ICE 智能资源配置引擎

系统安装部署手册（单机）

INSTALLATION GUIDE

目录

1. 目的

本文档详细描述了 ICE 智能资源配置引擎单机版本的安装步骤，以帮助用户全面了解和管理 ICE 系统的安装部署过程。

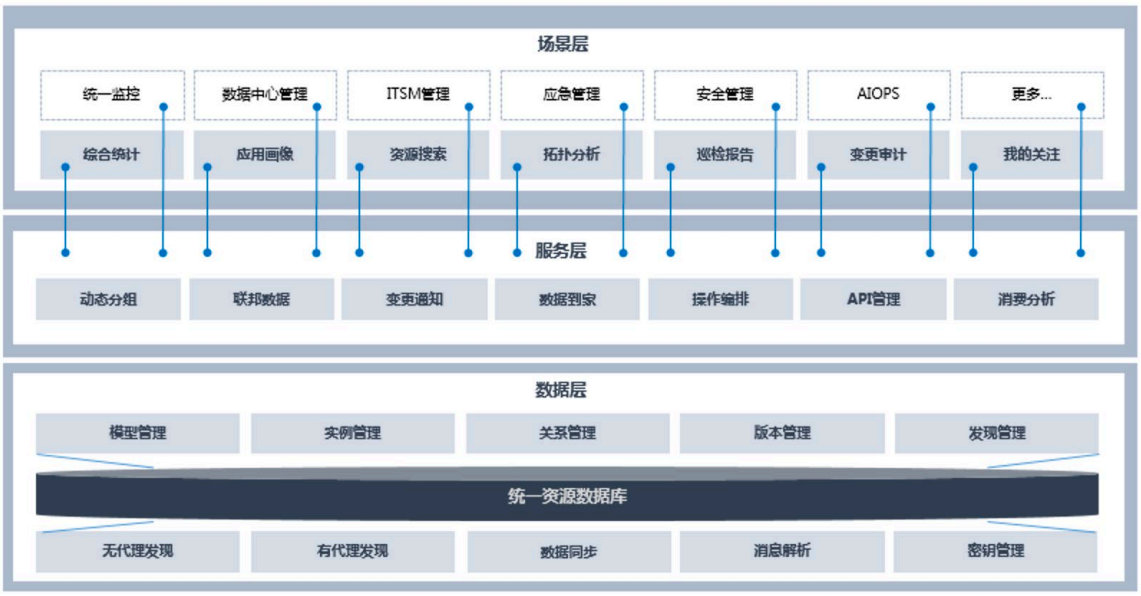
2. 使用范围

- 系统最终用户
- 系统管理用户

3. 系统概览

3.1. 总体架构

ICE智能资源配置引擎 （以下简称“ICE”）的总体架构如下：



ICE智能资源配置引擎总体架构

ICE智能资源配置引擎的整体架构自下而上分为三层：数据层、服务层和场景层。从数据生命周期角度看，数据层、服务层和场景层分别对应了数据生产、数据组装和数据消费的三个主要环节。其中：

数据层负责资源配置数据的自动发现、存储和管理。自动发现包括有代理发现、无代理发现、数据同步、消息解析、访问密钥管理等子功能；数据存储包括一套核心CMDB配置管理数据库；数据管理包括模型管理、实例管理、关系管理、版本管理和发现任务管理等子功能。

服务层负责将生产数据组装成基础服务，提供给不同类型的消费场景，是数据层到消费层的中间桥梁。服务层包括的基础服务有：动态分组、联邦数据、变更通知、数据到家、数据到家、操作编排、API管理和消费分析等。

消费层负责对资源配置的最终消费。消费层包含了一系列消费场景，并可基于CMDB服务层持续扩展和进化出新的消费场景。消费层中的消费场景从消费者角度，可分为CMDB自身场景和第三方场景两大类。

3.2. 组件版本

组件名称	版本号
Flink	1.9.0
Pulsar	2.7.0
Postgresql	13.3
Pgpool-II	4.2.2
Tengine	2.3.2
Nginx	1.17.3
Redis	6.0.10

