大数据Hadoop高薪直通车课程

初识 Hadoop 2.x

讲师：轩宇（北风网版权所有)

课程大纲

1 大数据应用发展前景

2 Hadoop 2.x概述

3 Hadoop 2.x生态系统

4 Hadoop 2.x环境搭建

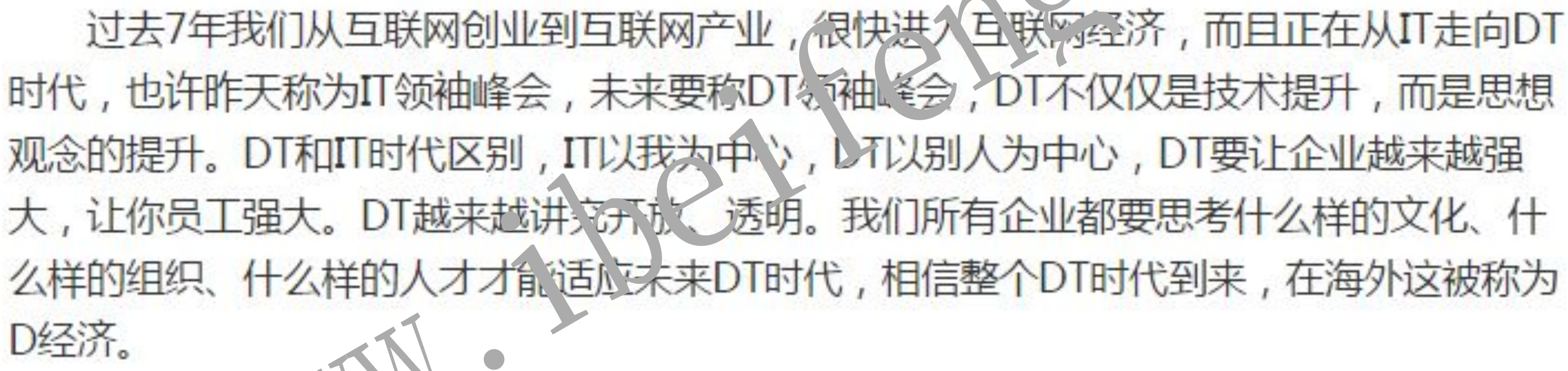
5 初识MapReduce应用

课程大纲

1. 大数据应用发展前景
2. Hadoop 2.x概述
3. Hadoop 2.x生态系统
4. Hadoop 2.x环境搭建
5. 初识MapReduce应用

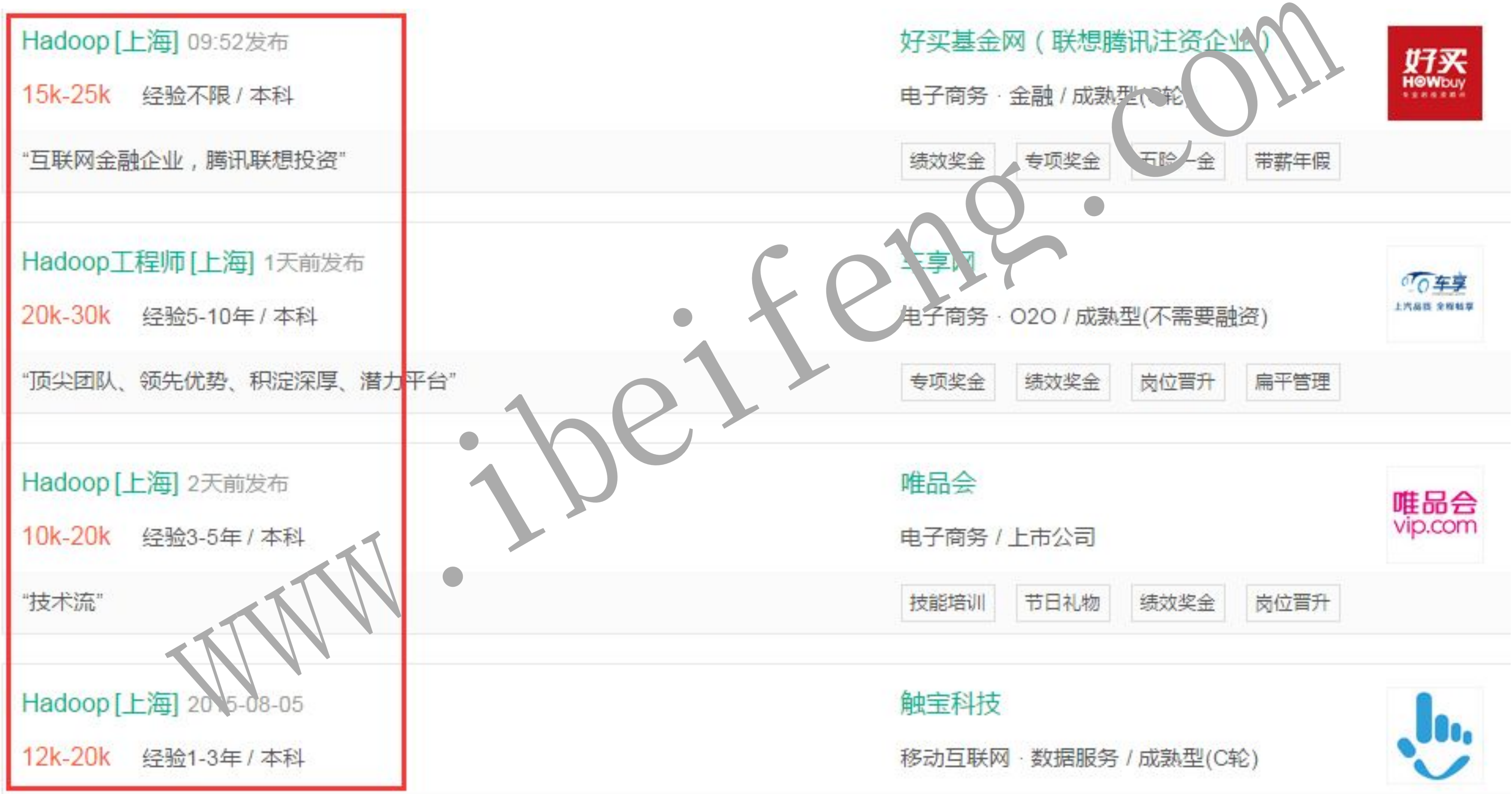
大数据应用发展前景

- 2015年中国（深圳）IT领袖峰会



过去7年我们从互联网创业到互联网产业，很快进入互联网经济，而且正在从IT走向DT时代，也许昨天称为IT领袖峰会，未来要称DT领袖峰会，DT不仅仅是技术提升，而是思想观念的提升。DT和IT时代区别，IT以我为中心，DT以别人为中心，DT要让企业越来越强大，让你员工强大。DT越来越讲究开放、透明。我们所有企业都要思考什么样的文化、什么样的组织、什么样的人才才能适应未来DT时代，相信整个DT时代到来，在海外这被称为D经济。

大数据应用发展前景



Hadoop [上海] 09:52发布

15k-25k 经验不限 / 本科

“互联网金融企业，腾讯联想投资”

好买基金网（联想腾讯注资企业）

电子商务 · 金融 / 成熟型(C轮)

绩效奖金 专项奖金 五险一金 带薪年假

Hadoop工程师 [上海] 1天前发布

20k-30k 经验5-10年 / 本科

“顶尖团队、领先优势、积淀深厚、潜力平台”

车享网

电子商务 · O2O / 成熟型(不需要融资)

专项奖金 绩效奖金 岗位晋升 扁平管理

Hadoop [上海] 2天前发布

10k-20k 经验3-5年 / 本科

“技术流”

唯品会

电子商务 / 上市公司

技能培训 节日礼物 绩效奖金 岗位晋升

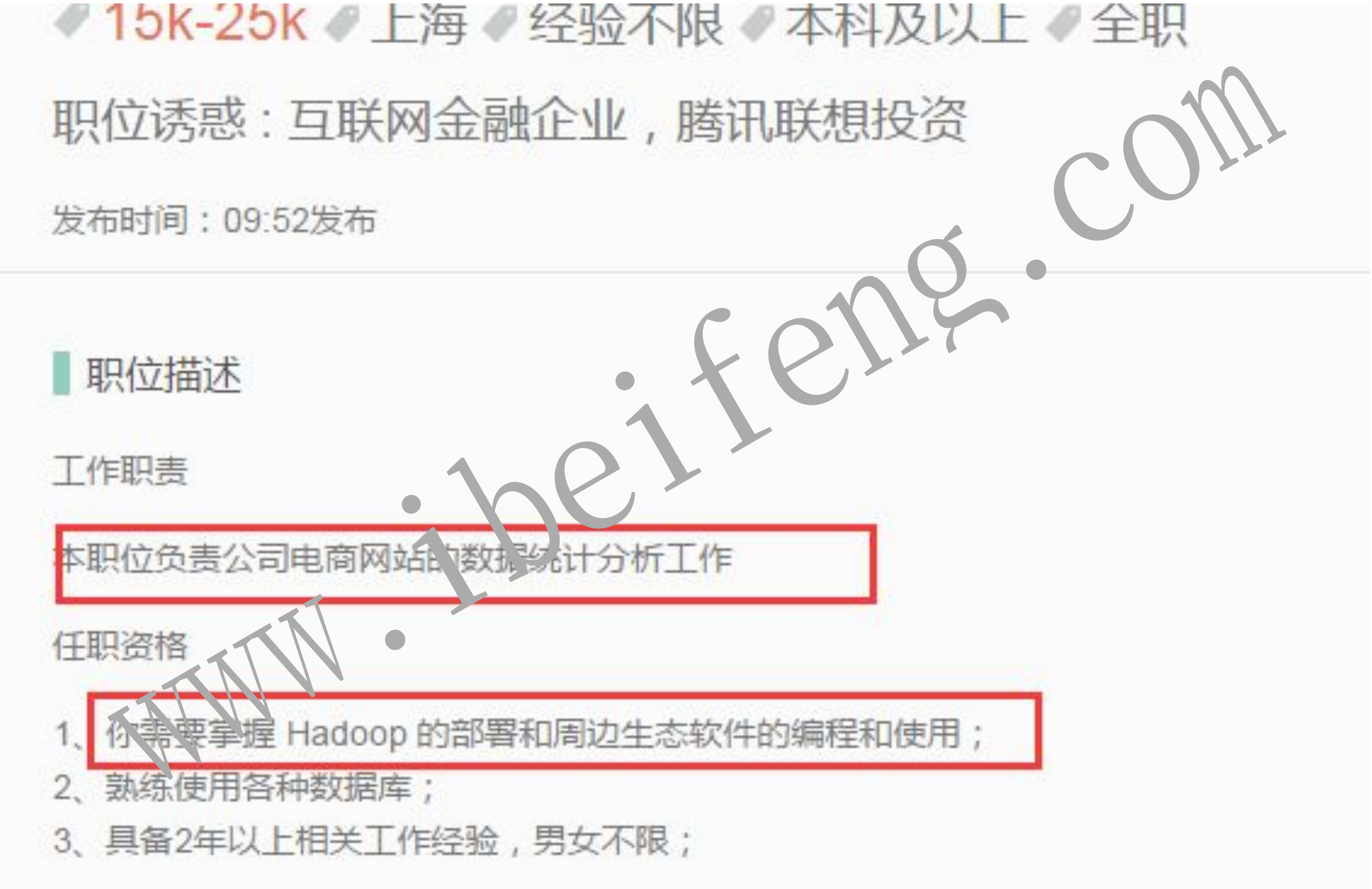
Hadoop [上海] 2015-08-05

12k-20k 经验1-3年 / 本科

触宝科技

移动互联网 · 数据服务 / 成熟型(C轮)

