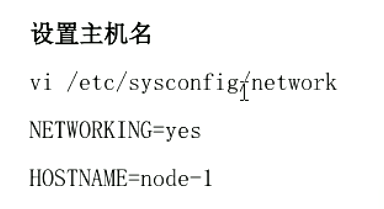
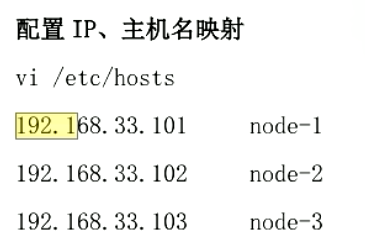
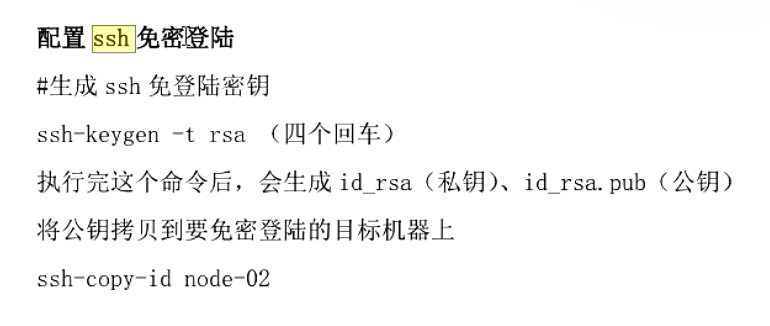
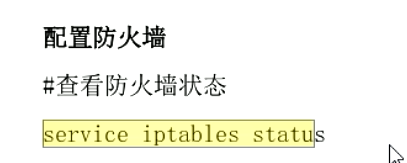
# 配置环境

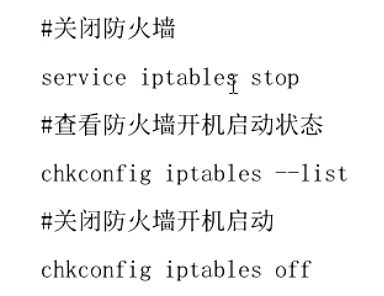










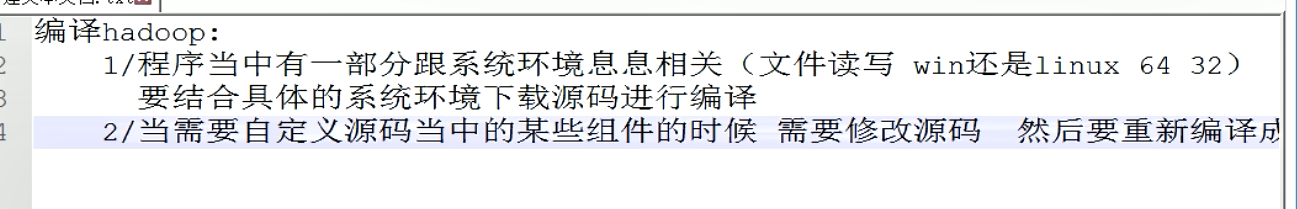


安装jdk、配置jdk环境变量

sudo vim ~/.bashrc



source /etc/profile(重新加载环境变量)

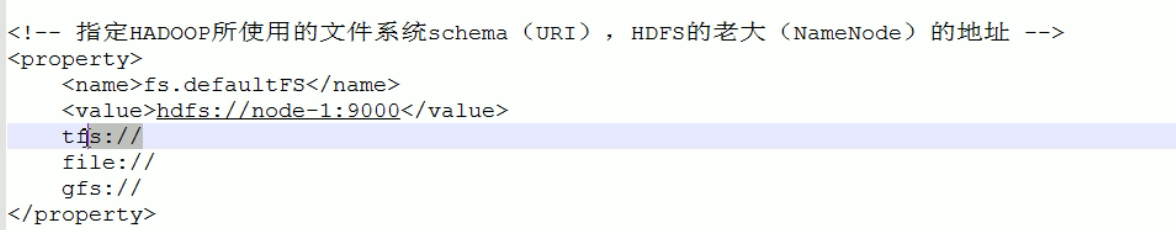


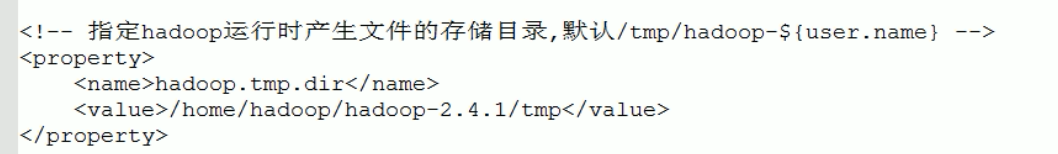
配置hadoop（配置文件在Hadoop安装目录的etc的hadoop目录下）:

1. 配置hadoop-env.sh

配置jdk路径



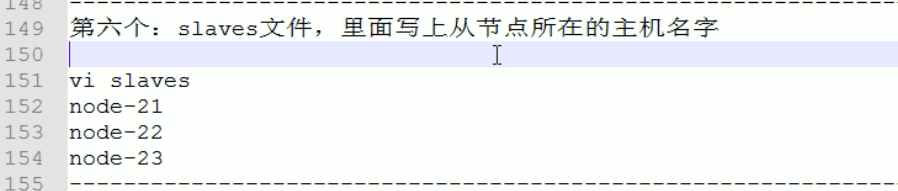


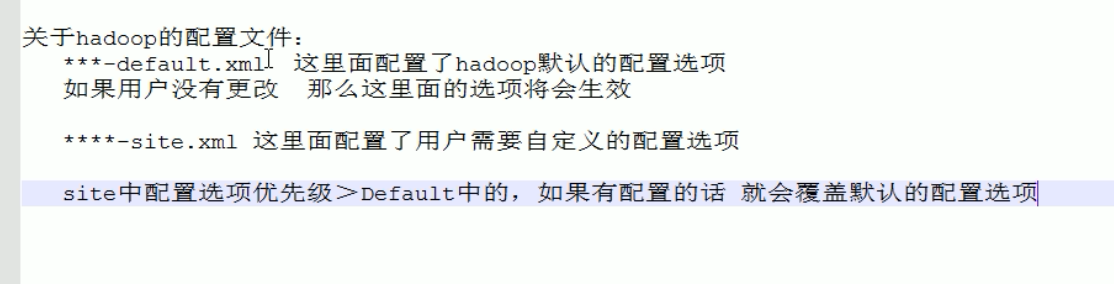




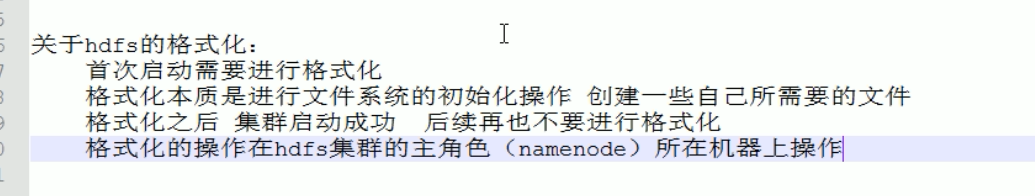




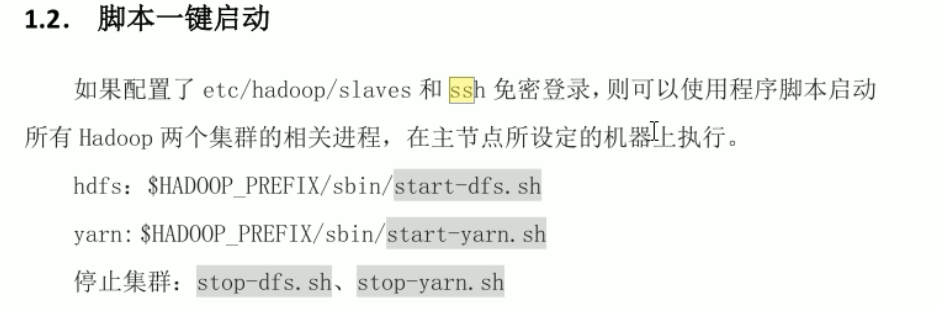






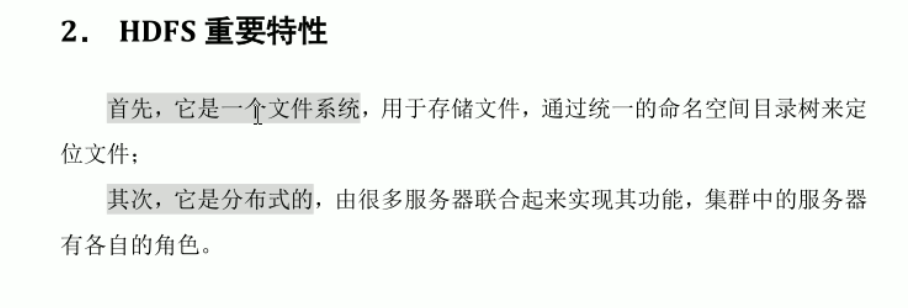


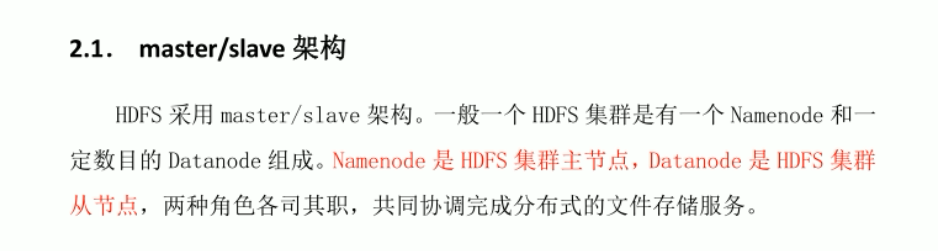


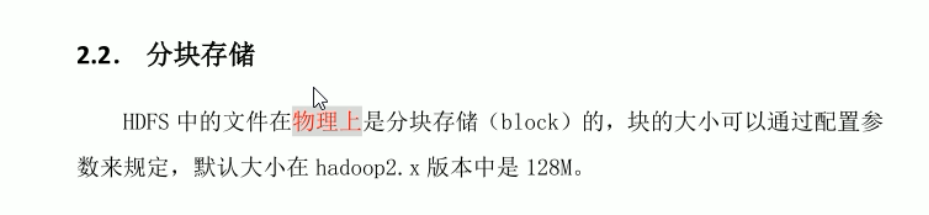


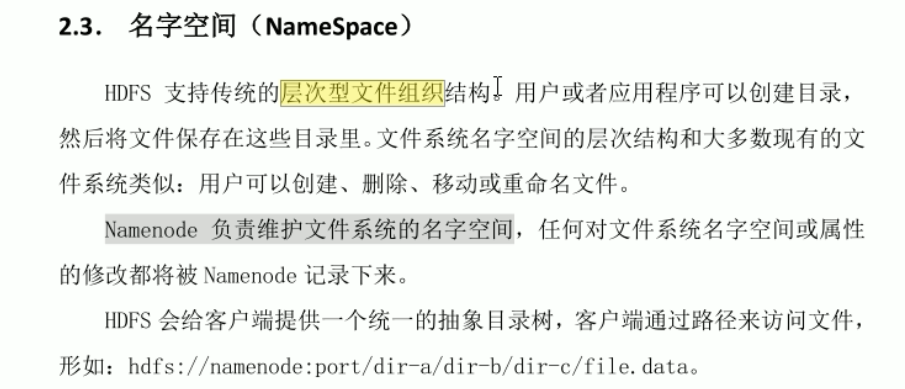


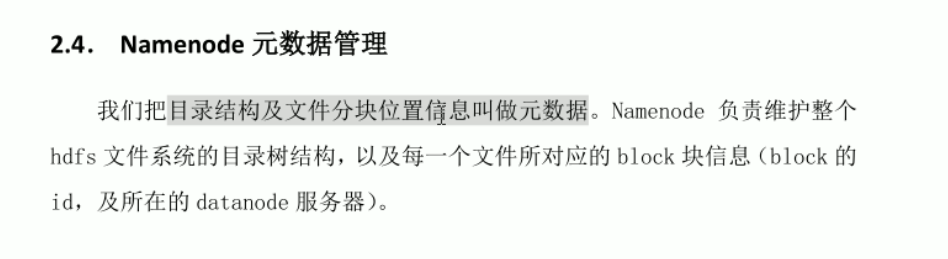