4. 单节点部署

4.1. 环境就绪

4.1.1. 操作系统

- REDHAT LINUX 7.x or above
- CENTOS 7.x or above

4.1.2. 主机配置

- CPU：4 核 or above
- 内存：16G or above
- 磁盘：200 G or above
- 网卡：千兆

4.1.3. 兼容浏览器

- Google Chrome 81 or above
- Firefox 6 or above
- Microsoft Edge 42 or above

4.2. 程序包就绪

- 主软件包：ice-release-package -[时间戳.].tar.gz

4.3. 数据包就绪

115.159.207.42 中包含了 2 个 demo 数据库文件包，供发布时按需使用：

cd /root/scm/ice-release/install/package

psql-20231127-151946-v2.6.8-demo.tar.gz

mongodb-20231127-152845-v2.6.8-demo.tar.gz

4.4. 单节点部署步骤

4.4.1. 添加 adminop 用户

在安装节点创建 adminop 用户，授予 sudo 权限：

groupadd adminop

useradd -g adminop adminop

passwd adminop << eof

adminop

adminop

eof

usermod -s /bin/bash adminop

usermod -a -G wheel adminop

usermod -a -G adm adminop

sudo visudo

打开 sudo 配置文件， 在文件最后，手工添加:
adminop ALL=(ALL) NOPASSWD: ALL

如果各集群节点挂载磁盘，请设置系统在启动时自动挂载该磁盘：

lsblk 识别磁盘名称

mkfs.ext4 /dev/sdd 格式化磁盘（如果该磁盘尚未格式化。该命令使用 ext4 文件系统）

mkdir /icesystem 创建挂载点目录

mount /dev/sdb /mnt/sdb 挂载磁盘

df -Th 检查挂载情况

echo '/dev/sdd /icesystem ext4 defaults 0 0' | sudo tee -a /etc/fstab 挂载磁盘 automount

（系统启动时自动挂载该磁盘）

4.4.2. 上传安装包

预先按 4.1 准备一台全新的主机实例；主机最小配置建议如下：

OS：Linux 7.x or above

CPU：4 核

MEM：16G

Disk：200G

使用 adminop 用户或其他任意具备系统软件安装权限的授权用户上传 ICE 系统主软件包至指定文件安装目录（下文以 adminop 为例）；

mv ice-release-package-[时间戳].tar.gz /home/adminop

*cd /home/adminop && sudo chown adminop:adminop *.**

tar xvfz ice-release-package-[时间戳]..tar.gz

*sudo chown -R adminop:adminop **

4.4.3. 确认主机防火墙端口已放开

提前确认本地防火墙已放开以下端口：27017,48080,5432,63799,6650,8601,8602,8630,8081

4.4.4. 执行安装脚本

以安装目录为 adminop 为例（已在目录 adminop 下）：

su – root (**NOTE：需要切换到 root 用户**)

