服务器

1.nginx 目录

/usr/local/nginx

2.nginx 最新映射

m.jingdakongjian.com / 443 proxy\_pass http://127.0.0.1:8080

www.jingdakongjian.com/ 443 http://127.0.0.1:8080/pc/index;

erp.jingdakongjian.com / 80 proxy\_pass http://127.0.0.1:8088;

data.jingdakongjian.com /shequ 443 proxy\_pass http://127.0.0.1:8089; 社区server

data.jingdakongjian.com /solr 443 proxy\_pass http://127.0.0.1:8983;

data.jingdakongjian.com / 443 proxy\_pass http://127.0.0.1:8089/shequ/

data.jingdakongjian.com /prize 443 proxy\_pass http://127.0.0.1:8081; prize server

data.jingdakongjian.com /prize\_h5 443 path; prize h5

3.tomcat服务管理

8080 代拿server

8088 代拿后台

8089 shequ server

8081 prize server

4.快速部署

service tomcat8080 publish /export/data/publish\_war/feimaotui.war

service tomcat8088 publish /export/data/publish\_war/feiamotui\_boss.war

service tomcat8089 publish /export/data/publish\_war/shequ.war

service tomcat8081 publish /export/data/publish\_war/prize.war

5.系统重启自动执行脚本

配置/etc/rc.local即可

现在rc.local中添加了exec /export/sh/boot\_my.sh ，然后在boot\_my.sh 添加了

#zookeeper 开机启动 /export/servers/zookeeper/zookeeper-3.4.14/bin/zkServer.sh start

zkServer.sh start

#redis

/usr/local/redis/bin/redis-server /usr/local/redis/etc/redis.conf

#tomcat

service tomcat8080 restart

service tomcat8088 restart

service tomcat8089 restart

#nginx

service nginx

6.mysql

password:Mjj!@#7654322

7.java文件执行

-cp 和-classpath一样；.代表当前目录 ：分隔符

nohup java -classpath .:lib/\* org.egg.utils.Test &

如果有conf

nohup java -classpath .conf:.lib/\* org.egg.utils.Test &

8.ab压测post

ab -n 1 -c 1 -p post.txt -T application/json http://10.13.144.219:8080/asr/d/102

post.txt为当前命令路径下的文件 内容为body json格式

# 头脑风暴：

开发测试部署影响问题

1.单机情况下，开发部署会导致服务不可用，是否可以平滑部署，非集群方式

方案：

部署时，copy临时容器来兼容老服务，待新服务可用时，毫秒级别切换流量

2.单机情况下,部署错误版本导致服务不可用（启动失败），平滑快速回滚稳定版本。

linux上可视化的操作

问题背景：

docker k8s 操作命令太长 记忆和操作成本高

方案：

可视化 用户选择不同指令【数字指令】来执行常用命令

可以自定义指令

扩展：

其他linux命令同可以如此优化，比如：

awk,定位日志，自定义全局指令

代码jvm风险预测

1.如果达到某个条件 会出现什么问题

2.通过加速器和压测来提前预测

3.可选择使用独立的沙盒环境去预测，模拟生产环境的现有的环境指标，cpu,mem,disk,io速度，网络延迟。

4.jvm宕机原因分析，建议修改方案。

自研监控平台 jvm（gc频率，oom,）,mem, io top cpu 智能分析（分析瓶颈原因：ncng top999,load,etc） 先出脑图【vm/docker版】；自动智能jvm调优推荐分析 打造中小企业的开源的智能分析工具

