

- 好处
  - ✓ 一个你可以完全自由掌控的 Linux 机器
  - ✓ 一个其他人可以访问的 IP
  - ✓ 可以作为你的博客、作品展示、简历展示
  - ✓ 比 GitHub 访问速度快很多
  - ✓ 香港地区机器，可以作为FQ代理，但是好像不能备案
  - ✓ 如果你备案了，你还可以把你的域名绑定到这台机器
- 代价
  - ✓ 100 人民币可以用 1000 小时左右
  - ✓ 可以和同学合买，但是人数不能太多，因为如果忘了关闭机器，100 元就用完了
  - ✓ 之前饥人谷提供免费的，很快 CPU 就不堪重负了

购买阿里云服务器

- 步骤

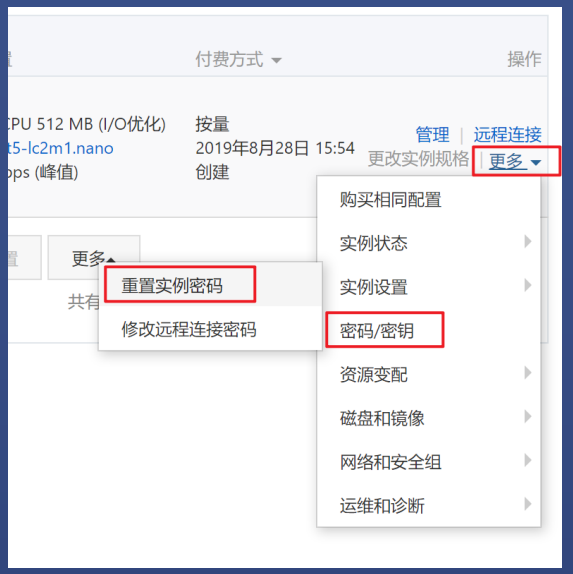
- ✓ 注册账号
- ✓ 按照阿里的要求提供手机、身份信息（不用来问我）
- ✓ 进入云服务器 ECS（找不到可以搜索）
- ✓ 创建实例 => 按量付费 => 入门级 => 选最便宜
- ✓ 镜像选择 Ubuntu 18.04 64位，因为最容易搜教程
- ✓ 其他都不用改，下一步
- ✓ 充值 100 元（如果嫌贵大家可以2到5人拼着买）
- ✓ 公网带宽选择按使用流量 + 1 Mbps
- ✓ 其他不改，下一步
- ✓ 勾选服务协议，创建实例，弹出界面点击管理控制台
- ✓ 等待实例启动成功，状态变为运行中
- ✓ 购买完毕

允许 8888 端口

- 加入安全组

- ✓ 按右图点击加入安全组
- ✓ 如果没有搜到安全组，就先新建一个
- ✓ 加入之后，点击安全组配置
- ✓ 点击配置规则
- ✓ 点击添加安全组规则
- ✓ 端口填写 8888/8888（任意改，也可填写 8880/8889）
- ✓ 授权对象填写 0.0.0.0/0 表示任何人都能访问
- ✓ 点击确定，并输入手机验证码
- ✓ 这个时候你就能通过 http://实例ip:8888 访问了