大数据应用发展前景



15k-25k 上海 经验不限 本科及以上 全职

职位诱惑：互联网金融企业，腾讯联想投资

发布时间：09:52发布

职位描述

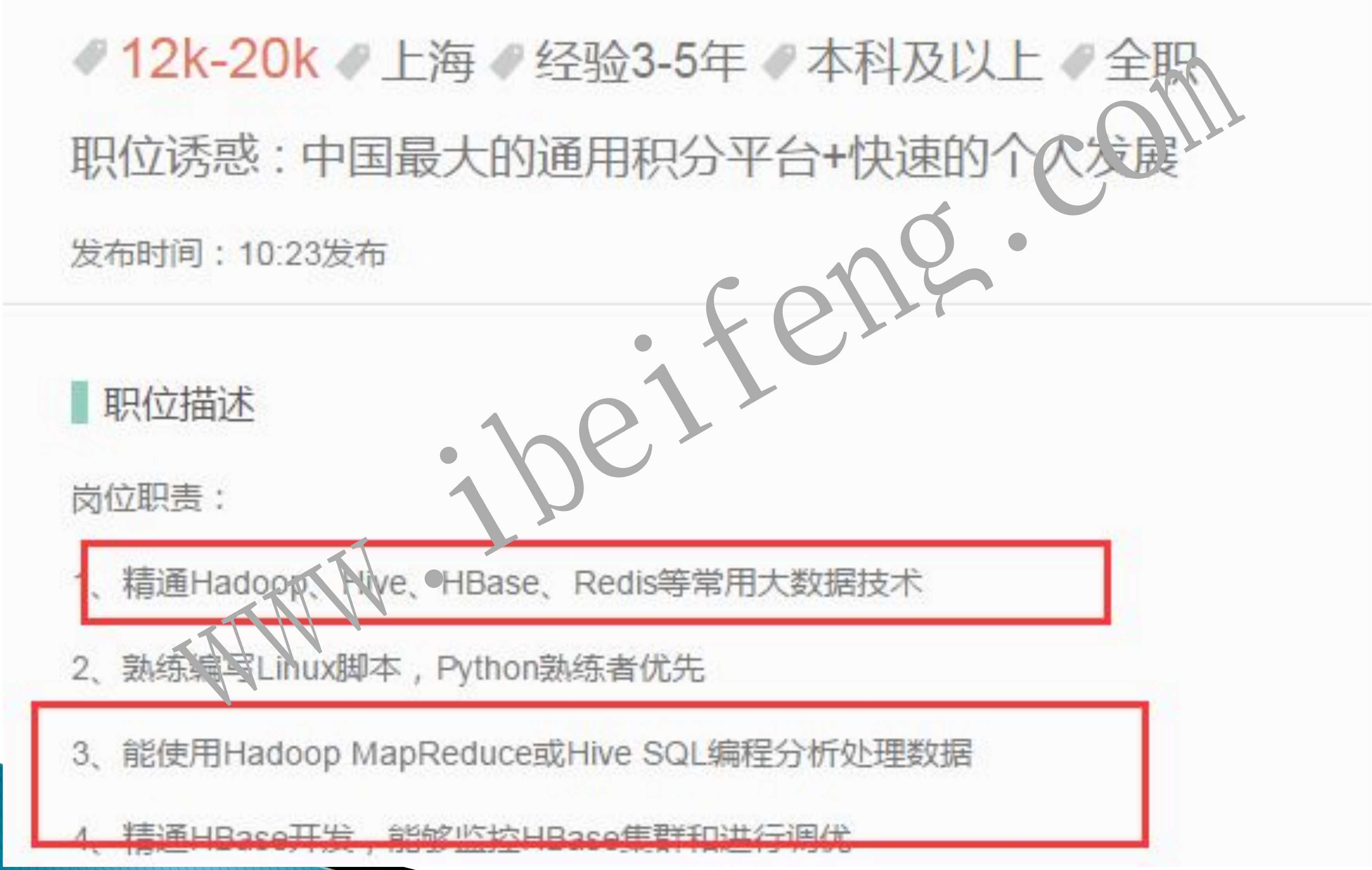
工作职责

本职位负责公司电商网站的数据统计分析工作

任职资格

1、你需要掌握 Hadoop 的部署和周边生态软件的编程和使用；
2、熟练使用各种数据库；
3、具备2年以上相关工作经验，男女不限；

大数据应用发展前景



12k-20k 上海 经验3-5年 本科及以上 全职

职位诱惑：中国最大的通用积分平台+快速的个人发展

发布时间：10:23发布

职位描述

岗位职责：

1、精通Hadoop、Hive、HBase、Redis等常用大数据技术

2、熟练编写Linux脚本，Python熟练者优先

3、能使用Hadoop MapReduce或Hive SQL编程分析处理数据

4、精通HBase开发，能够监控HBase集群和进行调优

课程大纲

1 大数据应用发展前景

2 Hadoop 2.x概述

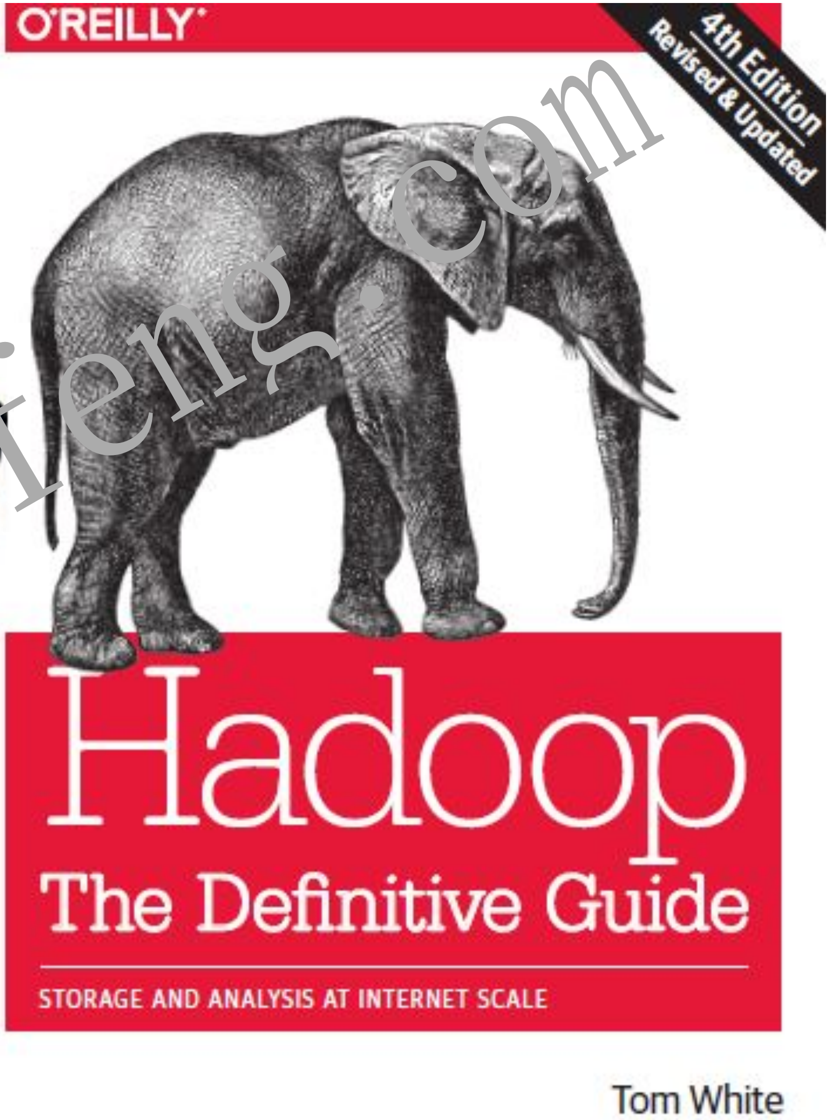
3 Hadoop 2.x生态系统

4 Hadoop 2.x环境搭建

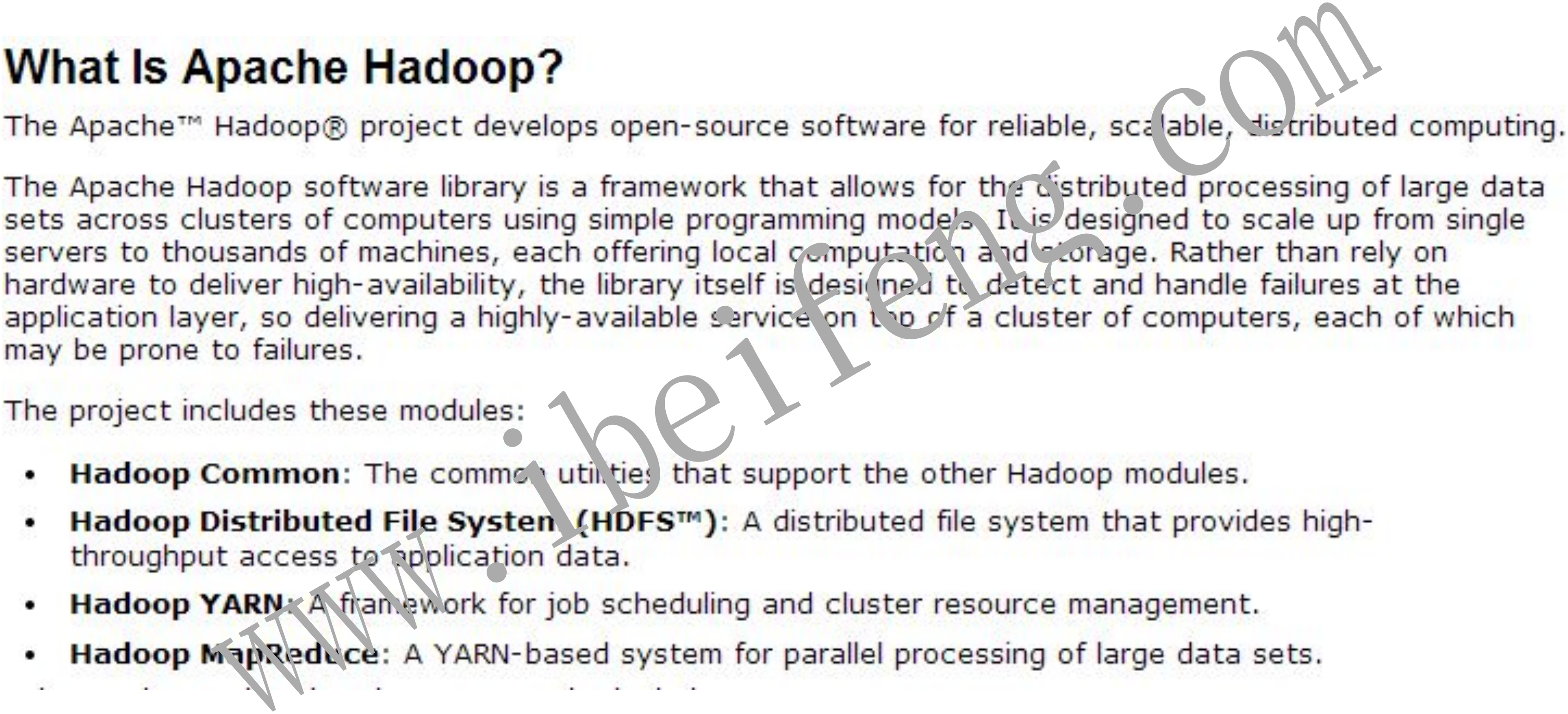
5 初识MapReduce应用

Hadoop 2.x概述

hadoop
Scalable Distributed Computing


O'REILLY
4th Edition
Revised & Updated
Hadoop
The Definitive Guide
STORAGE AND ANALYSIS AT INTERNET SCALE
Tom White

Hadoop 2.x概述



What Is Apache Hadoop?