#适用于新手的

5.高并发工具包，常见问题的高效解决方案

6.mock ide插件，无代码侵入 支持某个全类目录下的某个方法。

#自定义的saas

zk redis db mq job 分布式数据库中间件 监控平台 dubbo/spring-cloud

swagger,

docker

springboot

mvc框架

#压测，性能分析门槛高，链路追踪分析，

日志平台

监控平台

代码仓库

maven私服

部署平台

针对于小公司低成本的开发，运维项目

#痛点

开发人员压测成本高

生产环境定位问题成本高

压测监控命令

top -H -p pid 可以查看线程数，CPU使用率，内存情况，负载

#查看某个进程的线程数

top -H -p pid

#TP999 CPU使用率(top) IO使用情况（yum -y install iotop） load(top) 内存使用情况

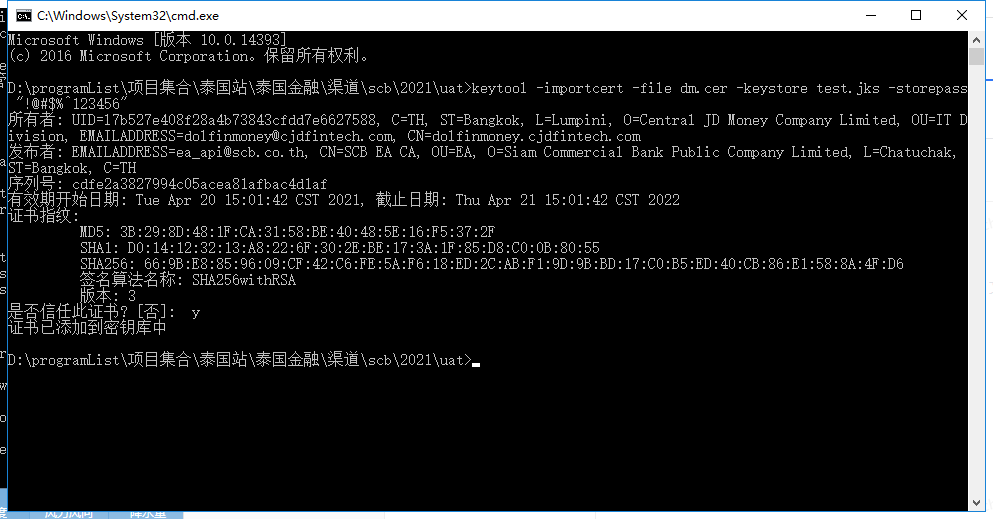
io是否哟瓶颈 iostat -x 1

1秒一次打印Io使用率 %util 如果是100%代表饱和

证书keytool命令

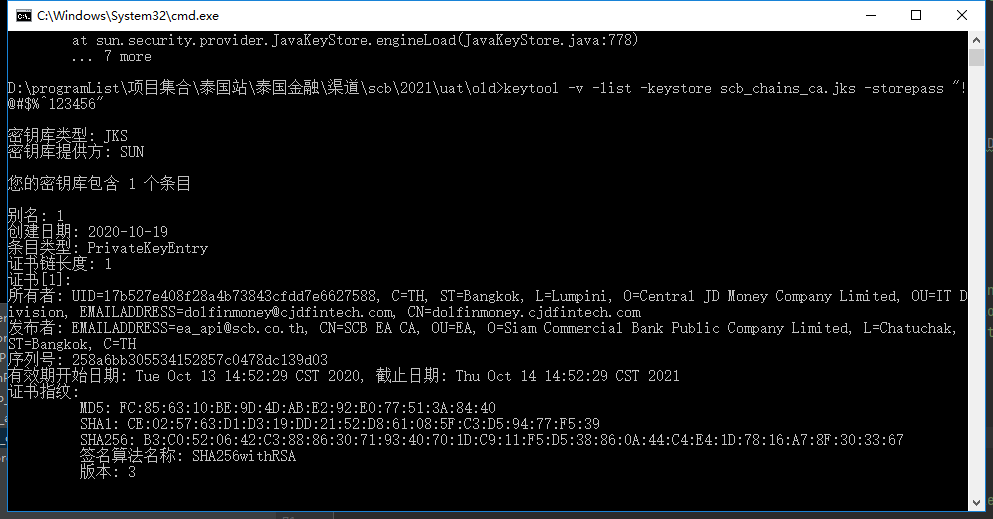
cer转jks

keytool -importcert -file dm.cer -keystore test.jks -storepass "!@#$%^123456"



查看jks

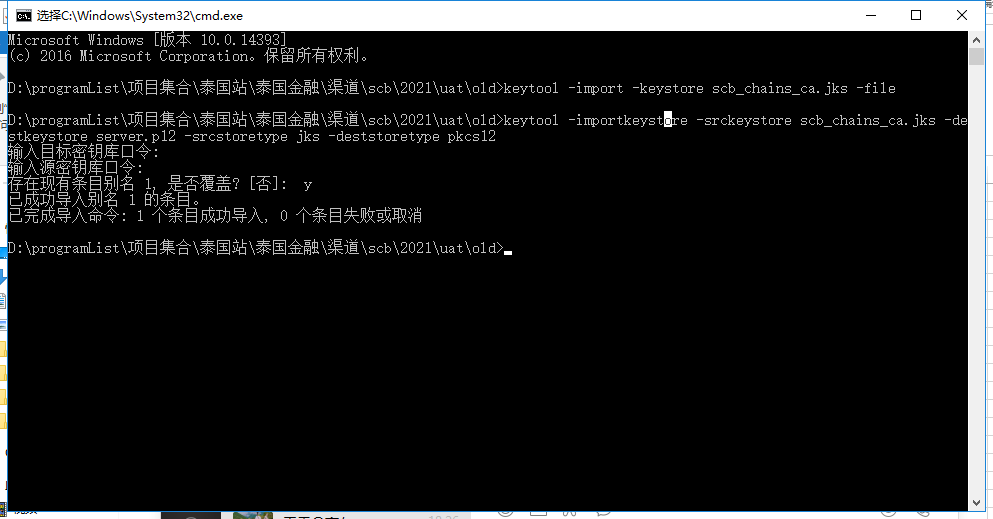
keytool -v -list -keystore uat\_dolfin-money\_SCB.jks -storepass "!@#$%^123456"



