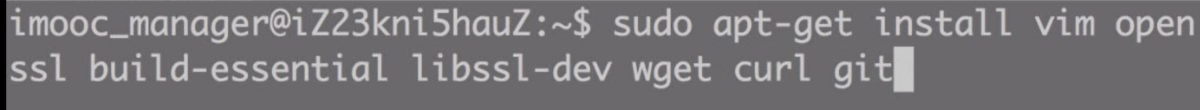
服务器搭建nodejs 环境

Sudo apt-get update 更新环境

安装软件



用nvm 来管理npm 版本

Github上找到nvm

Nginx实现反向代理

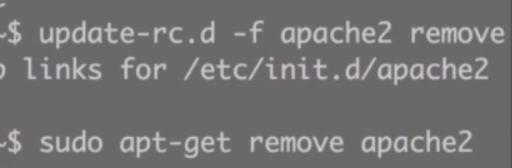
目前只是设置了8081端口可以访问，还不能访问80端口，可以通过nginx来将80端口的请求反向代理转发到node启动的8081端口

登陆服务器：



检查是否安装apach e





然后更新包列表

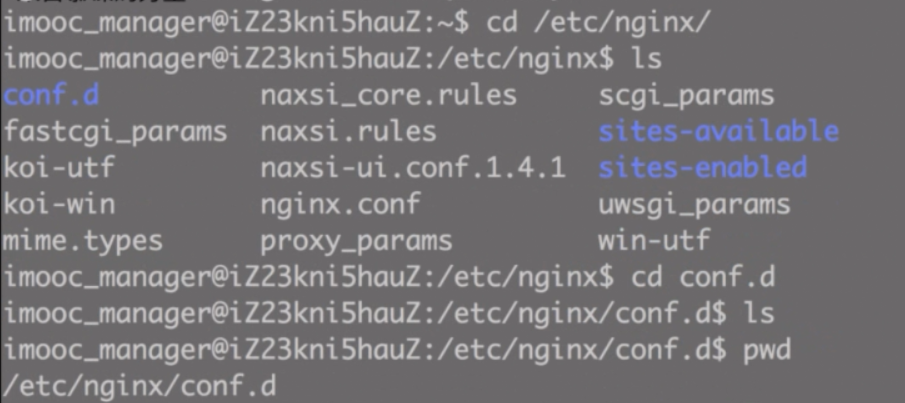
Sudo apt-get update

安装nginx

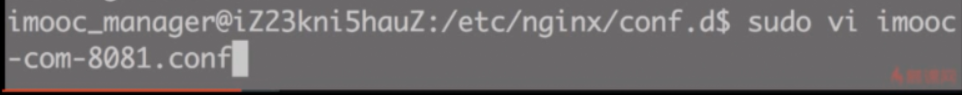
Sudo apt-get install nginx

查看版本

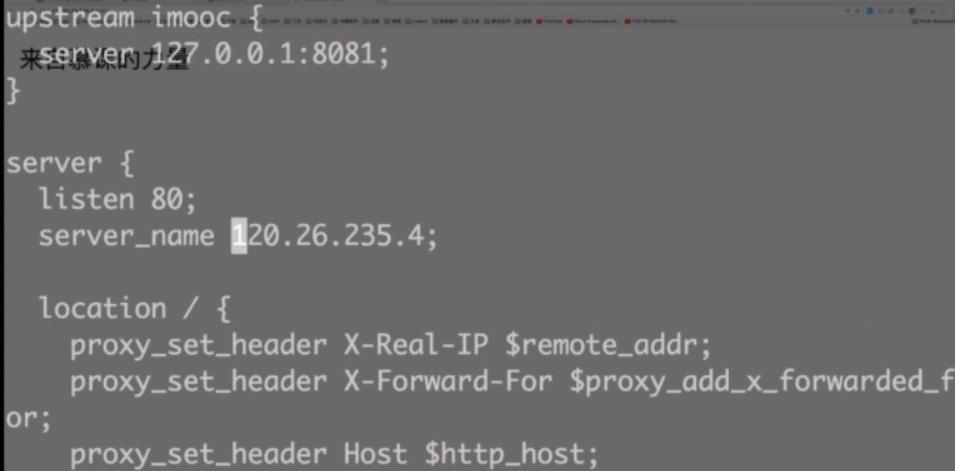
Nginx –v

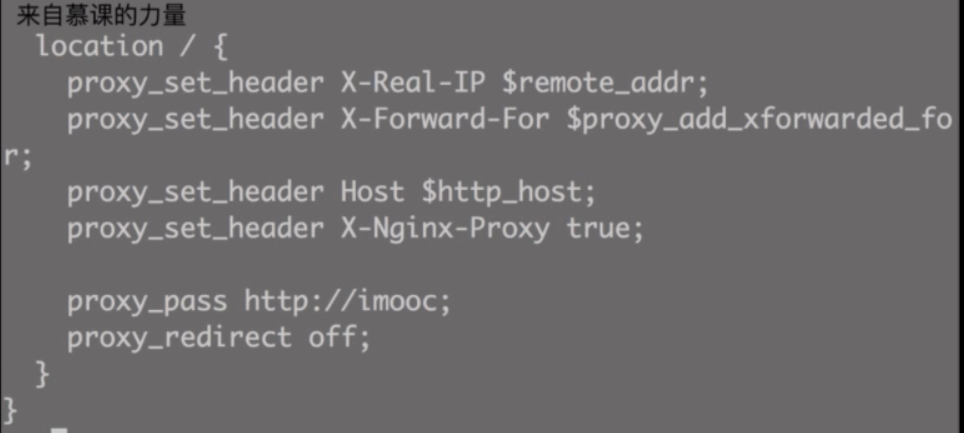


到conf.d 文件夹建立配置文件 imooc-com-8081.conf