cd install/

vim ice-deploy-config.conf

替换 *install_source_node={$ssh_ip:$ssh_port $ssh_user:$ssh_pwd}*中变量为本机实际信息：

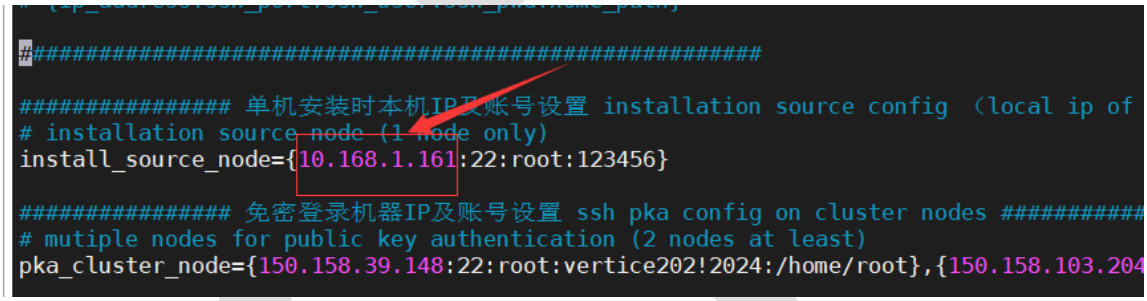
$ssh_ip：SSH 登录本机使用的 ip 地址

$ssh_port：SSH 登录本机使用的端口

$ssh_user：SSH 登录本机使用的用户名

$ssh_pwd：SSH 登录本机使用的用户密码

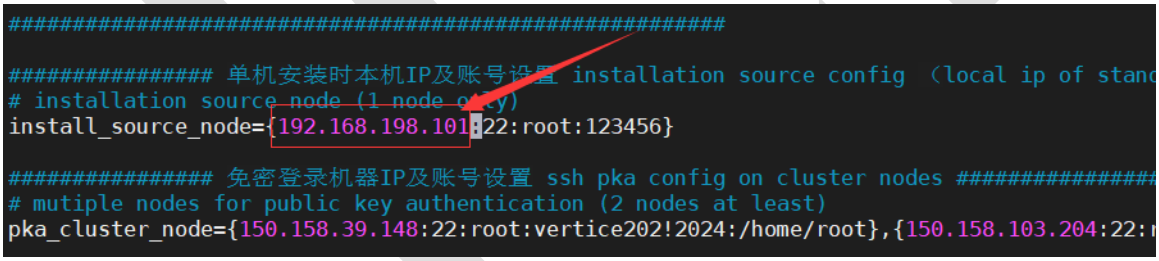
以下为示例，供参考：



将上图中红框内的 ip 地址修改为所装主机的 ip 地址（以主机 192.168.198.101 为例）：

按 i 命令进入 vim 的编辑模式

修改结果如下：



按 ESC 退出编辑模式，然后按英文状态下的冒号，wq 保存退出即可

开始安装：

cd install/

./ice-install-manager-nonroot.sh 01

主机环境检查 Check Env Readiness...

是否继续 Are you sure want to continue (y/n)?输入 y，开始安装

请确认本机 IP 地址 Please confirm local host ip address [use 192.168.198.101 by default]:直接按回车键

开始安装系统主程序包 Need to install ice core package（y/n）？输入 y，系统自动安装主软件包

Please select install package：

1） ice-20220331151729.tar.gz

2） mongodb-20220305-210339.tar.gz

3） psql-20220305-210356.tar.gz

输入 1，回车，开始安装 ice-20220331151729.tar.gz

是否安装 Need to install netscan tools （y/n）？输入 y，系统自动安装扫描组件

是否安装 Need to install expect&tcl tools (y/n)？输入 y，系统自动安装基础组件

是否安装 Need to install dos2unix tool (y/n)? 输入 y，系统自动安装 dos2unix 组件

是否安装 Need to install jq tools (y/n)? 输入 y，系统自动安装 jq 组件

是否安装 Need to install perl tools (y/n)? 输入 y，系统自动安装 perl 组件

是否安装 Need to install mysql shell tool（y/n)? 输入 y，系统自动安装 mysql 命令行工具

是否安装 Need to re-install psql-9.4 (y/n)？输入 y，系统开始重新安装数据库组件

是否安装 Need to deploy cmdb data package (y/n)？输入 y，系统开始部署配置库基准文件

Please select cmdb data package:

1） mongodb-20220305-210339.tar.gz

输入 1，回车

是否安装 Need to deploy psql data package (y/n)？输入 y，系统开始部署数据库基准文件

Please select psql data package:

1） psql-20220305-210356.tar.gz

输入 1，回车

是否安装 Need to deploy apm package (y/n)? **输入 n，跳过不安装**

是否安装 Need to deploy apm-cli tool (y/n)? **输入 n，跳过不安装**

确认本机 IP 地址 Before run 'ice-ops-manager.sh', please confirm the ip address of local host[default ip:192.168.198.101]: (说明：确认默认 default ip 地址无误，可直接按回车)回车

是否启动'ice-ops-manager.sh' Would you like to start 'ice-ops-manager.sh' tool (y/n)?输入 y，启动 ice-ops-manager.sh

Please input release IP[use 122.51.197.74\|122.51.184.68\|115.159.34.125 by default]: 回车

```
Update IP configuration...
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Please input release IP[use 122.51.197.74\|122.51.184.68\|115.159.34.125 by default]:
Use '122.51.197.74\|122.51.184.68\|115.159.34.125' as release IP...
eth0: error fetching interface information: Device not found
Please input target IP[use  by default]: 192.168.198.101
Update Shell IP...
```

手工输入主机 ip 地址（脚本默认找 ech0 中的 ip 地址，找不到需要手工输入）

Please input inner IP for subagent[use by default]:

```
Please input inner IP for subagent[use  by default]: 192.168.198.101
HOST_IP=192.168.198.101
# pulsar集群部署时，采用逗号分隔，如：pulsar://192.168.198.101:6650,12
PULSAR_URL=pulsar://192.168.198.101:6650
PG_IP=192.168.198.101
```