使用服务器



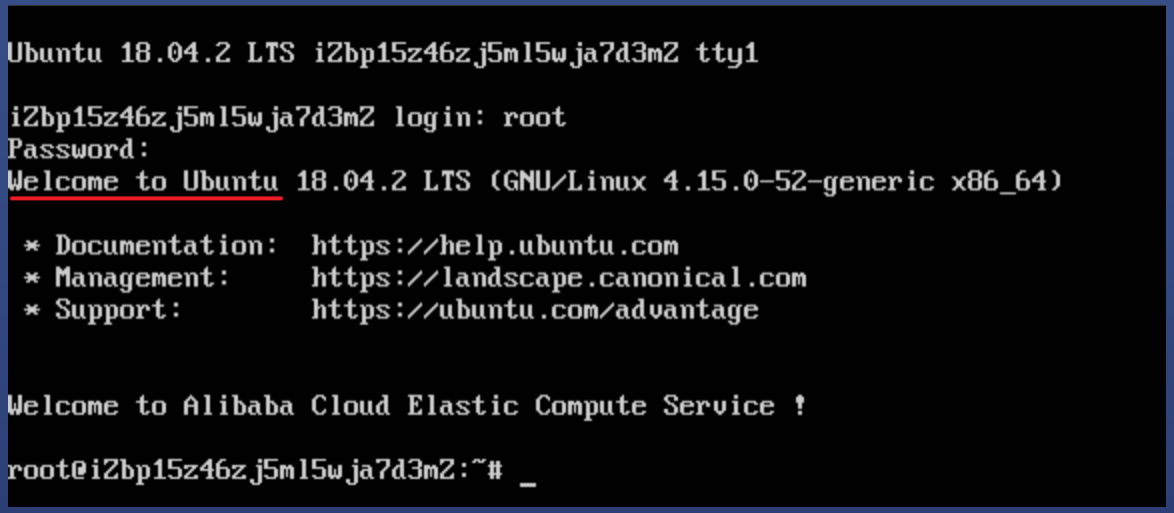
- 步骤

- ✓ 重置实例密码（见右图）
- ✓ 填写新的密码两次和手机验证码
- ✓ 新的密码需要大小写和数字
- ✓ 推荐写法：女朋友名字+你的名字+相识年份
- ✓ 比如：XyjyFyh2018
- ✓ 实例状态 => 重启实例
- ✓ 重启完了之后点击远程连接
- ✓ 把远程连接密码复制下来，假设是 666666
- ✓ 在实例上面新建标签，键为 key 值为 666666
- ✓ 再次点击远程连接，输入连接密码
- ✓ login: 后面输入 root
- ✓ password: 后面输入密码，输入的时候没有反应正常

你已经有一台 Linux 机器

不管是阿里云上的，还是本地的虚拟机

ssh 远程登录



- 步骤

- ✓ 首先登录 root 账户（阿里云网页登录或者其他方式登录均可）
- ✓ 成功看到 Welcome 之后
- ✓ echo '复制本地~/.ssh/id_rsa.pub内容' >> ~/.ssh/authorized_keys
- ✓ 我建议你把上面命令在 VSCode 上编辑好了再复制
- ✓ 新建终端，在本地运行 ssh root@实例ip
- ✓ 此时，你就可以在本地终端操作云服务器了！

成功界面

```
Fang@DESKTOP-P0DGHMB ~/Jirengu
$ ssh root@121.40.162.215
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-52-generic x86_64)

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


Welcome to Alibaba Cloud Elastic Compute Service !
```

- 刚才做了什么

✓ 把本地的公钥复制到阿里云的 ~/.ssh/authorized_keys
✓ 在本地用 ssh root@实例ip 来远程操作云机器
✓ 以后都用 ssh root@实例ip 的方式即可
✓ 你可以在 hosts 里给实例ip 取个别名
✓ 想要退出云机器，可以输入 exit 回车
✓ 如果卡了，可以直接退出终端 ctrl + W

如何防止 ssh 卡住

- 新人看不懂可以跳过这一张PPT

✓ 在 /etc/ssh/ssh_config 最后加下面两句话

```
Host *
  ServerAliveInterval 30
```

✓ 用 code 打开这个文件应该会失败

✓ 可以用两次 echo 搞定

```
echo "Host *" >> /etc/ssh/ssh_config
echo "  ServerAliveInterval 30" >> /etc/ssh/ssh_config
```