jks转p12

keytool -importkeystore -srckeystore scb\_chains\_ca.jks -destkeystore server.p12 -srcstoretype jks -deststoretype pkcs12

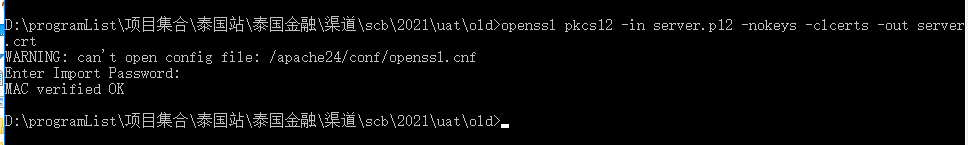
需要输入p12的口令和原jks的口令



p12转crt

openssl pkcs12 -in server.p12 -nokeys -clcerts -out server.crt

需要输入p12的口令



crt转cer

直接window双击crt 在点击复制到文件按钮

p12转key

openssl pkcs12 -in server.p12 -nocerts -nodes -out server.key

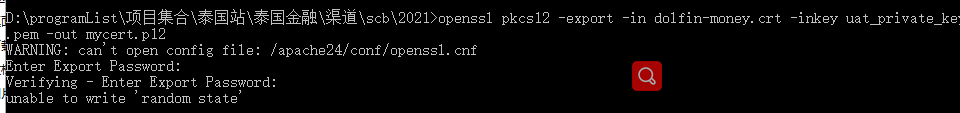
需要输入p12的口令



crt+key/pem 转p12

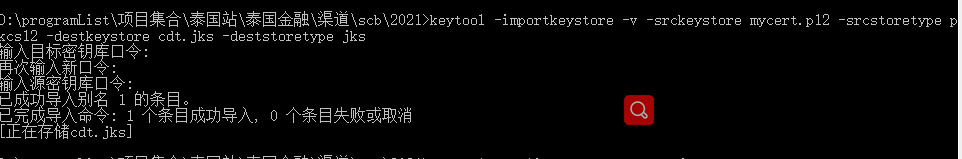
openssl pkcs12 -export -in dolfin-money.crt -inkey uat\_private\_key.pem -out mycert.p12

需要输入新p12文件的密码



p12转jks

keytool -importkeystore -v -srckeystore mycert.p12 -srcstoretype pkcs12 -destkeystore cdt.jks -deststoretype jks



crt+key转jks 需要先转p12在转jsk 其中转p12的密码和转jks都要用一个 否则在使用的时候回报java.security.UnrecoverableKeyException: Cannot recover key异常。

IDEA注册码

http://idea.iteblog.com/key.php

http://intellij.mandroid.cn/

http://idea.imsxm.com/

常用插件

Database Navigator

Code With Me

idea-mybaits-generator

lombok

one dark theme

redis simple

smart tomcat

zookeeper-

Alibaba Java Coding Guidelines

远程debug命令

java -Xdebug -Xrunjdwp:transport=dt\_socket,server=y,suspend=n,address=35005 --Djava.ext.dirs /home/wy/www/aqv.ordering.pickup/lib/\*.jar com.jdd.aqv.PickupApplication

jci

export JAVA\_DEBUG\_OPTS=-Xdebug -Xrunjdwp:transport=dt\_socket,server=y,suspend=n,address=35005

然后启动start.sh脚本即可

swagger生成接口文档

打开swagger2-generate-doc项目

执行 generateAsciiDocsToFile 单元测试

执行mvn generate-resources

curl https 双向证书 post请求命令demo

curl -v -k --request POST --header "Content-Type: application/json" --cert /home/wy/java/jdk1.8.0\_66/jre/lib/security/certificate.pem --key /home/wy/java/jdk1.8.0\_66/jre/lib/security/sit\_private\_key.pem https://api-partners-sit.scb.co.th/partners/v2/payment/creditCards/installment/paymentPlan