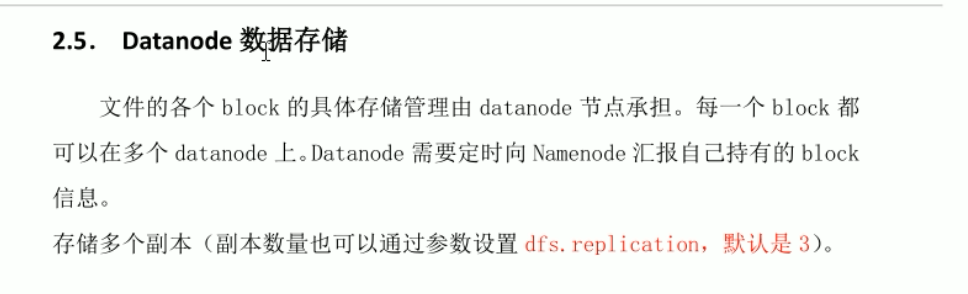


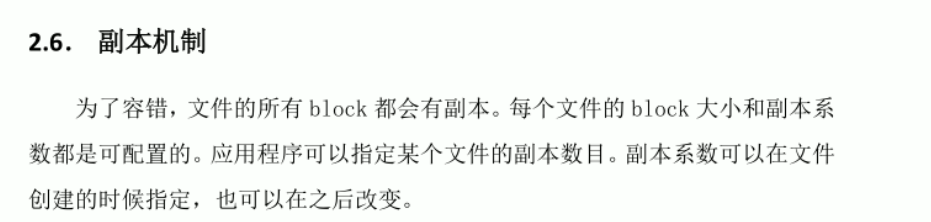


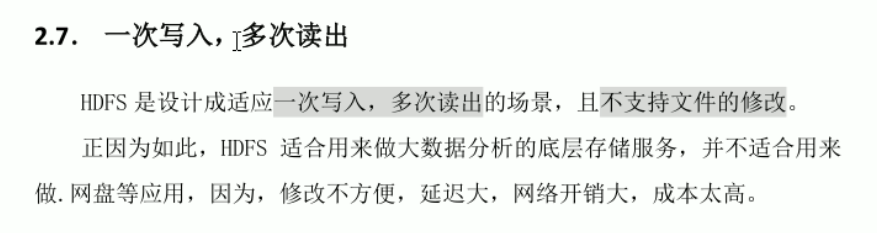


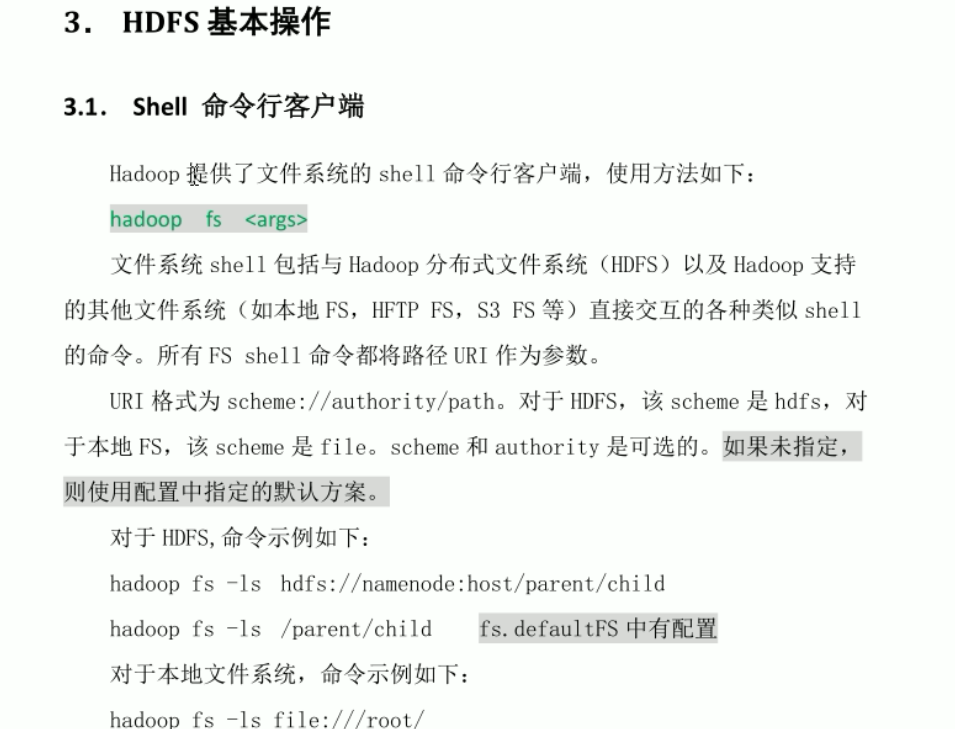




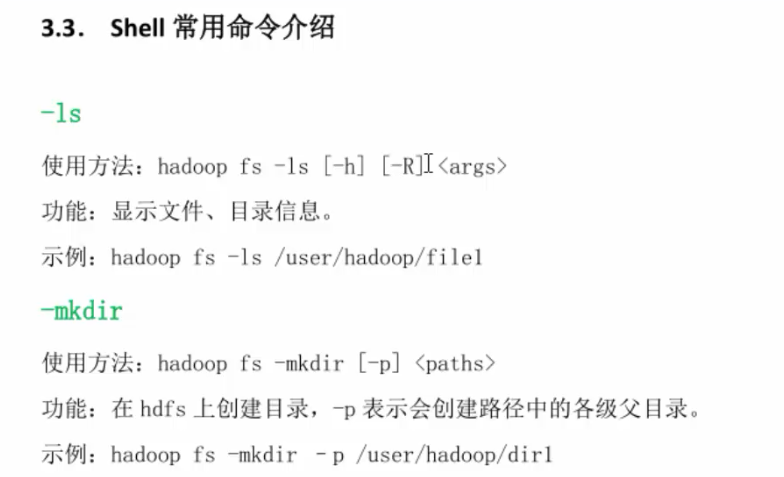


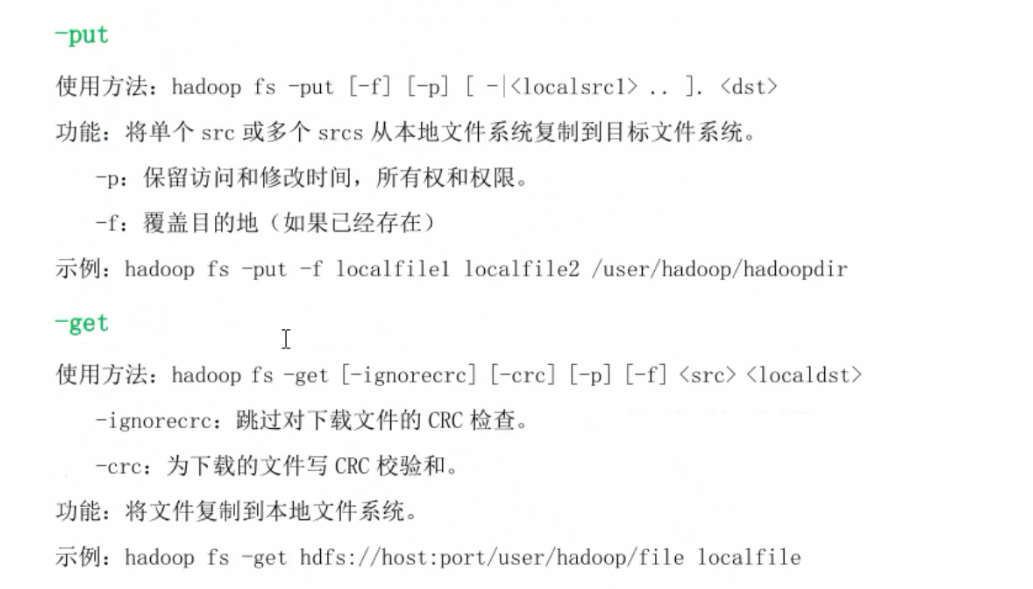


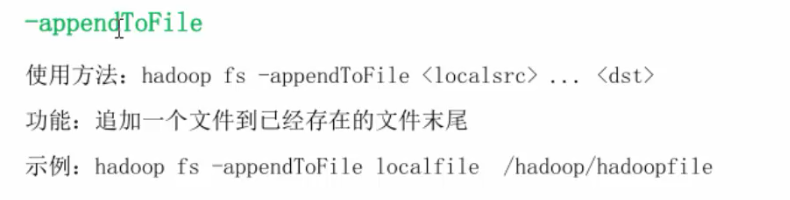


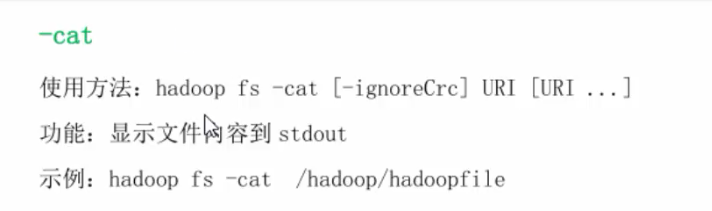


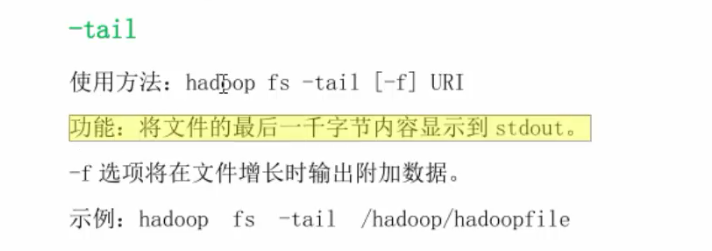


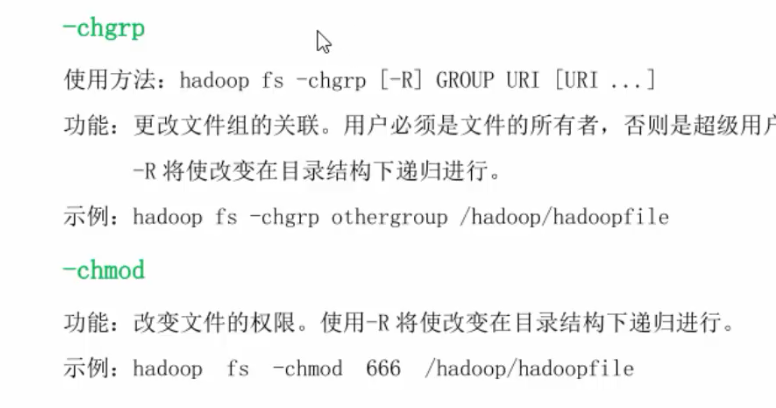