The Apache™ Hadoop® project develops open-source software for reliable, scalable, distributed computing.

The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.

The project includes these modules:

- **Hadoop Common**: The common utilities that support the other Hadoop modules.
- **Hadoop Distributed File System (HDFS™)**: A distributed file system that provides high-throughput access to application data.
- **Hadoop YARN**: A framework for job scheduling and cluster resource management.
- **Hadoop MapReduce**: A YARN-based system for parallel processing of large data sets.

What Is Apache Hadoop?

- Hadoop项目主要包括以下四个模块
  - **Hadoop Common**:

    为其他Hadoop模块提供基础设施。
  - **Hadoop HDFS**:

    一个高可靠、高吞吐量的分布式文件系统
  - **Hadoop MapReduce**:

    一个分布式的离线并行计算框架
  - **Hadoop YARN**:

    一个新的MapReduce框架，任务调度与资源管理

Apache Hadoop 起源

◆ Apache Lucene

开源的高性能全文检索工具包

◆ Apache Nutch

开源的 Web 搜索引擎

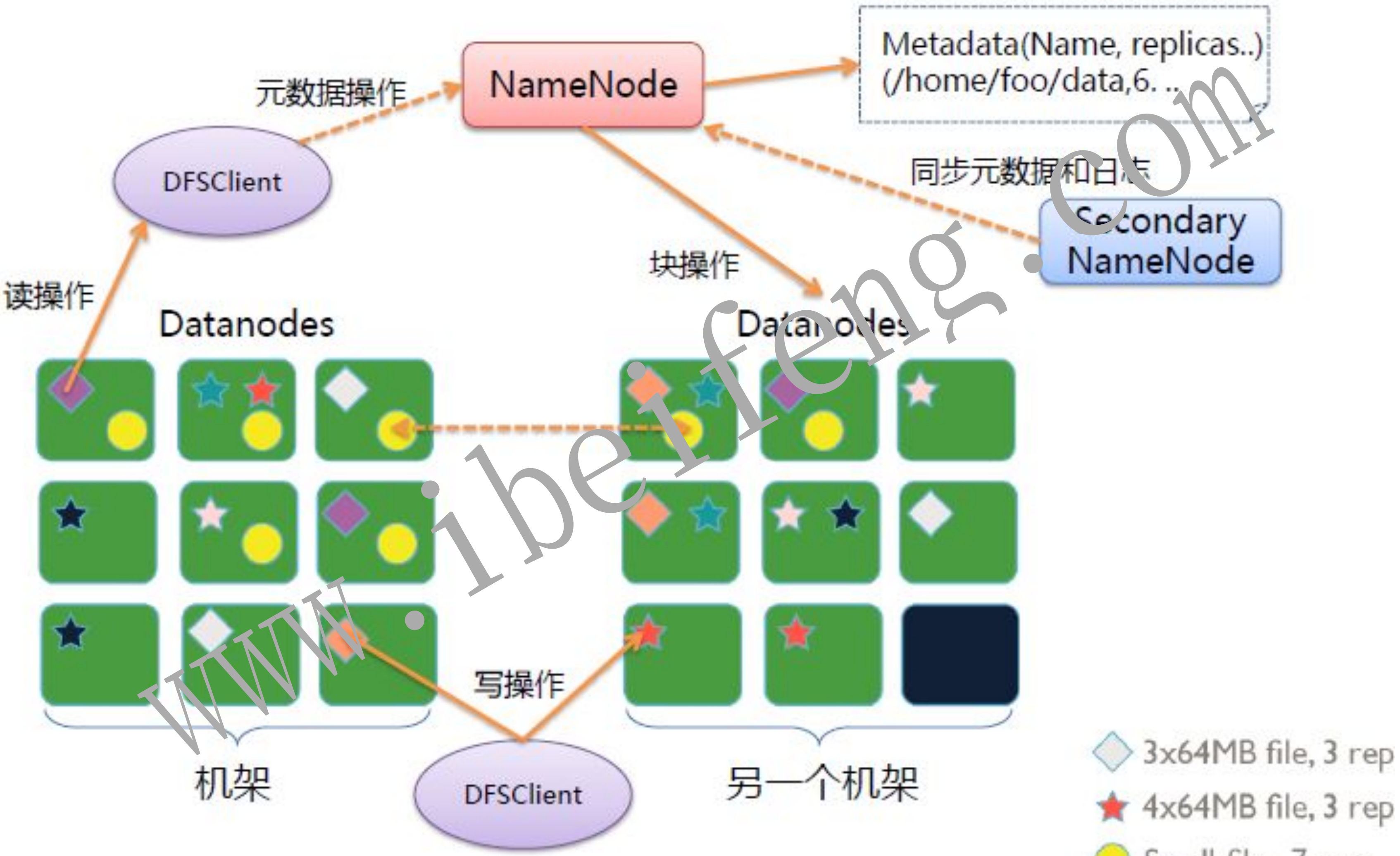
◆ Google 三大论文

MapReduce / GFS / BigTable

◆ Apache Hadoop

大规模数据处理

HDFS 系统架构图

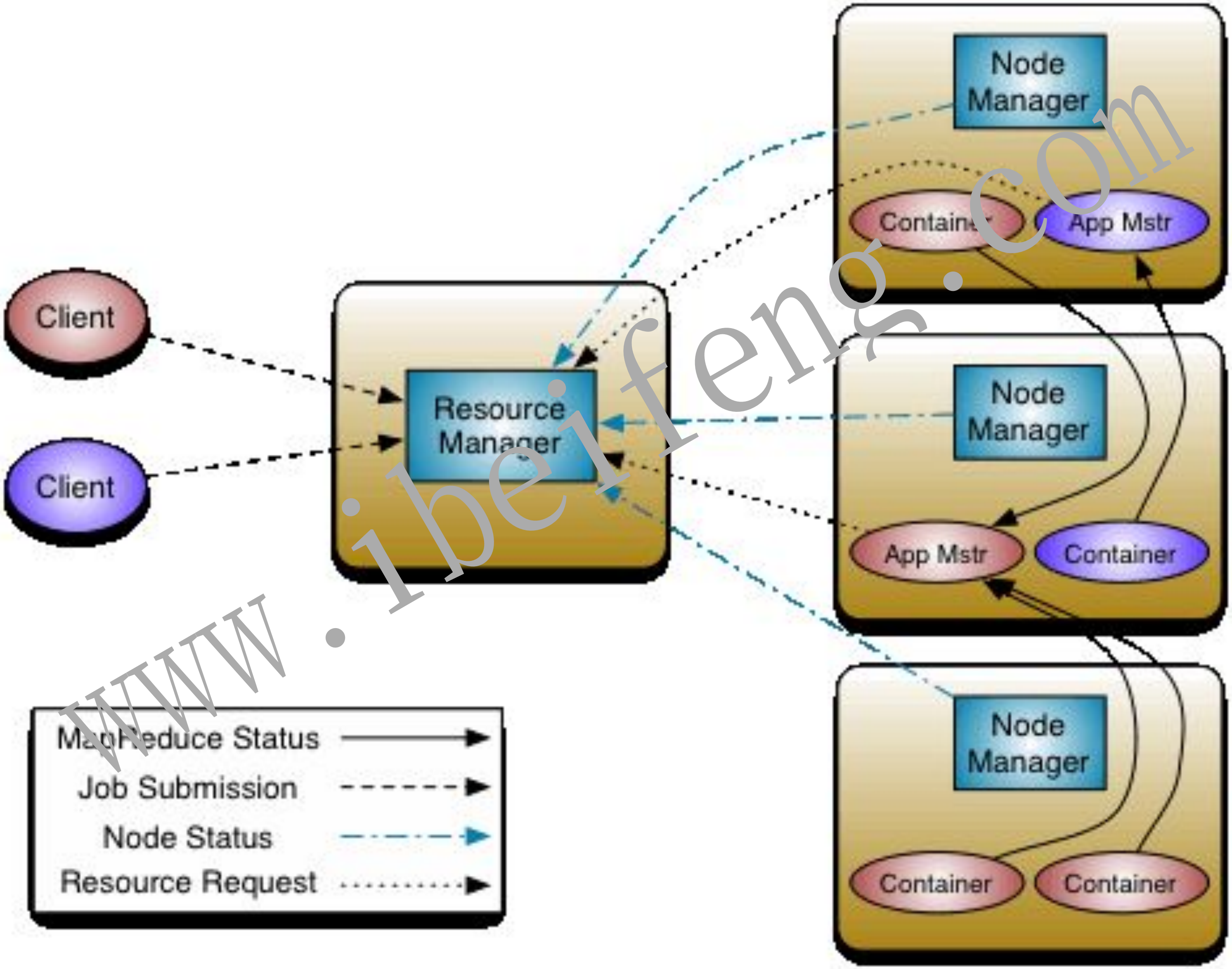


◇ 3x64MB file, 3 rep
★ 4x64MB file, 3 rep
● Small file, 7 rep

HDFS 服务功能

- NameNode是主节点，**存储文件的元数据**如文件名，文件目录结构，文件属性（生成时间,副本数,文件权限），以及每个文件的块列表和块所在DataNode等。
- DataNode 在本地文件系统**存储文件块数据，以及块数据的校验和**。
- Secondary NameNode 用来**监控HDFS状态的辅助后台程序**，每隔一段时间**获取HDFS元数据的快照**。

YARN 架构图


Node Manager
Container
App Mstr
Client
Client
Resource Manager
Node Manager
App Mstr
Container
Node Manager
Container
Container
MapReduce Status
Job Submission
Node Status
Resource Request