手工输入主机 ip 地址（未找到 ip 地址，需要手工输入）

演示机是单网卡，只需要在此处输入一次 ip

自动导入 mongo，如下图示例：

```
2022-04-03T10:24:02.378+0800    reading metadata for ice.searchLog from ice/searchLog.meta
2022-04-03T10:24:02.378+0800    reading metadata for ice.instance from ice/instance.metada
2022-04-03T10:24:02.378+0800    restoring ice.modelHistory from ice/modelHistory.bson
2022-04-03T10:24:02.387+0800    restoring ice.model from ice/model.bson
2022-04-03T10:24:02.396+0800    restoring ice.searchLog from ice/searchLog.bson
2022-04-03T10:24:02.457+0800    restoring ice.instance from ice/instance.bson
2022-04-03T10:24:03.364+0800    restoring indexes for collection ice.instance from metadat
2022-04-03T10:24:03.801+0800    restoring indexes for collection ice.modelHistory from met
2022-04-03T10:24:03.824+0800    restoring indexes for collection ice.model from metadata
2022-04-03T10:24:03.865+0800    finished restoring ice.model (198 documents, 0 failures)
2022-04-03T10:24:03.865+0800    reading metadata for ice.instanceHistory from ice/instance
2022-04-03T10:24:03.889+0800    restoring ice.instanceHistory from ice/instanceHistory.bso
2022-04-03T10:24:03.939+0800    finished restoring ice.modelHistory (3700 documents, 0 fai
2022-04-03T10:24:03.939+0800    reading metadata for ice.virtualNodeCache from ice/virtual
2022-04-03T10:24:03.987+0800    restoring ice.virtualNodeCache from ice/virtualNodeCache.b
2022-04-03T10:24:04.047+0800    restoring indexes for collection ice.instanceHistory from
2022-04-03T10:24:04.204+0800    finished restoring ice.instanceHistory (6922 documents, 0
```

自动导入 pg 库数据，如下图所示

```
COPY 102
COPY 0
COPY 215
 setval
--------
    404
(1 row)

COPY 0
 setval
--------
     84
(1 row)

COPY 36217
 setval
```

启动完成界面如下图所示：

```
Start IceDisDataCenter Successful
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
try to start OpenApi...
Waiting for localhost to listen on 6650...
Waiting for localhost to listen on 8080...
Waiting for localhost to listen on 5432...
Starting IceOpenApi.........
Start IceOpenApi Successful
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
try to start SA...
Waiting for localhost to listen on 6650...
Waiting for localhost to listen on 8080...
Waiting for localhost to listen on 5432...
Starting IceSuperAgent.........
Start IceSuperAgent Successful
Connected to web console successfully, you can use ICE NOW!
-----------------------------------
start finished.


Time Duration: 167 s
[root@ceshiji ice]#
```

安装并启动完成。

4.4.5. 云服务器补充设置（云服务器内外网地址不同时使用）

云服务器中输入 ifconfig 查看内网地址，如 10.0.4.2

分别修改以下 2 个文件中的 HOST_IP 参数

$ICE_HOME/app/sa/bin/run.sh

$ICE_HOME/app/subagent/bin/ice-subagent-start.sh

修改示例：HOST_IP=10.0.4.2

4.4.6. 执行启动脚本（手工）

系统将在安装后自行完成启动。如需手工启动，请执行以下操作：

cd ../ice

./ice-ops-manager.sh 01

或者

./ice-ops-manager.sh 回车

输入 01：重启 ICE 系统

4.4.7. 导入 License

cd /home/adminop/ice/app/script

./license_update.sh $license

NOTE: $license 变量用实际 license 字符串替换，license 字符串请向系统管理员申请

4.4.8. 访问系统

打开浏览器，输入以下 URL：

http://[ip]:48080/ice

系统登录账号请与项目交付团队联系，交付人员负责提供初始化账号和密码

即刻开始您的 ICE 旅程。

4.4.9. 检查端口是否开放

cd $ICE_HOME/app/selfmonitor/

vi ./ice-selfmonitor.conf

修改以下参数；

host_ip=111.229.75.202 (替换成本机 ip 地址)

target_services=dc,flink,mainserver,openapi,pulsar,redis,sa,mongo,subagent,pgsql

openapi_service_cluster_ip=10.0.4.2 (替换成本机 ip 地址；注意：云服务器此处要用内网地址，如 10.0.4.2)

mail_service_cluster_ip=111.229.75.202 (替换成本机 ip 地址)

pgsql_service_cluster_ip=111.229.75.202 (替换成本机 ip 地址)