✓ 不要少写了空格

✓ 重启终端生效，如果没生效就重启机器

✓ 我是从这里抄的命令，有问题不要问我

✓ 工具嘛，能用就行

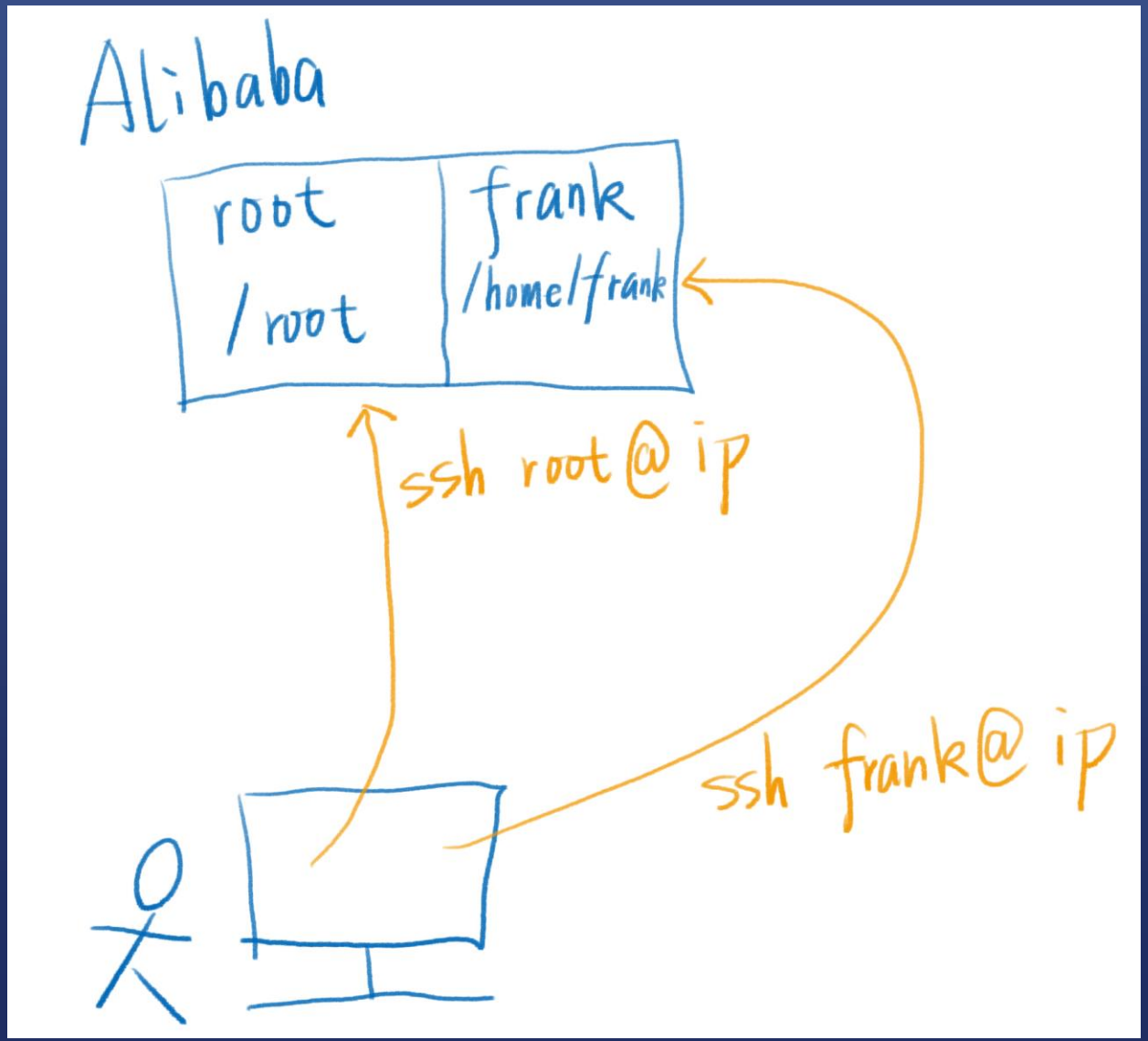
创建应用账户

- 为什么
  - Linux 的 root 账户拥有最高权限，一旦被攻克……
- 步骤
  - adduser frank
  - Enter new UNIX password: 输入密码
  - 再次输入密码，密码可以跟 root 的密码一样
  - 一直回车，直到结束，运行如下命令

```
mkdir  /home/frank/.ssh
cp ~/.ssh/authorized_keys /home/frank/.ssh/
chmod 755 /home/frank/.ssh/authorized_keys
chown frank:frank /home/frank/.ssh/authorized_keys
```

  - 现在 ssh frank@实例ip 就可以使用了

多账户示意图



sudo

- 给 frank 添加 sudo 权限
  - adduser frank sudo
- sudo 是什么
  - 全称 super user do
  - 类似于 Windows 的「以管理员身份运行」
  - 平时你不应该使用 root 账户，而是用 frank 账户
  - 遇到特殊操作，就在前面加 sudo，请出 root
  - 需要输入 frank 的密码，不是 root 的密码
  - 使用 ctrl + A 可以快速回到命令前面
  - sudo !! 的意思是用 sudo 执行上一句命令

安装 Node.js 8

- 步骤

- ✓ curl -sL https://deb.nodesource.com/setup_8.x | sudo bash -
- ✓ sudo sed -i 's/deb.nodesource.com\/node_8.x/mirrors.tuna.tsinghua.edu.cn\/nodesource\/deb_8.x/g' /etc/apt/sources.list.d/nodesource.list
- ✓ sudo apt-get update
- ✓ sudo apt-get install -y nodejs
- ✓ node -v
- ✓ npm -v
- ✓ npx -v

安装 git

- 命令
  - sudo apt install git
  - 遇到 [Y/n] 输入回车，或者 y 回车
  - git --version
- 注意
  - 由于无界面环境，安装不了 VSCode，vim 又太难
  - 所以不要使用 git commit -v 改用 -m "xxx"
  - 如果你想学习 vim，在本地电脑输入 vimtutor 看完
  - 新人不要浪费时间学习 vim
  - 遇到解决不了的问题，请自行搜索报错信息
  - 实在不行，就去阿里云页面重启机器即可

部署应用

- 下载代码
  - git clone https://github.com/FrankFang/nodejs-test.git 这个地址可以改成你的仓库的 https 地址
  - 只是下载，所以不要使用 ssh 地址，使用 https
- 启动应用
  - `cd nodejs-test`
  - `touch log`
  - 启动命令：`node server.js 8888 > log 2>&1 &`
  - 把启动命令做成 `start` 文件
  - 添加执行权限 `chmod +x ./start`
  - 运行 `sh ./start` 得到一个进程号 `pid`
  - `tail log` 看 `log` 内容
  - `kill -9 pid` 可以关掉进程
  - `killall node` 可以关掉所有 `node` 进程

如何重启应用

- 上传代码
  - ✓ 在本地改完代码
  - ✓ git push
- 下载代码
  - ✓ ssh frank@实例ip
  - ✓ cd nodejs-test
  - ✓ git pull
  - ✓ killall node（因为忘了进程号，实际上可以记下来）
  - ✓ sh ./start
  - ✓ 重启完毕

回顾一下

- 请求和响应分别有哪几部分
- 请求动词有哪些
- 状态码有哪些
- 用 curl 构造请求
- 用 Node.js 读请求，造响应
- HTML / CSS / JS 的本质都是字符串
- 部署 Node.js 应用到云服务器

再见

接下来进入 JS 的学习