YARN 服务功能

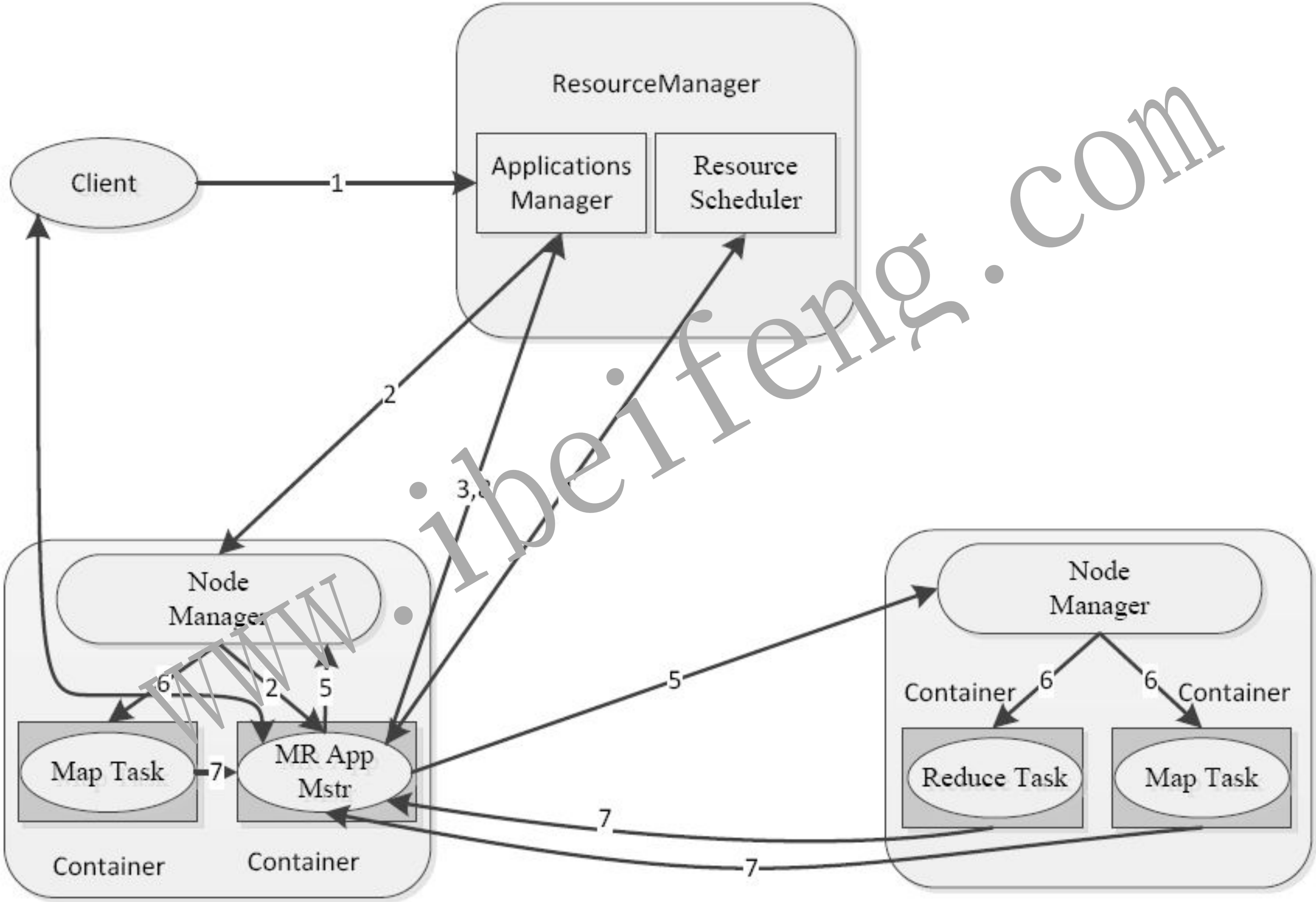
- ResourceManager
  - 处理客户端请求
  - 启动/监控ApplicationMaster
  - 监控NodeManager
  - 资源分配与调度
- NodeManager
  - 单个节点上的资源管理
  - 处理来自ResourceManager的命令
  - 处理来自ApplicationMaster的命令

- ApplicationMaster
  - 数据切分
  - 为应用程序申请资源，并分配给内部任务
  - 任务监控与容错
- Container
  - 对任务运行环境的抽象，封装了CPU、内存等多维资源以及环境变量、启动命令等任务运行相关的信息

离线计算框架 MapReduce

- 将计算过程分为两个阶段，Map和Reduce
  - ✓ Map 阶段并行处理输入数据
  - ✓ Reduce阶段对Map结果进行汇总
- Shuffle连接Map和Reduce两个阶段
  - ✓ Map Task将数据写到本地磁盘
  - ✓ Reduce Task从每个Map Task上读取一份数据
- 仅适合离线批处理
  - ✓ 具有很好的容错性和扩展性
  - ✓ 适合简单的批处理任务
- 缺点明显
  - ✓ 启动开销大、过多使用磁盘导致效率低下等

MapReduce on YARN



课程大纲

1 大数据应用发展前景

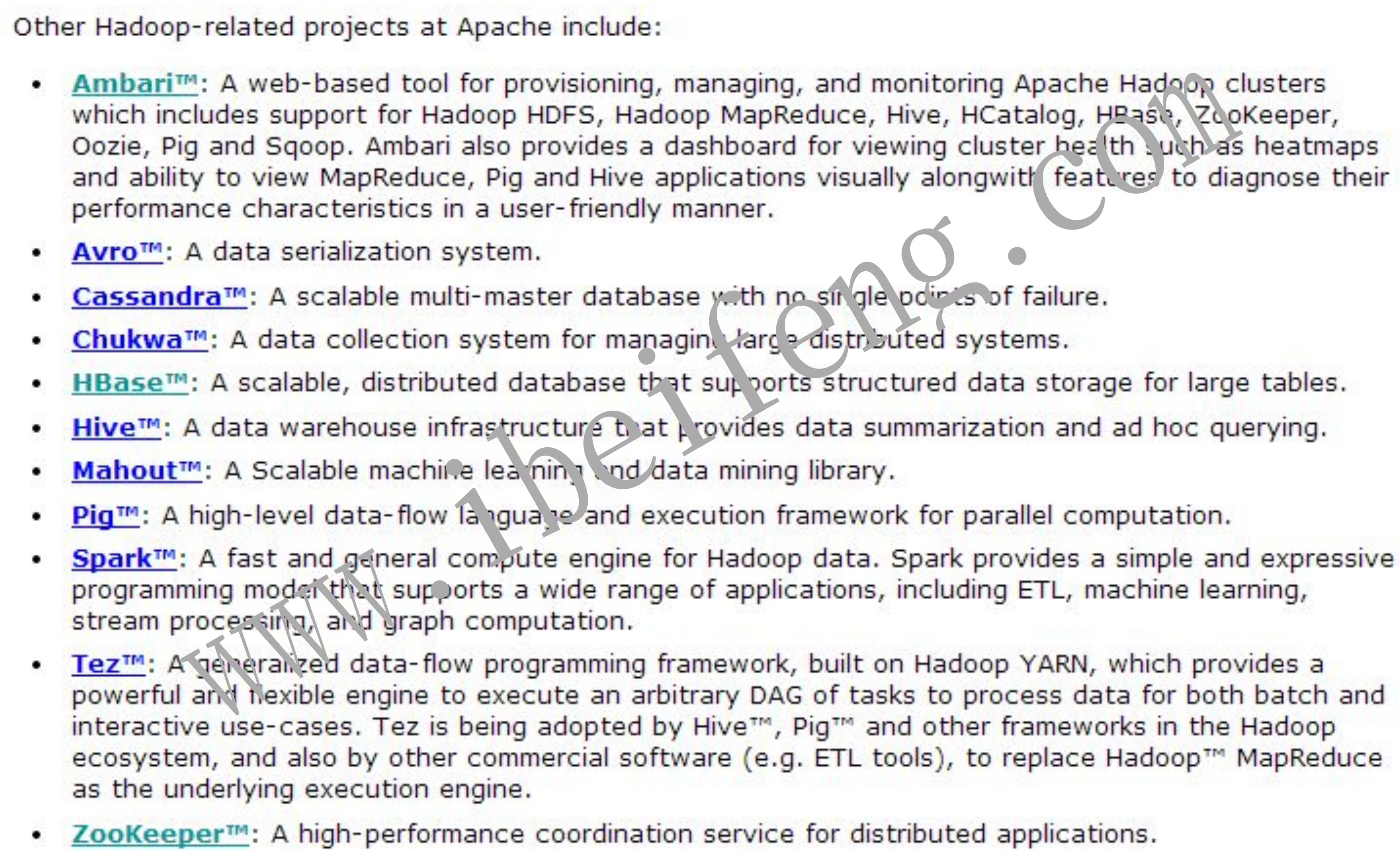
2 Hadoop 2.x概述

3 Hadoop 2.x生态系统

4 Hadoop 2.x环境搭建

5 初识MapReduce应用

Hadoop 2.x 生态系统



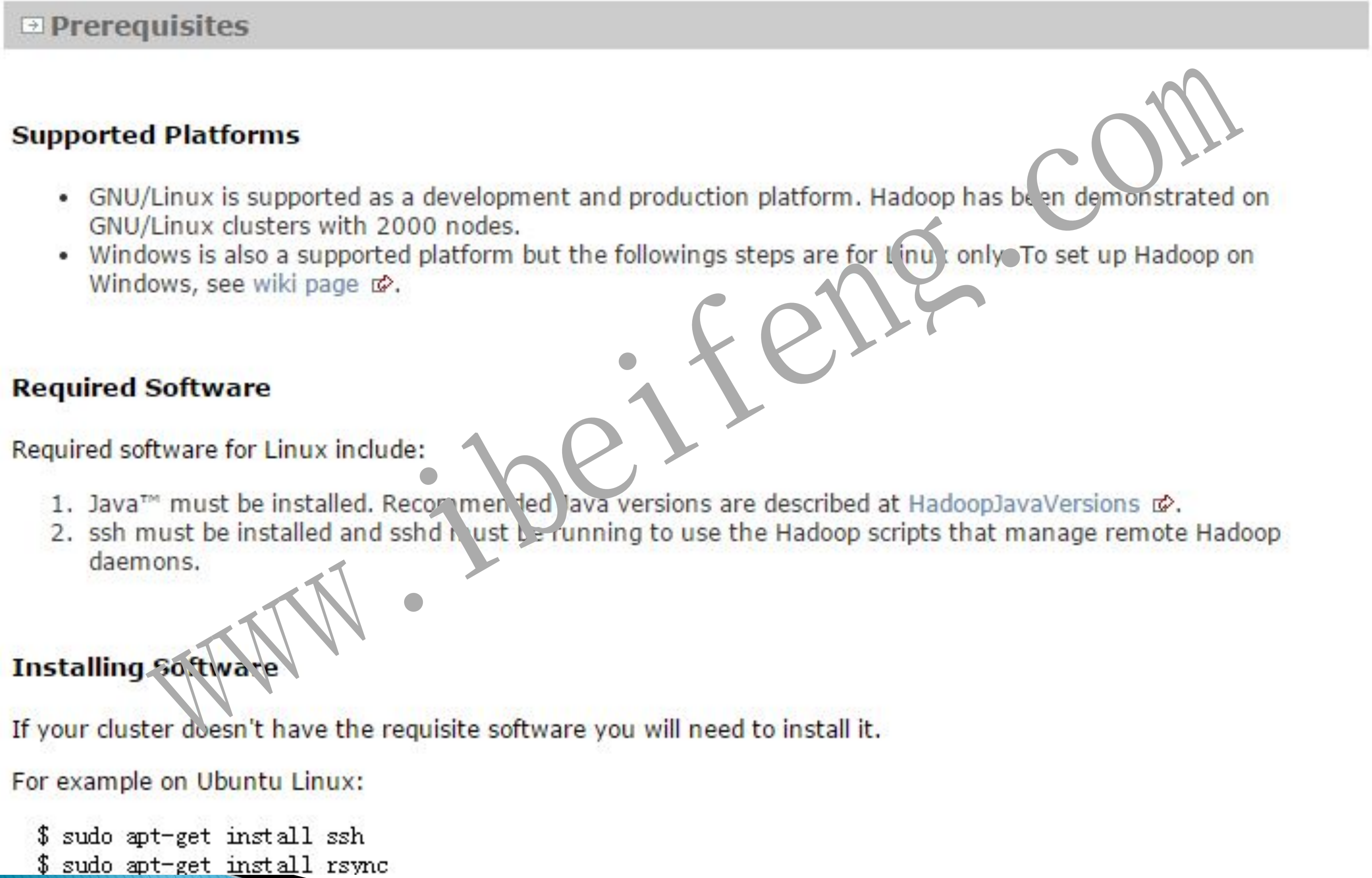
Other Hadoop-related projects at Apache include:

- **Ambari™**: A web-based tool for provisioning, managing, and monitoring Apache Hadoop clusters which includes support for Hadoop HDFS, Hadoop MapReduce, Hive, HCatalog, HBase, ZooKeeper, Oozie, Pig and Sqoop. Ambari also provides a dashboard for viewing cluster health such as heatmaps and ability to view MapReduce, Pig and Hive applications visually alongwith features to diagnose their performance characteristics in a user-friendly manner.
- **Avro™**: A data serialization system.
- **Cassandra™**: A scalable multi-master database with no single points of failure.
- **Chukwa™**: A data collection system for managing large distributed systems.
- **HBase™**: A scalable, distributed database that supports structured data storage for large tables.
- **Hive™**: A data warehouse infrastructure that provides data summarization and ad hoc querying.
- **Mahout™**: A Scalable machine learning and data mining library.
- **Pig™**: A high-level data-flow language and execution framework for parallel computation.
- **Spark™**: A fast and general compute engine for Hadoop data. Spark provides a simple and expressive programming model that supports a wide range of applications, including ETL, machine learning, stream processing, and graph computation.
- **Tez™**: A generalized data-flow programming framework, built on Hadoop YARN, which provides a powerful and flexible engine to execute an arbitrary DAG of tasks to process data for both batch and interactive use-cases. Tez is being adopted by Hive™, Pig™ and other frameworks in the Hadoop ecosystem, and also by other commercial software (e.g. ETL tools), to replace Hadoop™ MapReduce as the underlying execution engine.
- **ZooKeeper™**: A high-performance coordination service for distributed applications.