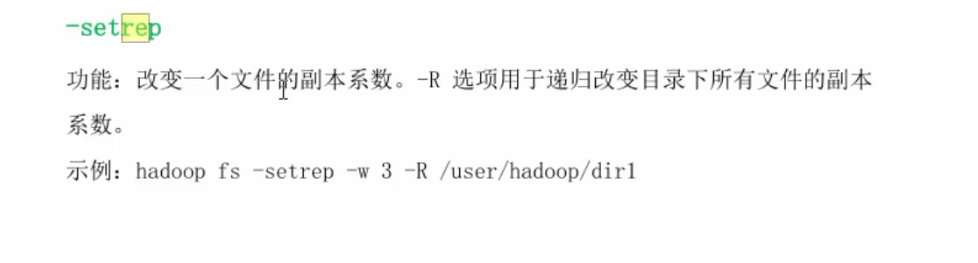


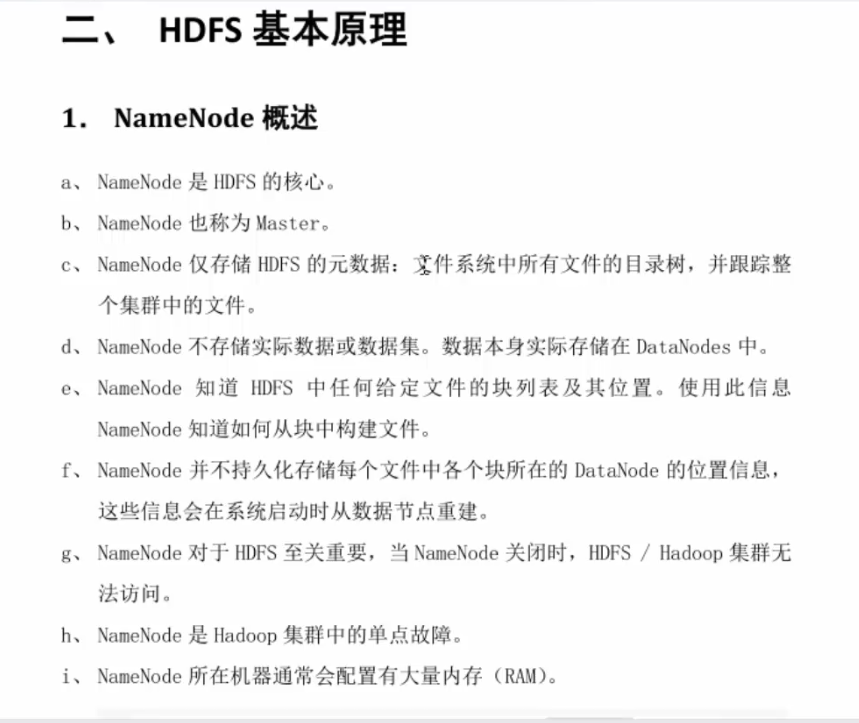


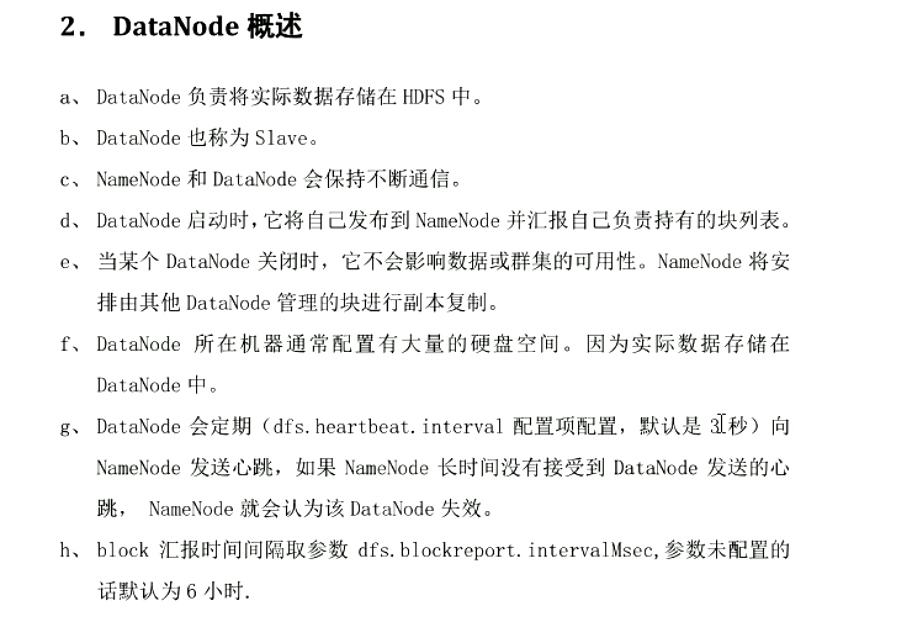


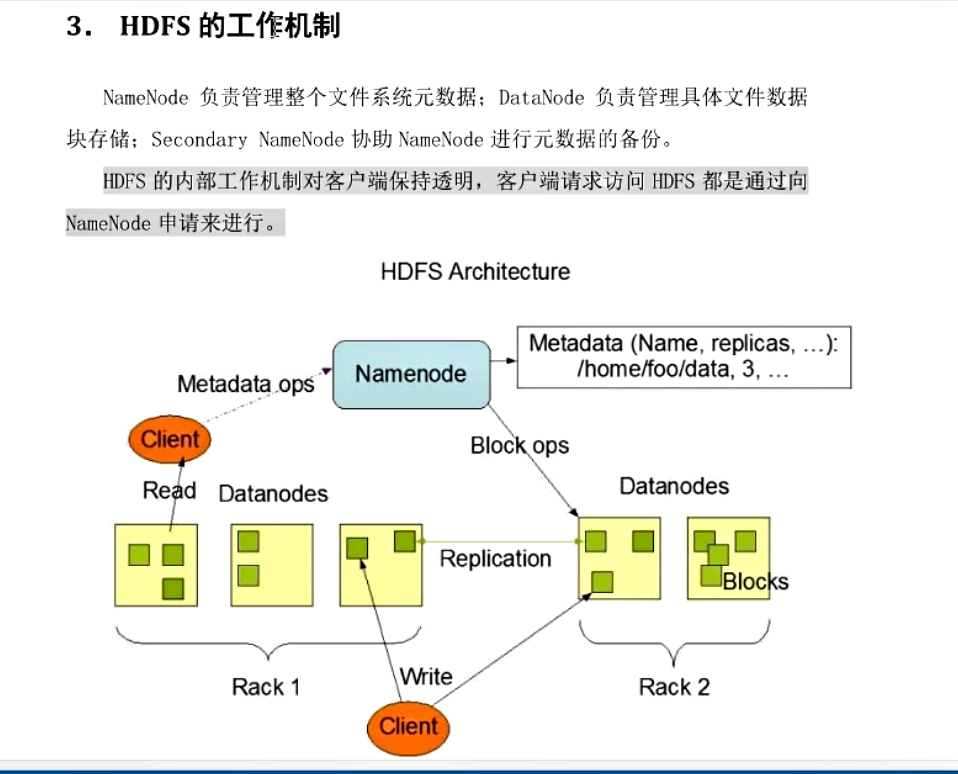


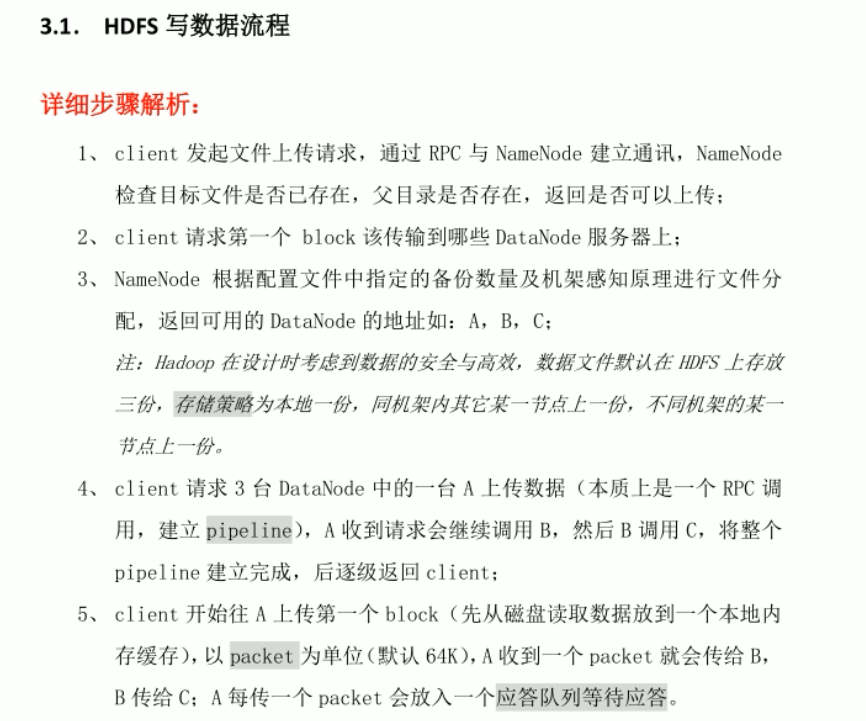


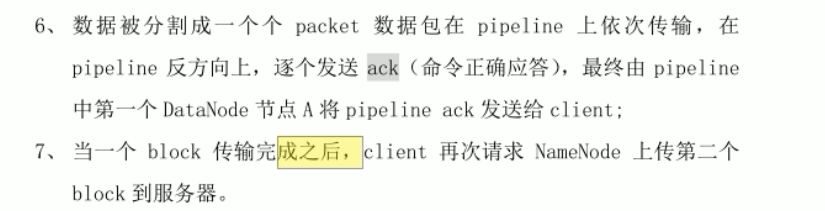




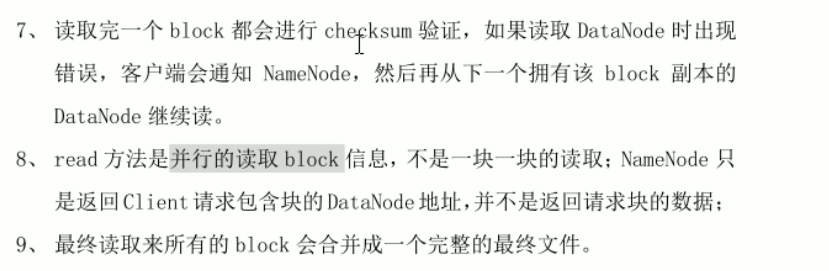




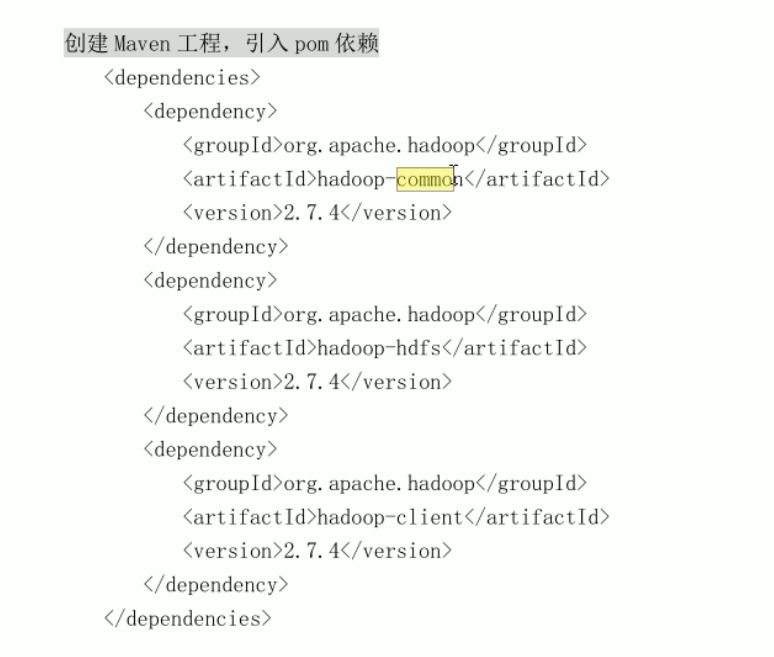


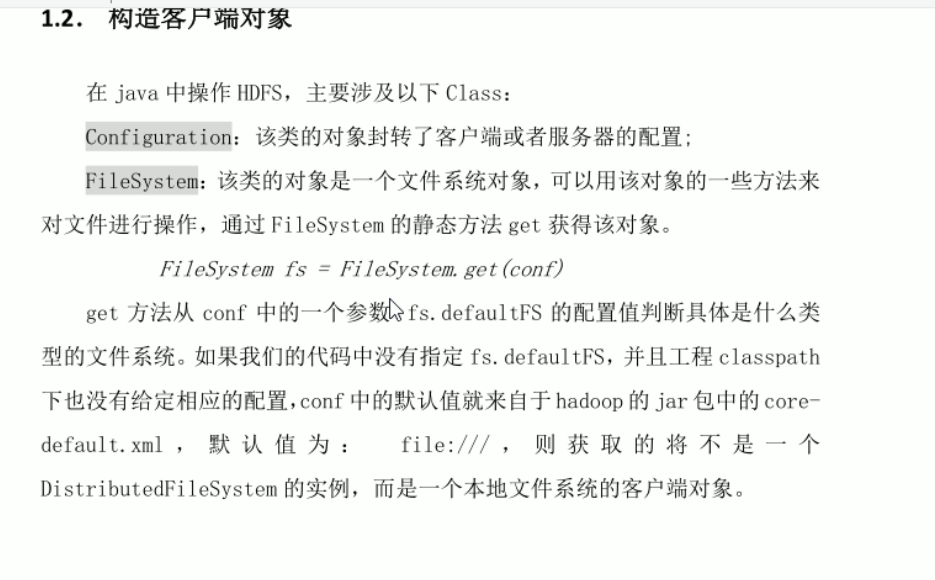




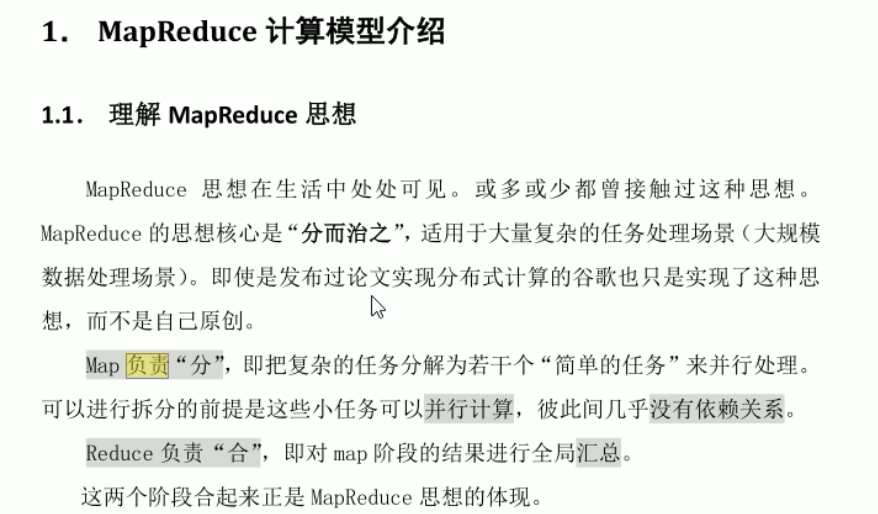