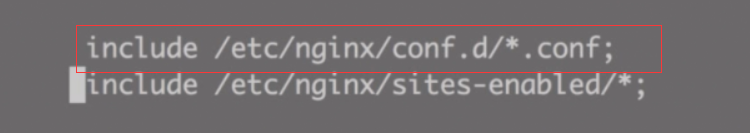


编辑配置文件





返回到nginx 目录，找到nginx.conf文件



然后检测nginx配置是否有误

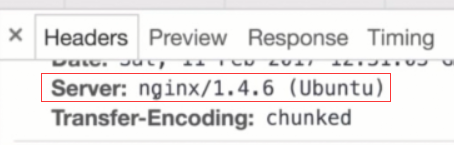
Sudo nginx –t

没问题的话：

Sudo nginx reload

这样就能通过ip地址直接访问80端口

在headers里可以看到服务器server信息

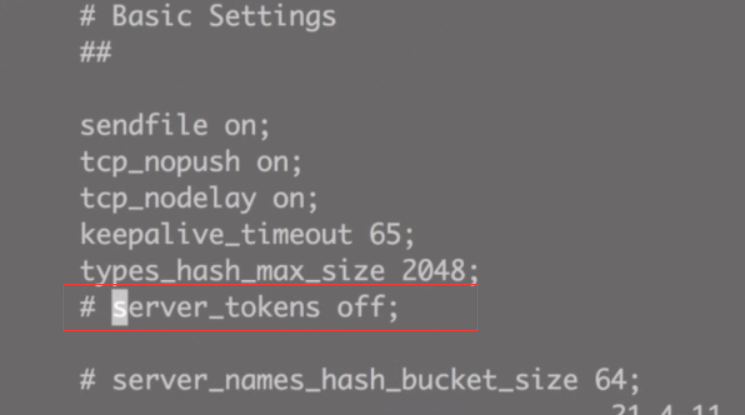


如果想隐藏这些信息

进入到nginx中， 编辑nginx.conf

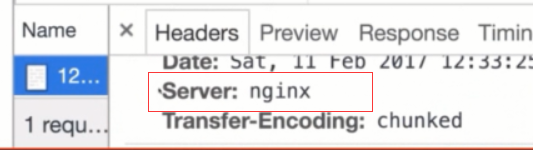


找到http中设置



去掉，然后重启

Sudo service nginx reload



这样就没有版本信息，对外不是那么透明了

利用DNSPod管理域名解析

1. 域名是需要一个域名解析服务器 DNS解析
2. 一个域名只能对应一个ip地址，但一个ip地址（服务器）可以对应了多个域名，（赌博网站搜索被降权，你也受牵连）