课程大纲

1. 大数据应用发展前景
2. Hadoop 2.x概述
3. Hadoop 2.x生态系统
4. Hadoop 2.x环境搭建
5. 初识MapReduce应用

Prerequisites



Prerequisites

Supported Platforms

- GNU/Linux is supported as a development and production platform. Hadoop has been demonstrated on GNU/Linux clusters with 2000 nodes.
- Windows is also a supported platform but the followings steps are for Linux only. To set up Hadoop on Windows, see wiki page.

Required Software

Required software for Linux include:

1. Java™ must be installed. Recommended Java versions are described at HadoopJavaVersions.
2. ssh must be installed and sshd must be running to use the Hadoop scripts that manage remote Hadoop daemons.

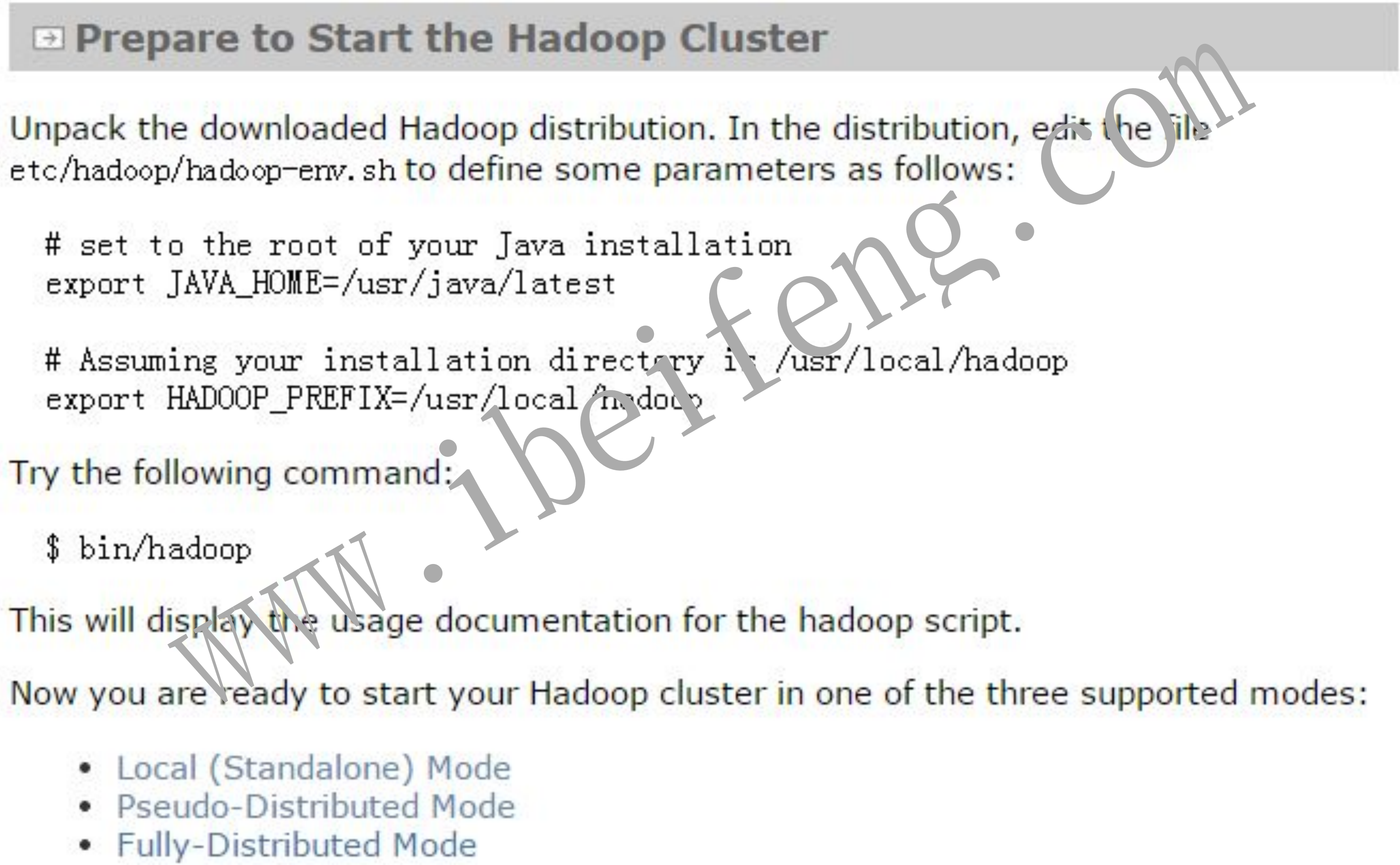
Installing Software

If your cluster doesn't have the requisite software you will need to install it.

For example on Ubuntu Linux:

```
$ sudo apt-get install ssh
$ sudo apt-get install rsync
```

Prepare to Start the Hadoop Cluster



Prepare to Start the Hadoop Cluster

Unpack the downloaded Hadoop distribution. In the distribution, edit the file `etc/hadoop/hadoop-env.sh` to define some parameters as follows:

```
# set to the root of your Java installation
export JAVA_HOME=/usr/java/latest

# Assuming your installation directory is /usr/local/hadoop
export HADOOP_PREFIX=/usr/local/hadoop
```

Try the following command:

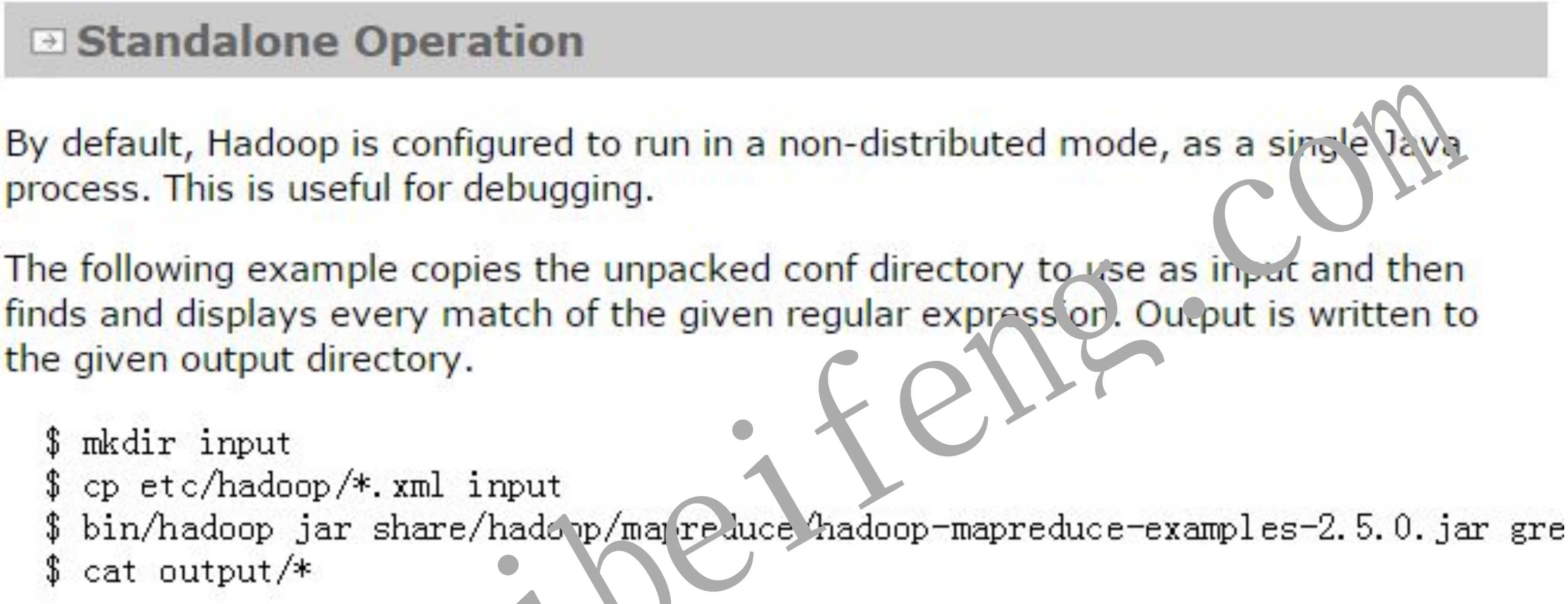
```
$ bin/hadoop
```

This will display the usage documentation for the hadoop script.

Now you are ready to start your Hadoop cluster in one of the three supported modes:

- Local (Standalone) Mode
- Pseudo-Distributed Mode
- Fully-Distributed Mode

Standalone Operation



Standalone Operation

By default, Hadoop is configured to run in a non-distributed mode, as a single Java process. This is useful for debugging.

The following example copies the unpacked conf directory to use as input and then finds and displays every match of the given regular expression. Output is written to the given output directory.

```
$ mkdir input
$ cp etc/hadoop/*.xml input
$ bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.5.0.jar gre
$ cat output/*
```

- $ mkdir input
- $ cp etc/hadoop/*.xml input
- $ bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.5.0.jar grep input output 'dfs[a-z.]+'
- $ cat output/*