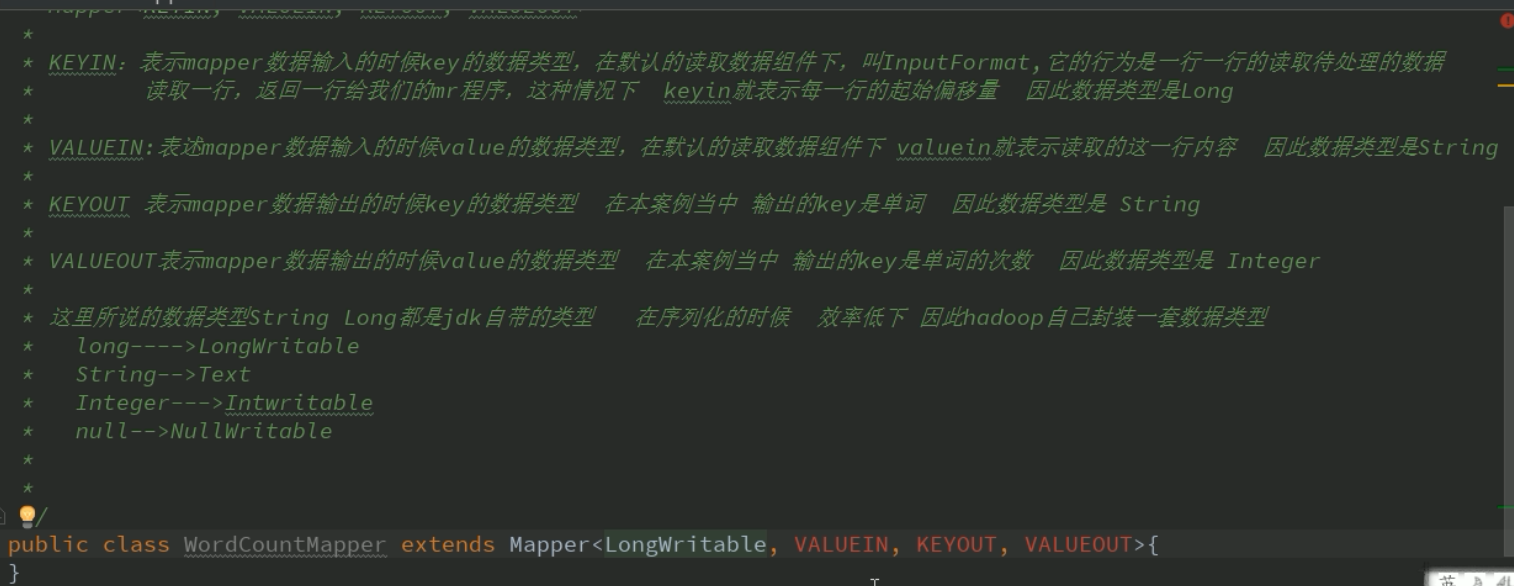


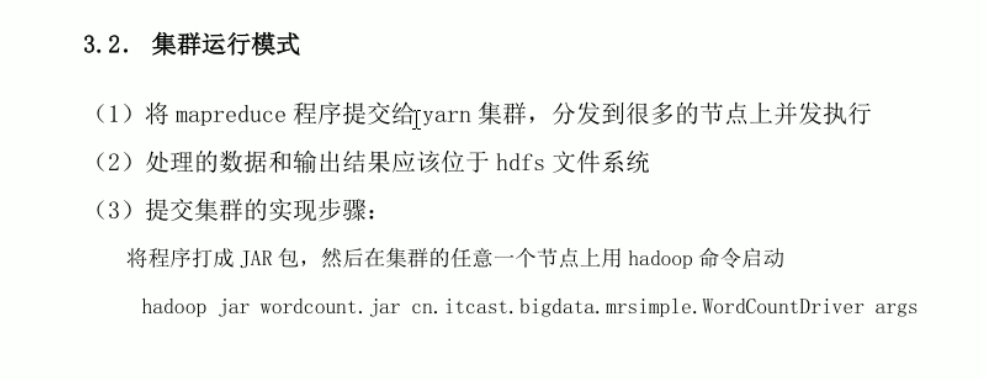


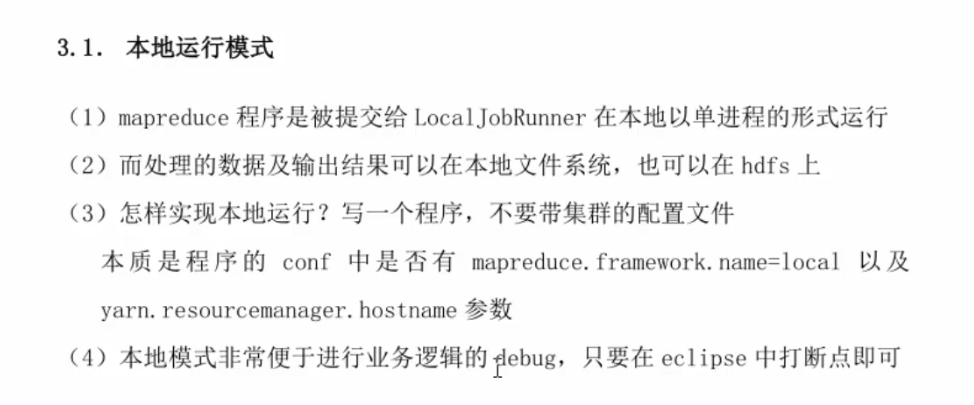












# Apache Flume

1. 安装和配置
   1. 解压，然后在conf的flume-env.sh中配置java的路径
   2. 配置采集方案sources、channels和sinks

#定义这个agent组件中各组件的名字

a1.sources=r1

a1.sinks = k1

a1.channels = c1

#描述和配置souce组件：r1

a1.sources.r1.type = netcat

a1.sources.r1.bind = localhost

a1.sources.r1.port = 44444

#描述和配置sink组件 k1

a1.sinks.k1.type = logger

#描述和配置channel组件，此处使用是内存缓存的方式

a1.channels.c1.type = memory

a1.channels.c1.capacity = 1000

a1.channels.c1.transactionCapacity = 100

#描述和配置source channel sink之间的连接关系

a1.sources.r1.channels = c1

#当a1.sinks.k1.channels会报错

a1.sinks.k1.channel = c1

* 1. 启动命令

bin/flume-ng agent --conf conf/ --conf-file conf/netcat-logger.conf --name a1 -Dflume.root.logger=INFO,console

