# [Tomcat开启APR运行模式，优化并发性能](http://blog.csdn.net/goldenfish1919/article/details/78859620)

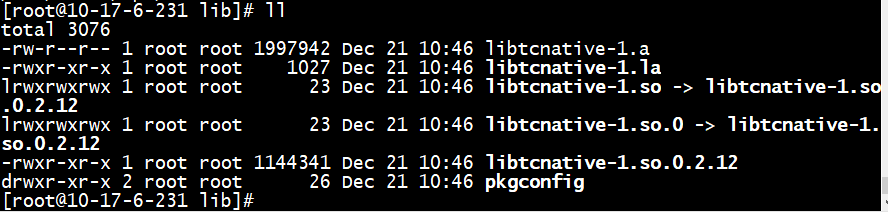
## 安装依赖库

yum install apr-devel

yum install openssl-devel

## 安装apr动态库

进入tomcat的bin目录，解压tomcat-native.tar.gz文件，并进入tomcat-native-1.2.7-src/native目录，执行./configure && make && make install 命令，动态库默认安装在/usr/local/apr/lib目录，如下所示：



## 配置APR本地库到系统共享库搜索路径中

修改catalina.sh:

JAVA\_OPTS="-server -Xms2048M -Xmx2048M -XX:MetaspaceSize=128M -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=$CATALINA\_HOME/logs/heap.dump -Djava.library.path=/usr/local/apr/lib"

【注意】开启了apr之后，jvm用到的native内存会增大，因此要适当调大Metaspace空间,添加JVM选项：-XX:MetaspaceSize=128m

## 关闭SSL

如果不想启用SSL，将server.xml中apr模式下ssl关闭即可：

<Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="off" />

## 修改Server.xml

<Connector port="8080" protocol="org.apache.coyote.http11.Http11AprProtocol"

connectionTimeout="20000" redirectPort="8443" />

## SpringBoot这样来配置

@Bean

public TomcatEmbeddedServletContainerFactory tomcatEmbeddedServletContainerFactory() {

TomcatEmbeddedServletContainerFactory tomcatFactory = new TomcatEmbeddedServletContainerFactory();

tomcatFactory.setProtocol("org.apache.coyote.http11.Http11AprProtocol");

tomcatFactory.addConnectorCustomizers(new TomcatConnectorCustomizer() {

@Override

public void customize(Connector connector) {

Http11AprProtocol handler = (Http11AprProtocol)connector.getProtocolHandler();

handler.setXXX();//对tomcat进行其他的定制

}

});

return tomcatFactory;

}