Pseudo-Distributed Operation

- 下载

  https://archive.apache.org/dist/hadoop/common/

- 解压

```
$ tar -zxvf hadoop-2.5.0.tar.gz
```

- 修改配置文件（$HADOOP_HOME/etc/hadoop/目录下）

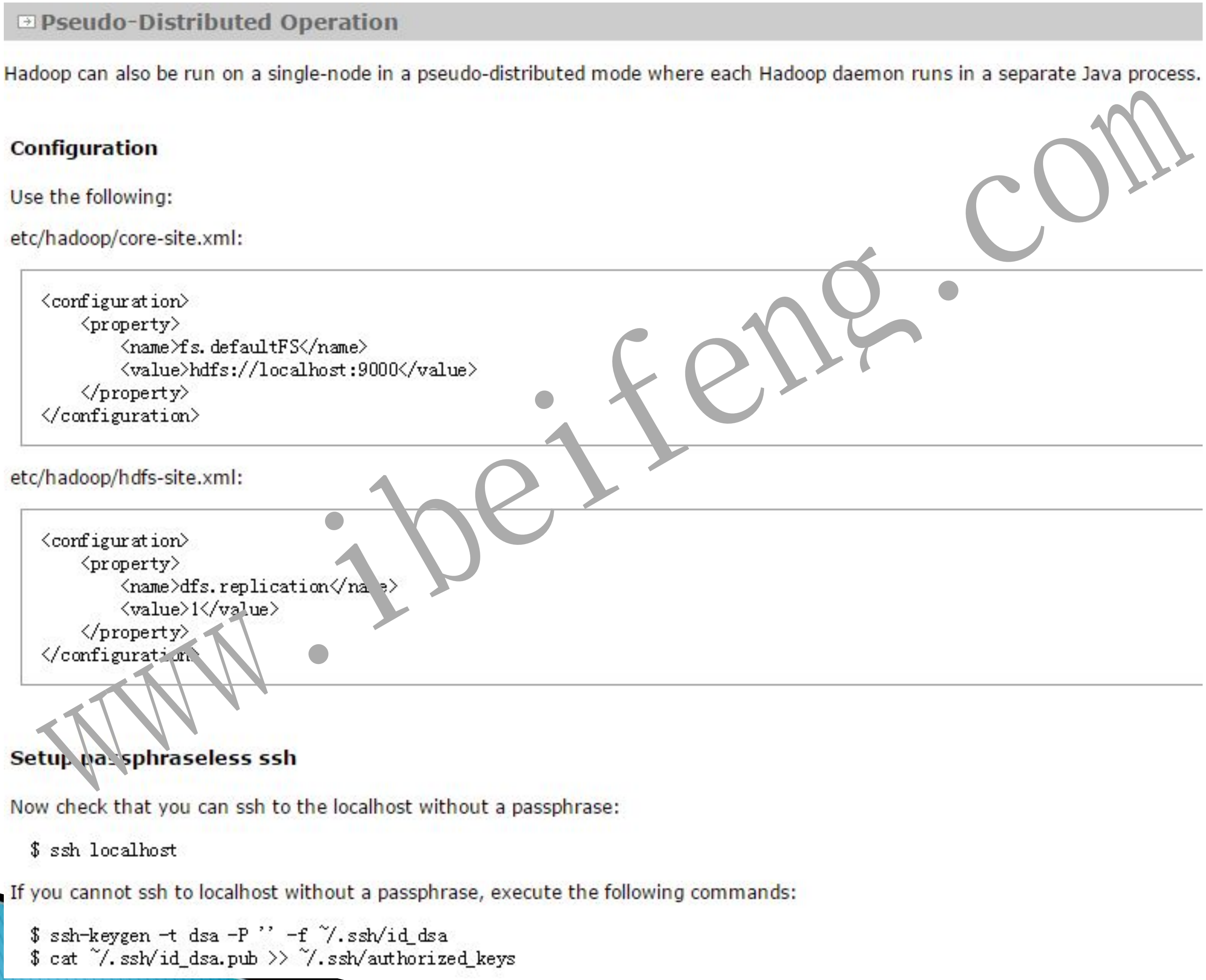
  hadoop-env.sh、core-site.xml、hdfs-site.xml、yarn-site.xml、mapred-site.xml

- 替换本地库

```
$ rm -rf ./$HADOOP_HOME/lib/native/

$ cp -r $HADOOP_SRC_HOME/hadoop-dist/target/hadoop-2.5.0/lib/native/*  $HADOOP_HOME/lib/native/
```

Pseudo-Distributed Operation



Pseudo-Distributed Operation

Hadoop can also be run on a single-node in a pseudo-distributed mode where each Hadoop daemon runs in a separate Java process.

Configuration

Use the following:

etc/hadoop/core-site.xml:

```
<configuration>
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://localhost:9000</value>
    </property>
</configuration>
```

etc/hadoop/hdfs-site.xml:

```
<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
</configuration>
```

Setup passphraseless ssh

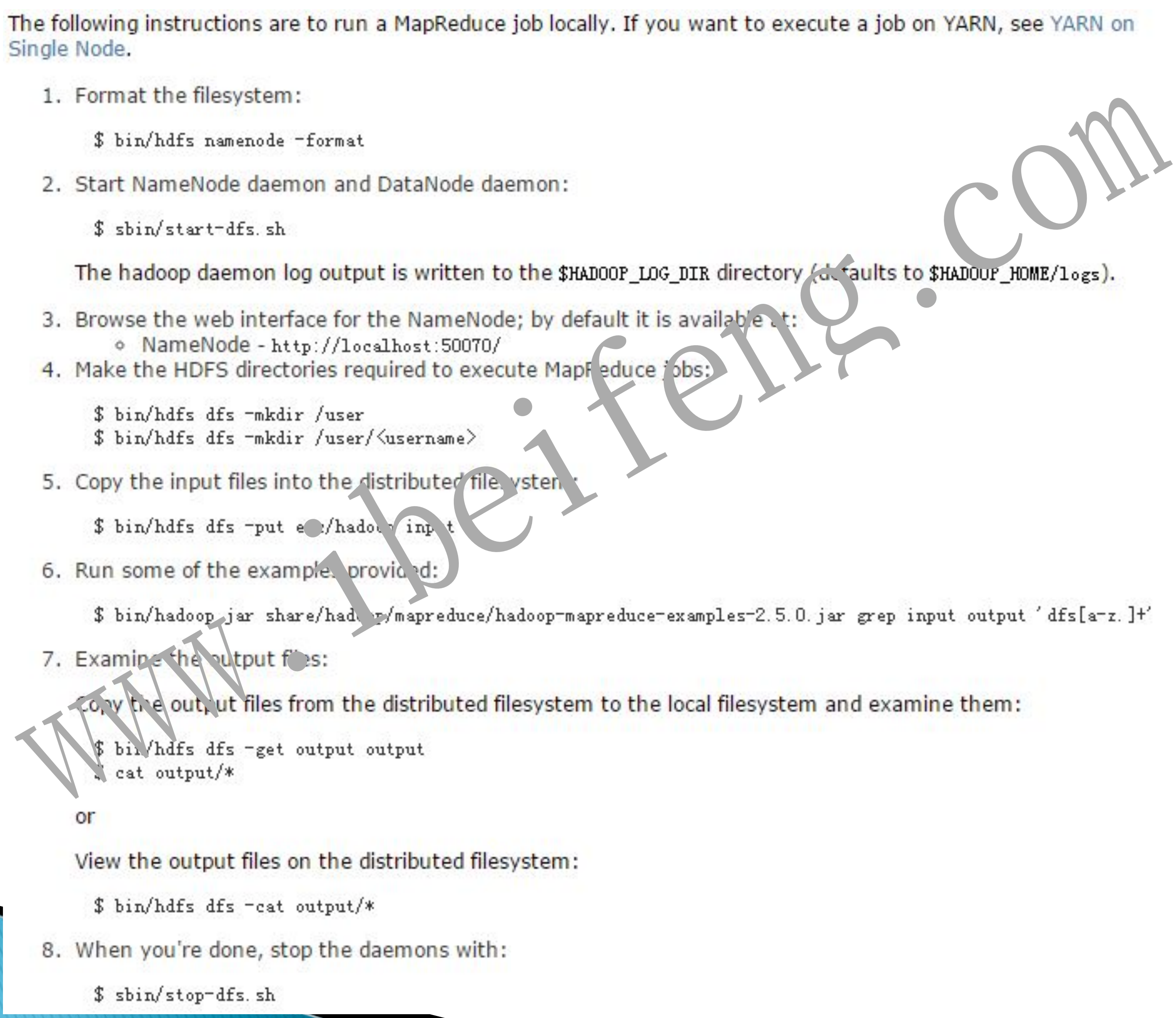
Now check that you can ssh to the localhost without a passphrase:

```
$ ssh localhost
```

If you cannot ssh to localhost without a passphrase, execute the following commands:

```
$ ssh-keygen -t dsa -P '' -f ~/.ssh/id_dsa
$ cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
```

Pseudo-Distributed Operation



The following instructions are to run a MapReduce job locally. If you want to execute a job on YARN, see YARN on Single Node.

1. Format the filesystem:

```
$ bin/hdfs namenode -format
```

2. Start NameNode daemon and DataNode daemon:

```
$ sbin/start-dfs.sh
```

   The hadoop daemon log output is written to the `$HADOOP_LOG_DIR` directory (defaults to `$HADOOP_HOME/logs`).

3. Browse the web interface for the NameNode; by default it is available at:
   - NameNode - `http://localhost:50070/`
4. Make the HDFS directories required to execute MapReduce jobs:

```
$ bin/hdfs dfs -mkdir /user
$ bin/hdfs dfs -mkdir /user/<username>
```

5. Copy the input files into the distributed filesystem:

```
$ bin/hdfs dfs -put etc/hadoop input
```

6. Run some of the examples provided:

```
$ bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.5.0.jar grep input output 'dfs[a-z.]+'
```

7. Examine the output files:

   Copy the output files from the distributed filesystem to the local filesystem and examine them:

```
$ bin/hdfs dfs -get output output
$ cat output/*
```

   or

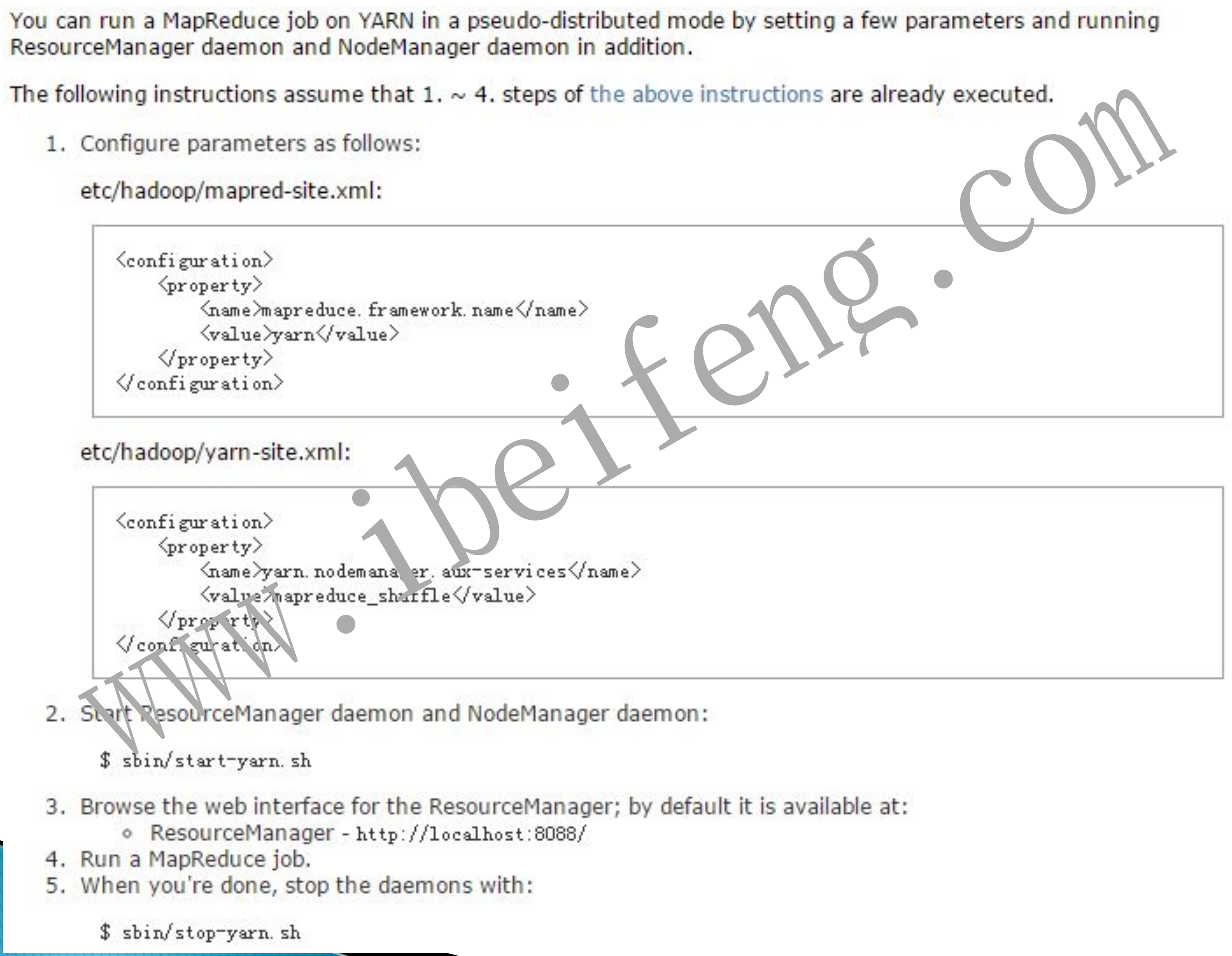
   View the output files on the distributed filesystem:

```
$ bin/hdfs dfs -cat output/*
```

8. When you're done, stop the daemons with:

```
$ sbin/stop-dfs.sh
```

Pseudo-Distributed Operation



You can run a MapReduce job on YARN in a pseudo-distributed mode by setting a few parameters and running ResourceManager daemon and NodeManager daemon in addition.