1. 采集目录到hdfs的采集方案

# Name the components on this agent

a1.sources = r1

a1.sinks = k1

a1.channels = c1

# Describe/configure the source

##注意：不能往监控目中重复丢同名文件,否则flume报错之后不会再进行工作

a1.sources.r1.type = spooldir

a1.sources.r1.spoolDir = /root/logs2

a1.sources.r1.fileHeader = true

# Describe the sink

a1.sinks.k1.type = hdfs

a1.sinks.k1.channel = c1

a1.sinks.k1.hdfs.path = /flume/events/%y-%m-%d/%H%M/

a1.sinks.k1.hdfs.filePrefix = events-

#控制文件夹的滚动频率

a1.sinks.k1.hdfs.round = true

a1.sinks.k1.hdfs.roundValue = 10

a1.sinks.k1.hdfs.roundUnit = minute

#控制文件的滚动频率

a1.sinks.k1.hdfs.rollInterval = 3 #时间维度

a1.sinks.k1.hdfs.rollSize = 20　　#文件大小维度

a1.sinks.k1.hdfs.rollCount = 5　　#event数量维度

a1.sinks.k1.hdfs.batchSize = 1

a1.sinks.k1.hdfs.useLocalTimeStamp = true

#生成的文件类型，默认是Sequencefile，可用DataStream，则为普通文本

a1.sinks.k1.hdfs.fileType = DataStream

# Use a channel which buffers events in memory

a1.channels.c1.type = memory

a1.channels.c1.capacity = 1000

a1.channels.c1.transactionCapacity = 100

# Bind the source and sink to the channel

a1.sources.r1.channels = c1

a1.sinks.k1.channel = c1

1. 采集文件到hdfs

这个的意义在于由于很多日志文件是不断追加在一个log文件的，所以用此采集方案可以将新的日志不断放至hdfs中

# Name the components on this agent

a1.sources = r1

a1.sinks = k1

a1.channels = c1

# Describe/configure the source

a1.sources.r1.type = exec

a1.sources.r1.command = tail -F /root/logs/test.log

a1.sources.r1.channels = c1

# Describe the sink

a1.sinks.k1.type = hdfs

a1.sinks.k1.channel = c1

a1.sinks.k1.hdfs.path = /flume/tailout/%y-%m-%d/%H-%M/

a1.sinks.k1.hdfs.filePrefix = events-

a1.sinks.k1.hdfs.round = true

a1.sinks.k1.hdfs.roundValue = 10

a1.sinks.k1.hdfs.roundUnit = minute

a1.sinks.k1.hdfs.rollInterval = 3

a1.sinks.k1.hdfs.rollSize = 20

a1.sinks.k1.hdfs.rollCount = 5

a1.sinks.k1.hdfs.batchSize = 1

a1.sinks.k1.hdfs.useLocalTimeStamp = true

#生成的文件类型，默认是Sequencefile，可用DataStream，则为普通文本

a1.sinks.k1.hdfs.fileType = DataStream

# Use a channel which buffers events in memory

a1.channels.c1.type = memory

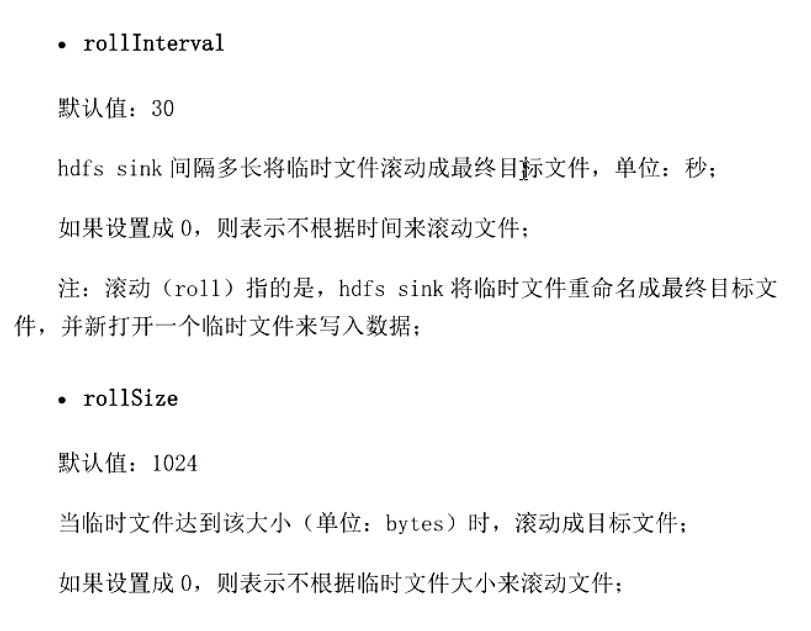
a1.channels.c1.capacity = 1000

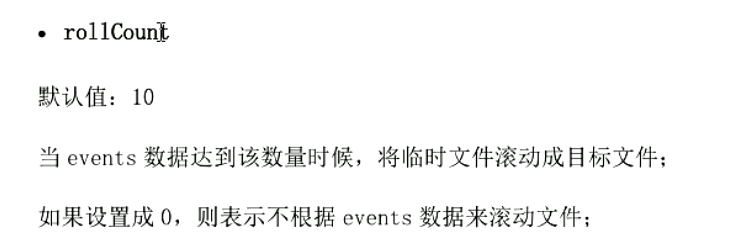
a1.channels.c1.transactionCapacity = 100

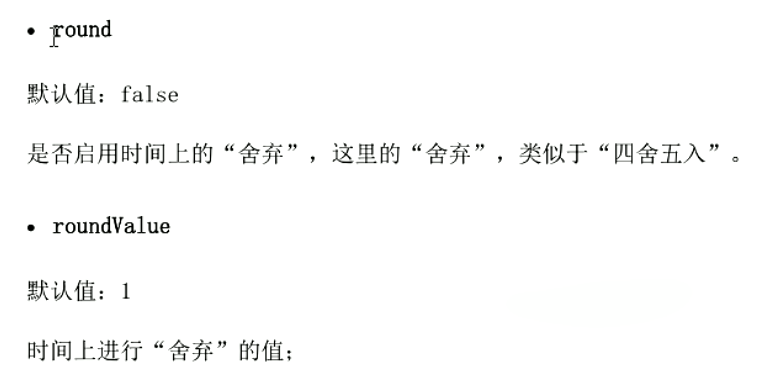
# Bind the source and sink to the channel