可以通过dnspod将域名管理统一到一起





Cname: 域名，哪怕服务器地址发生变化，不用频繁切换ip地址

TTL: 600

A记录和CNAME

五个二级域名的 都映射到同一个服务器

五个二级域名的 都映射到同一个服务器

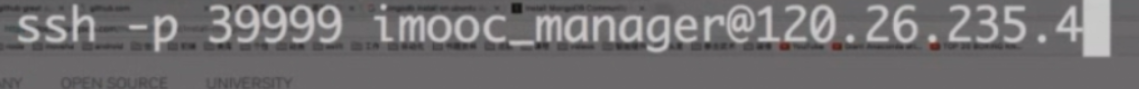


服务器配置和MongDB 安装

一般数据和应用要分别放在不同服务器

1. 安装mongodb

登录服务器



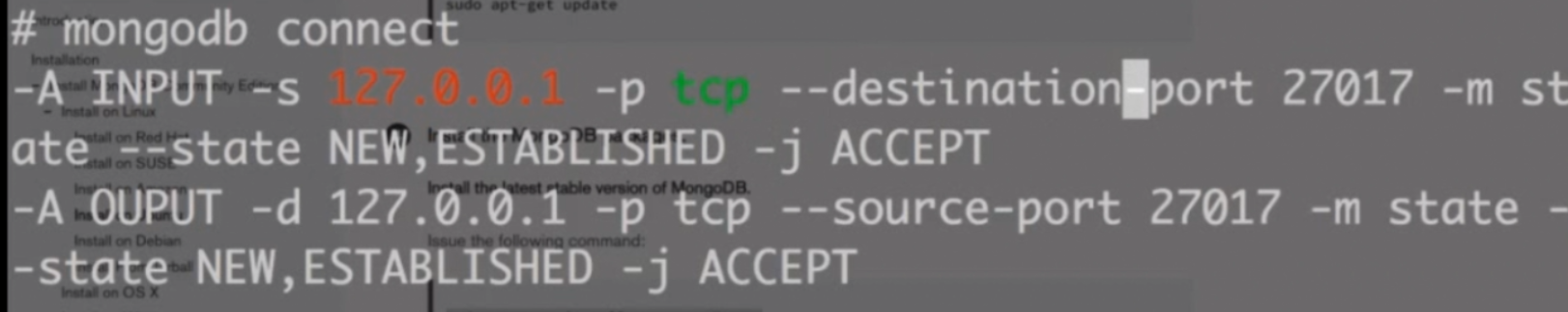
**查看mongdb官方文档安装即可**

………

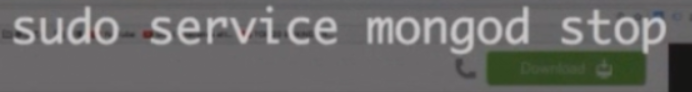
安装成功后可以测试



开启127.0.0.1 防火墙



停止



修改mongdb的默认端口27017，改成19999

。。。。

**数据库导入导出**

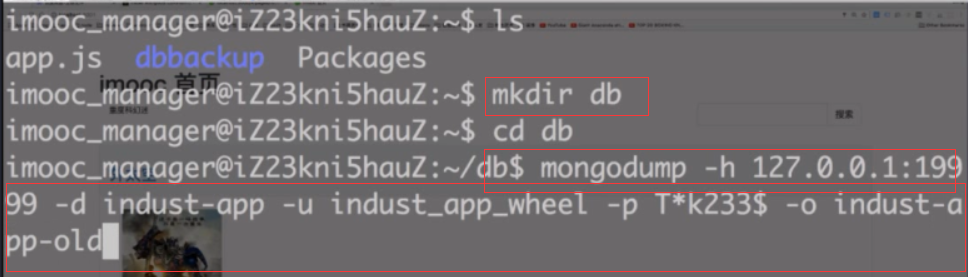
1. 如何从一台线上一台数据库迁移到另外一台服务器？

比如从一台老的服务器的数据迁移到一台新的服务器上呢？

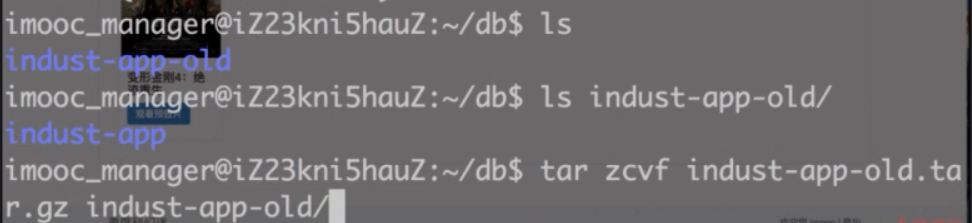
前提是把用户 账号 密码这些都设定好了，这里不再细说，可以查看以前的视频

示例，建立db文件夹，备份数据库 indust-app 到indust-app-old 文件夹

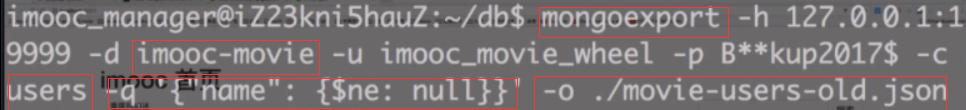
**线上操作一个数据库**



**压缩打包**



**线上导出一张表**



指定哪个数据库下面的 那个表，可以跟上查询条件 –q 到指定的文件