The following instructions assume that 1. ~ 4. steps of the above instructions are already executed.

1. Configure parameters as follows:

   etc/hadoop/mapred-site.xml:

```
<configuration>
    <property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
    </property>
</configuration>
```

   etc/hadoop/yarn-site.xml:

```
<configuration>
    <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
    </property>
</configuration>
```

2. Start ResourceManager daemon and NodeManager daemon:

```
  $ sbin/start-yarn.sh
```

3. Browse the web interface for the ResourceManager; by default it is available at:
   - ResourceManager - http://localhost:8088/
4. Run a MapReduce job.
5. When you're done, stop the daemons with:

```
  $ sbin/stop-yarn.sh
```

启动 HDFS

- 格式化 NameNode

```
$ bin/hdfs namenode -format
```

- 启动 NameNode

```
$ sbin/hadoop-daemon.sh start namenode
```

- 启动 DataNode

```
$ sbin/hadoop-daemon.sh start datanode
```

- HDFS 监控WEB页面

  http://hadoop-senior.ibeifeng.com:50070/

- 启动 SecondaryNameNode

```
$ sbin/hadoop-daemon.sh start secondarynamenode
```

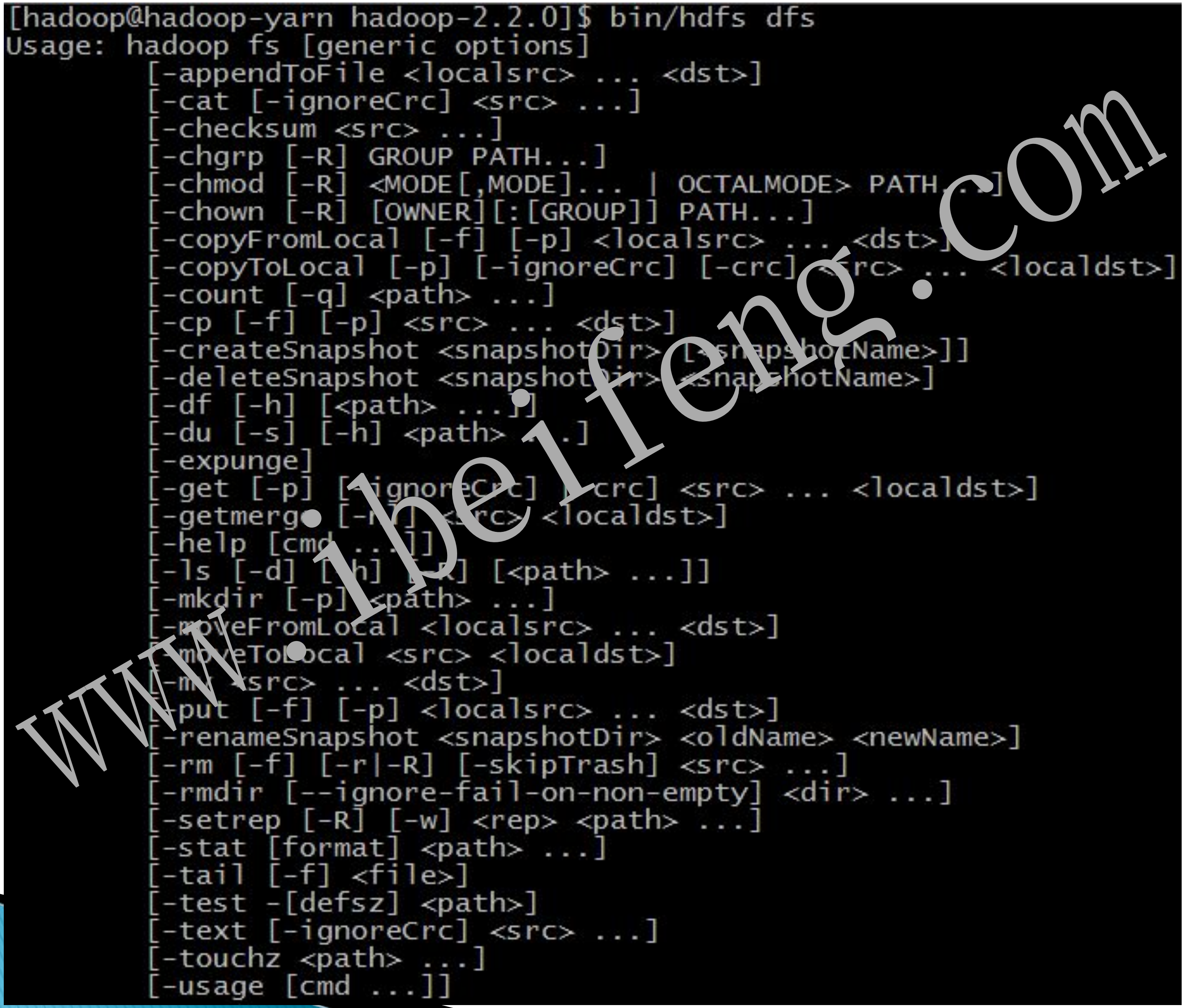
- SecondaryNameNode 监控WEB页面

  http://hadoop-yarn.dragon.org:50090/

日志文件

- 启动日志文件目录 $HADOOP_HOME/logs
- 分析日志文件的格式【log】和【out】
  - **.log**：通过log4j记录的，记录大部分应用程序的日志信息
  - **.out**：记录标准输出和标准错误日志，少量记录
- 日志文件的命名规则
  【框架名称-用户名-进程名-主机名-日志格式后缀】

HDFS Shell 命令



```
[hadoop@hadoop-yarn hadoop-2.2.0]$ bin/hdfs dfs
Usage: hadoop fs [generic options]
        [-appendToFile <localsrc> ... <dst>]
        [-cat [-ignoreCrc] <src> ...]
        [-checksum <src> ...]
        [-chgrp [-R] GROUP PATH...]
        [-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...]
        [-chown [-R] [OWNER][:[GROUP]] PATH...]
        [-copyFromLocal [-f] [-p] <localsrc> ... <dst>]
        [-copyToLocal [-p] [-ignoreCrc] [-crc] <src> ... <localdst>]
        [-count [-q] <path> ...]
        [-cp [-f] [-p] <src> ... <dst>]
        [-createSnapshot <snapshotDir> [<snapshotName>]]
        [-deleteSnapshot <snapshotDir> <snapshotName>]
        [-df [-h] [<path> ...]]
        [-du [-s] [-h] <path> ...]
        [-expunge]
        [-get [-p] [-ignoreCrc] [-crc] <src> ... <localdst>]
        [-getmerge [-nl] <src> <localdst>]
        [-help [cmd ...]]
        [-ls [-d] [-h] [-R] [<path> ...]]
        [-mkdir [-p] <path> ...]
        [-moveFromLocal <localsrc> ... <dst>]
        [-moveToLocal <src> <localdst>]
        [-mv <src> ... <dst>]
        [-put [-f] [-p] <localsrc> ... <dst>]
        [-renameSnapshot <snapshotDir> <oldName> <newName>]
        [-rm [-f] [-r|-R] [-skipTrash] <src> ...]
        [-rmdir [--ignore-fail-on-non-empty] <dir> ...]
        [-setrep [-R] [-w] <rep> <path> ...]
        [-stat [format] <path> ...]
        [-tail [-f] <file>]
        [-test -[defsz] <path>]
        [-text [-ignoreCrc] <src> ...]
        [-touchz <path> ...]
        [-usage [cmd ...]]
```

启动 YARN

- 启动 ResourceManager

```
$ sbin/yarn-daemon.sh start resourcemanager
```

- 启动 NodeManager

```
$ sbin/yarn-daemon.sh start nodemanager
```

- 查看启动守护进程

```
$ jps
```

- 查看日志
  - yarn-beifeng-resourcemanager-hadoop-senior.ibeifeng.com.log
  - yarn-beifeng-nodemanager-hadoop-senior.ibeifeng.com.log
- 登陆监控WEB页面

  http://hadoop-senior.ibeifeng.com:8088/

  http://hadoop-senior.ibeifeng.com:8042/

课程大纲

1 大数据应用发展前景

2 Hadoop 2.x概述

3 Hadoop 2.x生态系统

4 Hadoop 2.x环境搭建

5 初识MapReduce应用

运行MapReduce程序

- **PI 程序**

  $ bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.5.0.jar pi 10 20

- **WEB UI 监控**
  - Node 状况
  - MapReduce Application 状况
  - Container 分配
- **运行WordCount应用**

历史服务器

- 查看已经运行完成的MapReduce作业记录，比如用了多少个Map、用了多少个Reduce、作业提交时间、作业启动时间、作业完成时间等信息。
- 默认情况下，历史服务器是没有启动的。
- 启动

```
$ sbin/mr-jobhistory-daemon.sh start historyserver
```

- WEB UI

http://hadoop-senior.ibeifeng.com:19888/

- 停止

```
$ sbin/mr-jobhistory-daemon.sh stop historyserver
```

Hadoop 2.x 配置文件

- **默认的**
  - 位置：$HADOOP_HOM/share/hadoop 中对应project文件夹中的Jar文件中
  - 文件名：core-default.xml、hdfs-default.xml、yarn-default.xml、mapred-default.xml
- **自定义的**
  - 位置：$HADOOP_HOME/etc/hadoop
  - 文件名：core-site.xml、hdfs-site.xml、yarn-site.xml、mapred-site.xml

启动 HDFS 和 YARN 方式

- 启动方式
  - 方式一，逐一启动

    hadoop-daemon.sh、yarn-daemon.sh
  - 方式二，分开启动

    start-dfs.sh、start-yarn.sh
  - 方式三，一起启动

    start-all.sh

配置 SSH 无密钥登陆

- 使用密码登录，每次都必须输入密码，非常麻烦。SSH还提供了公钥登录，可以省去输入密码的步骤。
- 所谓"公钥登录"，就是用户将自己的公钥储存在远程主机上。登录的时候，远程主机会向用户发送一段随机字符串，用户用自己的私钥加密后，再发回来。远程主机用事先储存的公钥进行解密，如果成功，就证明用户是可信的，直接允许登录shell，不再要求密码。
- 生成公钥/私钥对