selfmonitor_mail_to=wangl@qq.com (替换成管理员邮箱地址)

deploy_mode=standalone

alive_mode=keepalive **（NOTE：默认 unknown，设置成 keepalive 后，系统发现服务状态异常后将自动重启服务）**

退出 vi
同目录下执行
./ice-selfmonitor-main.sh standalone

程序运行并自动查看以下端口状态，0 为服务拨测成功，代表服务运行正常；1 为服务拨测失败，代表服务运行异常

Step 4: Check core service port status...
---
mainserver_port_status:0
nginx_port_status:
nginx_h5_web_port_status:
datacenter_port_status:0
pulsar_main_port_status:0
pulsar_discover_port_status:0
superagent_port_status:1 （服务异常）
subagent_port_status:0
subagent_ipscan_port_status:1 （服务异常）
flink_masters_port_status:1 （服务异常）
pg_port_status:
redis_port_status_s:0
redis_port_status_n1: （集群模式时使用，单机模式忽略）
redis_port_status_n2: （集群模式时使用，单机模式忽略）
openapi_port_status:0

运行以下命令将自动启动所有拨测失败的服务
./ice-selfmonitor-main.sh standalone keepalive

如需启动持续监测系统的自检程序，进入$ICE_HOME 下，运行
./ice-ops-manager.sh 18

除上面的自检拨测程序外，也可参考下表手工拨测系统内服务组件的运行状态

一般需要检查是否开放的端口包括：27017,48080,5432,63799,6650,8601,8602,8630,8081

服务	拨测命令
Mainserver	*timeout 1s curl -vv telnet://[host_ip]:48080*
Mongo	*timeout 1s curl -vv telnet://[host_ip]:27017*
Postgresql	*timeout 1s curl -vv telnet://[host_ip]:5432*
Redis	*timeout 1s curl -vv telnet://[host_ip]:63799*
Pulsar	*timeout 1s curl -vv telnet://[host_ip]:6650*
DC	*timeout 1s curl -vv telnet://[host_ip]:8601*
Openapi	*timeout 1s curl -vv telnet://[host_ip]:8602*
Subagent	*timeout 1s curl -vv telnet://[host_ip]:8630*
Flink	*timeout 1s curl -vv telnet://[host_ip]:8081*

4.4.10. 参数调优

Mainserver 运行性能调优

如主机配置内存较小，可能需要调整部分服务的启动内存参数。下面以 Mainserver 为例:

$ICE_HOME/app/main/app/bin/mainServer

vi start.sh

*将下面的“***-Xms4096m -Xmx4096m***”调整为“***-Xms2048m -Xmx2048m***”*

export JAVA_OPTS="$JAVA_OPTS -XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=256m ***-Xms4096m -Xmx4096m*** *-Xmn1536m -XX:+UseG1GC -XX:MaxGCPauseMillis=10 -XX:+ParallelRefProcEnabled -XX:+UnlockExperimentalVMOptions -XX:+DoEscapeAnalysis -XX:ParallelGCThreads=32 -XX:ConcGCThreads=32 -XX:G1NewSizePercent=50 -XX:+DisableExplicitGC -XX:-ResizePLAB -XX:+PrintGCDetails -Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=9802 -Dcom.sun.management.jmxremote.authenticate=false -Dcom.sun.management.jmxremote.ssl=false -Dfile.encoding=UTF8 -Djava.awt.headless=true -Djava.security.auth.login.config=$CATALINA_HOME/conf/jaas.config"*

低配置环境下，可执行以下命令一次性将 java 启动参数设置为 1024M

find ./ -name ".sh" -type f -exec sed -i 's|\-Xms4096m \-Xmx4096m|\-Xms1024m \-Xmx1024m|g' {} \;*

Mainserver 启动性能调优

1，按需清理 /root/ice/app/main/app/web/server/下的无用 web 目录，一般可清理的目录包括 webssh，ice_black，wechat，ice_white，topology_white 等

```
mv webssh ./..
mv wechat ./..
mv ice_black ./..
mv ice_white ./..
mv topology_white ./..
```