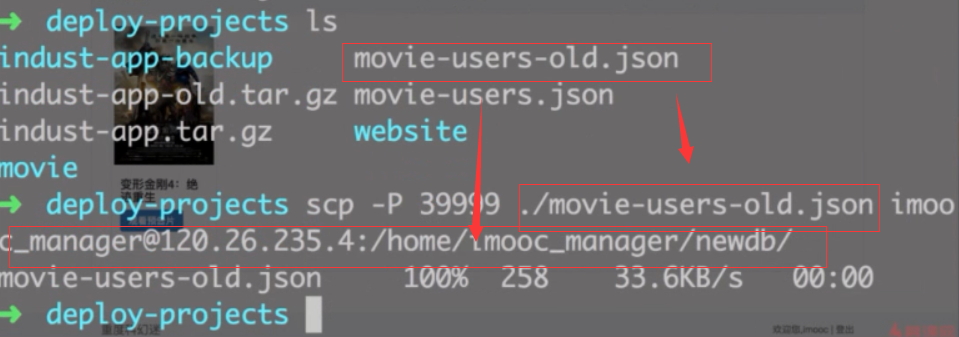
**如何从本地的数据迁移到线上？？？？？**

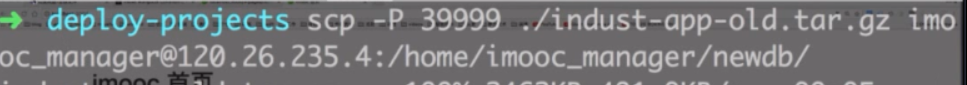
先从服务其上下载一些数据 一个数据库的tar包和一张user表



将下载到本地的数据在上传到新的目标服务器

在服务器上建立一个newdb文件夹来模拟存放上传的文件





**数据库备份**

可以通过脚本来设置定时任务 容灾机制 去备份数据库 这是一种比较简单实用的

**方法**

在目录下建立一个task 目录

Mkdir task

重命名文件夹 tasks

Mv task tasks

进入tasks

Cd tasks

建立备份的脚本文件movie.backup.sh

Vi movie.backup.sh

首先建立电影网站的备份脚本

#！表示是一个可执行脚本

申明变量 backUpFolder备份的路径 date\_now 时间格式 backFileName备份的文件名

现在可以编写可执行脚本了

1. 进入备份目录

Cd $backUpFolder

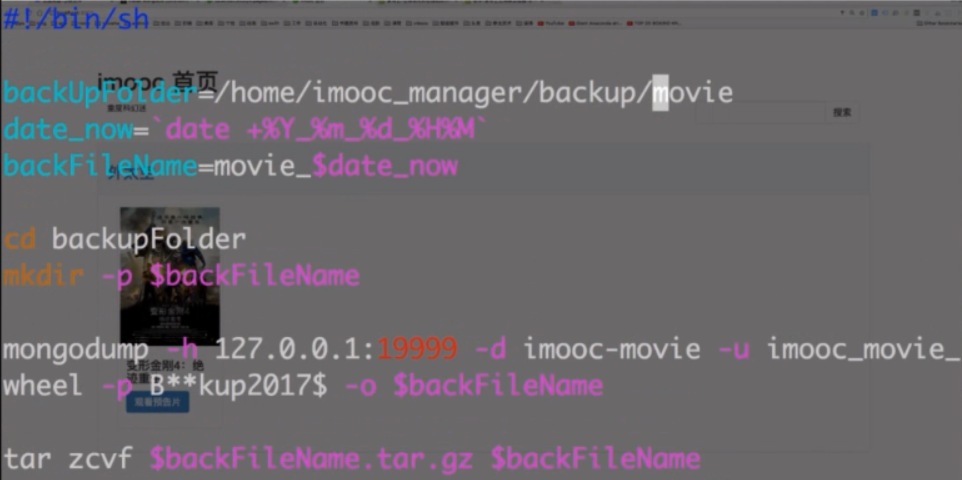
1. 建立一个备份的文件夹

Mkdir –p $backFIleName

1. 进行mongdb数据库的备份

Mongodump –h 主机名 –d 数据库 –u 用户 –p 密码 –o 备份到指定的文件夹

1. 用tar zcvf 压缩打包的文件名.tar.gz 到打包的文件夹