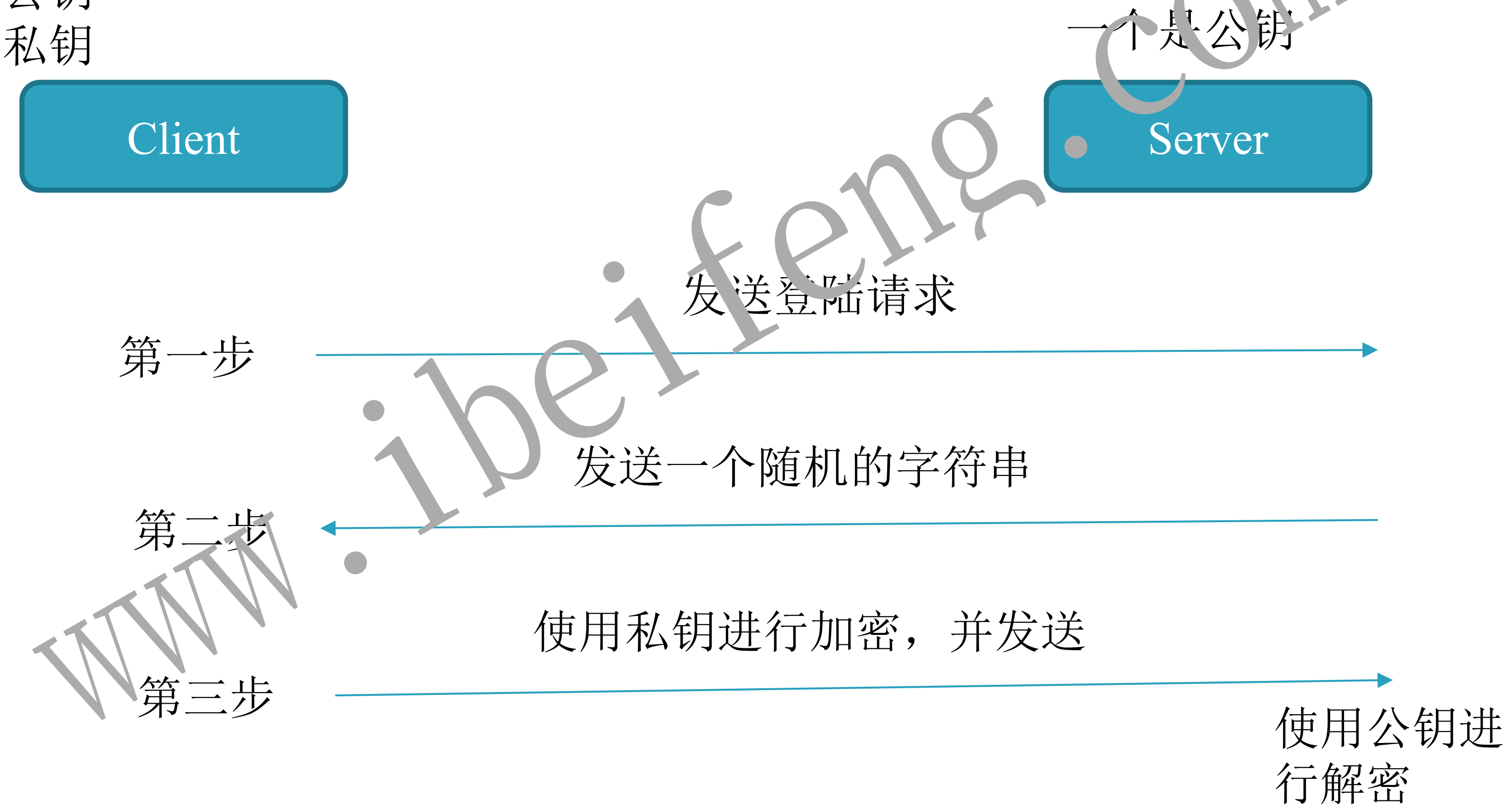
```
$ ssh-keygen -t rsa
```

- 在$HOME/.ssh/目录下，会新生成两个文件：id_rsa.pub和id_rsa。前者是你的公钥，后者是你的私钥。
- 拷贝公钥至authorized_keys文件

```
$ ssh-copy-id hadoop-senior.ibeifeng.com
```

配置 SSH 无密钥登陆

一对加密
一个是公钥
一个是私钥



一个是公钥

Client

Server

发送登陆请求

第一步

发送一个随机的字符串

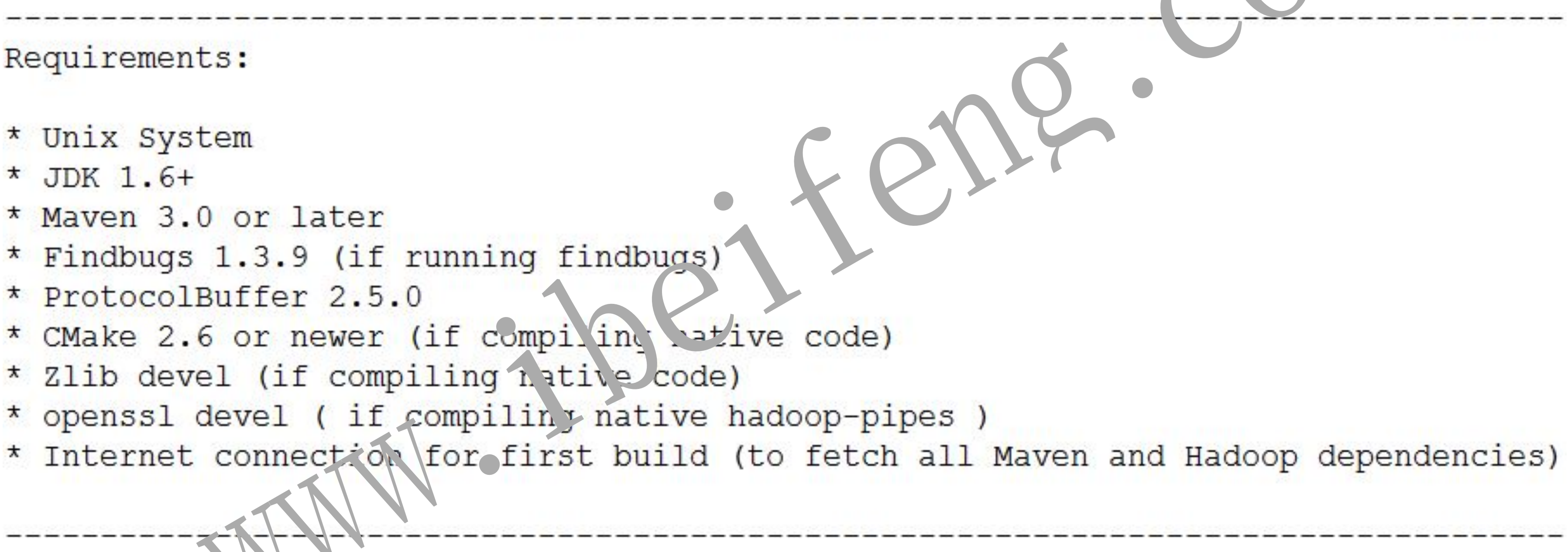
第二步

使用私钥进行加密，并发送

第三步

使用公钥进行解密

Requirements

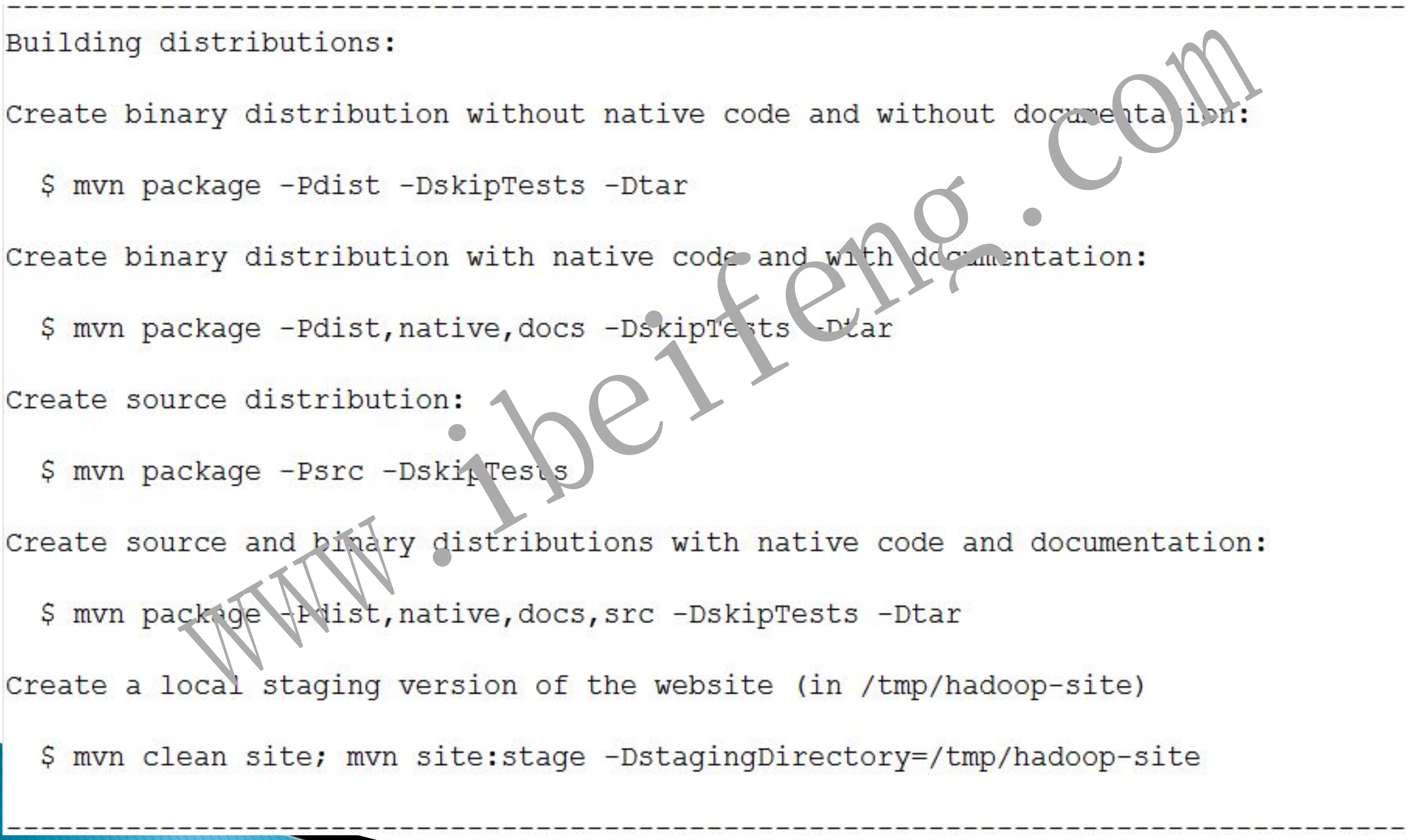


```
----------------------------------------------------------------------------------
Requirements:

* Unix System
* JDK 1.6+
* Maven 3.0 or later
* Findbugs 1.3.9 (if running findbugs)
* ProtocolBuffer 2.5.0
* CMake 2.6 or newer (if compiling native code)
* Zlib devel (if compiling native code)
* openssl devel ( if compiling native hadoop-pipes )
* Internet connection for first build (to fetch all Maven and Hadoop dependencies)

----------------------------------------------------------------------------------
```

编译安装



```
----------------------------------------------------------------------------------
Building distributions:

Create binary distribution without native code and without documentation:

  $ mvn package -Pdist -DskipTests -Dtar

Create binary distribution with native code and with documentation:

  $ mvn package -Pdist,native,docs -DskipTests -Dtar

Create source distribution:

  $ mvn package -Psrc -DskipTests

Create source and binary distributions with native code and documentation:

  $ mvn package -Pdist,native,docs,src -DskipTests -Dtar

Create a local staging version of the website (in /tmp/hadoop-site)

  $ mvn clean site; mvn site:stage -DstagingDirectory=/tmp/hadoop-site

----------------------------------------------------------------------------------
```