2，$ICE_HOME/app/main/app/bin/mainServer/conf/catalina.properties 中添加 jarsToSkip 的配置

```
tomcat.util.scan.StandardJarScanFilter.jarsToSkip=\
serializer.jar,\
xercesImpl.jar,\
xml-apis.jar,\
annotations-api.jar,\
ant-junit*.jar,\
```

ant-launcher.jar,
ant.jar,
asm-.jar,*
aspectj.jar,*
bootstrap.jar,
catalina-ant.jar,
catalina-ha.jar,
catalina-storeconfig.jar,
catalina-tribes.jar,
catalina.jar,
cglib-.jar,*
cobertura-.jar,*
commons-beanutils.jar,*
commons-codec.jar,*
commons-collections.jar,*
commons-daemon.jar,
commons-dbcp.jar,*
commons-digester.jar,*
commons-fileupload.jar,*
commons-httpclient.jar,*
commons-io.jar,*
commons-lang.jar,*
commons-logging.jar,*
commons-math.jar,*
commons-pool.jar,*
dom4j-.jar,*
easymock-.jar,*
ecj-.jar,*
el-api.jar,
geronimo-spec-jaxrpc.jar,*
h2.jar,*
hamcrest-.jar,*
hibernate.jar,*
httpclient.jar,*
icu4j-.jar,*
jasper-el.jar,
jasper.jar,
jaspic-api.jar,
jaxb-.jar,*
jaxen-.jar,*
jdom-.jar,*

jetty-.jar,*
jmx-tools.jar,
jmx.jar,
jsp-api.jar,
jstl.jar,
jta.jar,*
junit-.jar,*
junit.jar,
log4j.jar,*
mail.jar,*
objenesis-.jar,*
oraclepki.jar,
oro-.jar,*
servlet-api-.jar,*
servlet-api.jar,
slf4j.jar,*
taglibs-standard-spec-.jar,*
tagsoup-.jar,*
tomcat-api.jar,
tomcat-coyote.jar,
tomcat-dbcp.jar,
tomcat-i18n-.jar,*
tomcat-jdbc.jar,
tomcat-jni.jar,
tomcat-juli-adapters.jar,
tomcat-juli.jar,
tomcat-util-scan.jar,
tomcat-util.jar,
tomcat-websocket.jar,
tools.jar,
websocket-api.jar,
wsdl4j.jar,*
xmlParserAPIs-.jar,*
xmlParserAPIs.jar,
batik.jar *
groovy.jar,*
gson.jar,*
guava.jar,*
poi.jar,*
pdq.jar,
a-poi.jar,

aspectjweaver-.jar,*
bcpkix-jdk15on-.jar,*
bcprov-jdk15on-.jar,*
bsh-.jar,*
caffeine-.jar,*
checker-qual-.jar,*
curvesapi-.jar,*
db2jcc-.jar,*
error_prone_annotations-.jar,*
esapi-.jar,*
expreval-.jar,*
ezmorph-.jar,*
fastjson-.jar,*
hutool-.jar,*
j2objc-annotations-.jar,*
jackson-.jar,*
jersey-.jar,*
jline-.jar,*
json-.jar,*
pinyin4j-.jar,*
protobuf-.jar,*
stringtemplate-.jar,*
ST4-.jar,*
true.jar,*
pulsar-.jar,*
ehcache-.jar,*
ojdbc8-.jar,*
activemq-.jar,*
zstd-.jar,*
xalan-.jar,*
mysql-.jar,*
ant-.jar,*
xmlbeans-.jar,*
javaee-.jar,*
antlr-.jar,*
transmittable-.jar,*
testng-.jar,*
javax-.jar,*
lz4-.jar,*
snmp4j-.jar,*
quartz-.jar,*

picocli-.jar,*
kahadb-.jar,*
qdox-.jar,*
ognl-.jar,*
mapper-.jar,*
antisamy-.jar,*
jasypt-.jar,*
nekohtml-.jar,*
mqtt-.jar,*
hawtdispatch-.jar,*
validation-.jar,*
jcommander-.jar,*
jboss-.jar,*
classmate-.jar,*
activation-.jar,*
java-diff-utils-.jar,*
truelicense-.jar,*
persistence-.jar,*
hawtbuf-.jar,*
trueswing-.jar,*
SparseBitSet-.jar,*
geronimo-.jar,*
jsr305-.jar,*
truexml-.jar,*
opentest4j-.jar,*
animal-sniffer-annotations-.jar,*
jcip-annotations-.jar,*
apiguardian-.jar,*
org.apache.oltu.oauth2.jar,*
cas-.jar,*
mybatis-.jar,*
sqljdbc4-.jar,*
jedis-.jar,*
mongo-.jar,*
xom-.jar