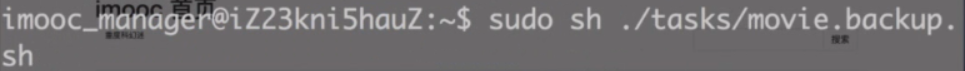




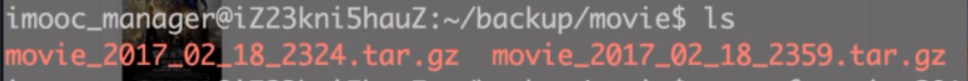
上图错误 cd $backUpFolder(￥ 大写U)

Cd到**用户**的根目录，创建backup文件夹 创建movie文件夹

Cd到更目录执行sh脚本



可以到指定目录下查看备份文件是否成功

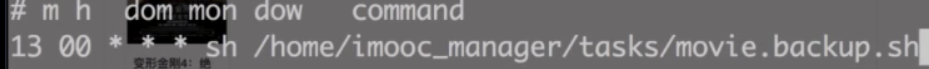


下一步是指定系统的定时备份任务

通过crontab –e 进入nano的编辑器



可以了解crantab的一些定时任务，编辑系统命令，保存并退出



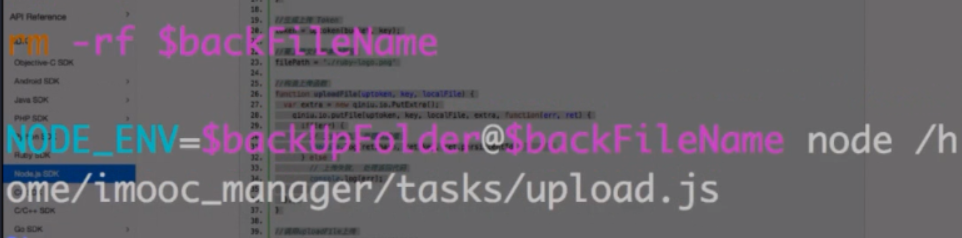
备份到本地后，这些还是存在一些不稳定，最好把数据备份一份到云端比如七牛云

可以通过node.js 来执行备份脚本

具体可以看qiniu的官方示例，如果有不懂，在倒回来看看

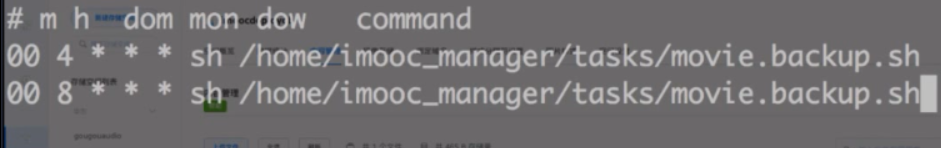
建立一个js脚本 uploade.js

在sh脚本中增加一行执行命令



同时设定在凌晨4点和8点执行一次备份

Crontab



这样就完成了 项目数据库的打包 备份和上传到私有云

一个数据库最好创建一个sh 脚本 这样就不会因为一个备份失败而相互影响

如果要安装mysql怎么办呢？

其实是一样的，其他的备份和导入导出可以查看相关的文档

Sudo apt-get install mysql-server mysql-clilent

Mysql –u –root –p