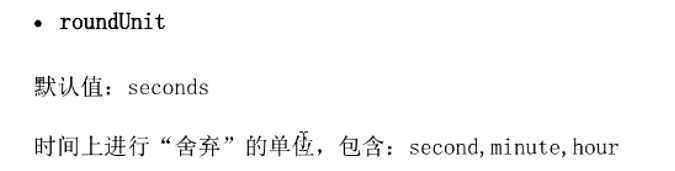
a1.sources.r1.channels = c1

a1.sinks.k1.channel = c1

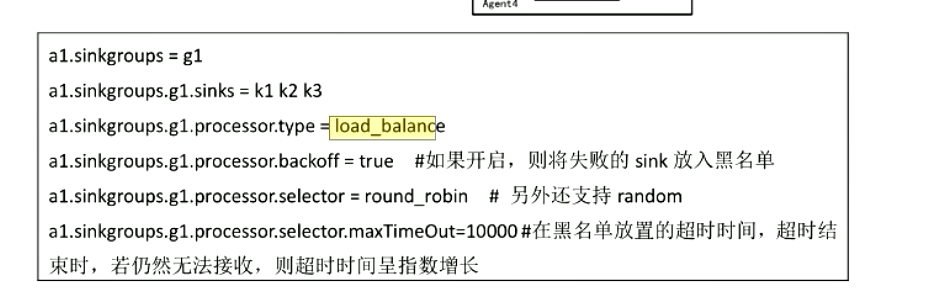








1. 负载均衡



host1 load-balance设置

host2 机器1

host3 机器2

#### Host1：

#Define a memory channel called c1 on a1

a1.channels = c1

a1.sources = r1

a1.sinks = k1 k2

a1.sinkgroups = g1

a1.sinkgroups.g1.sinks = k1 k2

a1.sinkgroups.g1.processor.type = load\_balance

a1.sinkgroups.g1.processor.selector = round\_robin

a1.sinkgroups.g1.processor.backoff = true

a1.channels.c1.type = file

a1.channels.c1.checkpointDir = /tmp/flume/loadcheckpoint

a1.channels.c1.dataDirs = /tmp/flume/loaddata

a1.sources.r1.channels = c1

a1.sources.r1.type = avro

a1.sources.r1.bind = 0.0.0.0

a1.sources.r1.port = 41415

a1.sinks.k1.channel = c1

a1.sinks.k1.type = avro

a1.sinks.k1.hostname = host2

a1.sinks.k1.port = 41414

a1.sinks.k2.channel = c1

a1.sinks.k2.type = AVRO

a1.sinks.k2.hostname = host3

a1.sinks.k2.port = 41414

#### Host2：

a2.channels = c1

a2.sources = r1

a2.sinks = k1

a2.channels.c1.type = FILE

a1.channels.c1.checkpointDir = /tmp/flume/checkpoint

a1.channels.c1.dataDirs = /tmp/flume/data

a2.sources.r1.channels = c1

a2.sources.r1.type = AVRO

a2.sources.r1.bind = 0.0.0.0

a2.sources.r1.port = 41414

a2.sinks.k1.channel = c1

a2.sinks.k1.type = file\_roll

a2.sinks.k1.sink.directory = /tmp/load/

a2.sinks.k1.sink.rollInterval = 0

#### Host3:

a2.channels = c1

a2.sources = r1

a2.sinks = k1

a2.channels.c1.type = FILE

a1.channels.c1.checkpointDir = /tmp/flume/checkpoint

a1.channels.c1.dataDirs = /tmp/flume/data

a2.sources.r1.channels = c1

a2.sources.r1.type = AVRO

a2.sources.r1.bind = 0.0.0.0

a2.sources.r1.port = 41414

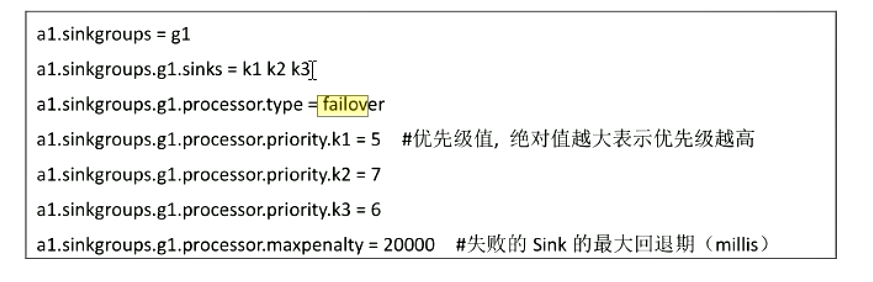
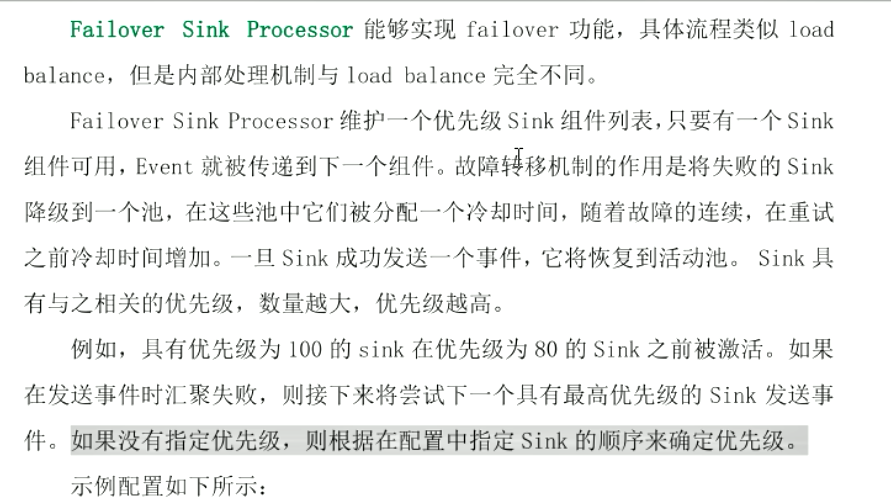
a2.sinks.k1.channel = c1

a2.sinks.k1.type = file\_roll

a2.sinks.k1.sink.directory = /tmp/load/

a2.sinks.k1.sink.rollInterval = 0

1. 容错



1. Flume拦截器

拦截器是简单的插件式组件，设置在source和channel之间。source接收到的时间，在写入channel之前，拦截器都可以进行转换或者删除这些事件。每个拦截器只处理同一个source接收到的事件。可以自定义拦截器。

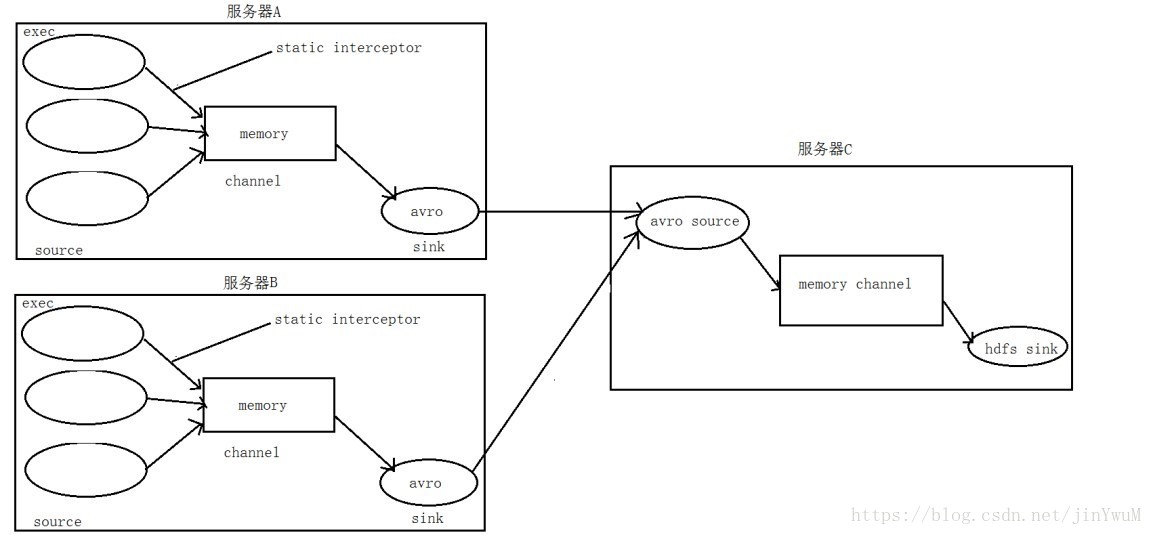
假设有两台服务器（A、B ）收集实时生产日志，日志类型分别是：access.log、nginx.log、web.log

将两台服务器的日志汇总到一台服务器上（C），然后统计存储到HDFS文件系统中，且存储目录结构为

/source/logs/access/20180911/events.x

/source/logs/nginx/20180911/ events.x

/source/logs/web/20180911/ events.x



1. 在服务器 node01、node02 上创建配置文件 exec\_source.properties

# Name the components on this agent

a1.sources = r1 r2 r3

a1.sinks = k1

a1.channels = c1

# Describe/configure the source

# 元数据类型

a1.sources.r1.type = exec

a1.sources.r1.command = tail -F /export/data/logs/access.log

# 拦截器别名

a1.sources.r1.interceptors = i1

# 拦截器类型

a1.sources.r1.interceptors.i1.type = static

# 拦截的键值对

a1.sources.r1.interceptors.i1.key = type

a1.sources.r1.interceptors.i1.value = access

a1.sources.r2.type = exec

a1.sources.r2.command = tail -F /export/data/logs/nginx.log

a1.sources.r2.interceptors = i2

a1.sources.r2.interceptors.i2.type = static

a1.sources.r2.interceptors.i2.key = type

a1.sources.r2.interceptors.i2.value = nginx

a1.sources.r3.type = exec

a1.sources.r3.command = tail -F /export/data/logs/web.log

a1.sources.r3.interceptors = i3

a1.sources.r3.interceptors.i3.type = static

a1.sources.r3.interceptors.i3.key = type

a1.sources.r3.interceptors.i3.value = web

# 下沉的目的地

# Describe the sink

a1.sinks.k1.type = avro

a1.sinks.k1.hostname = node03

a1.sinks.k1.port = 41414

# Use a channel which buffers events in memory

a1.channels.c1.type = memory

a1.channels.c1.capacity = 2000000

a1.channels.c1.transactionCapacity = 100000

# Bind the source and sink to the channel

a1.sources.r1.channels = c1

a1.sources.r2.channels = c1

a1.sources.r3.channels = c1

a1.sinks.k1.channel = c1

1. 在服务器 node03 上创建配置文件avro\_source\_hdfs.properties

# 定义agent名， source、channel、sink的名称

a1.sources = r1

a1.sinks = k1

a1.channels = c1

# 定义source

a1.sources.r1.type = avro

a1.sources.r1.bind = node03

a1.sources.r1.port =41414

# 添加时间拦截器，将没有时间戳的event添加上时间戳

a1.sources.r1.interceptors = i1

a1.sources.r1.interceptors.i1.type = org.apache.flume.interceptor.TimestampInterceptor$Builder

# 定义channels

a1.channels.c1.type = memory

a1.channels.c1.capacity = 20000

a1.channels.c1.transactionCapacity = 10000

# 定义sink，%{type}为获取到上一层定义的变量值

a1.sinks.k1.type = hdfs

a1.sinks.k1.hdfs.path=hdfs://node01:9000/source/logs/%{type}/%Y%m%d

a1.sinks.k1.hdfs.filePrefix =events

a1.sinks.k1.hdfs.fileType = DataStream

a1.sinks.k1.hdfs.writeFormat = Text

# 时间类型

#a1.sinks.k1.hdfs.useLocalTimeStamp = true

# 生成的文件不按条数生成

a1.sinks.k1.hdfs.rollCount = 0

# 生成的文件不按时间生成

a1.sinks.k1.hdfs.rollInterval = 30

# 生成的文件按大小生成

a1.sinks.k1.hdfs.rollSize = 10485760

#a1.sinks.k1.hdfs.rollSize =0

# 批量写入hdfs的个数

a1.sinks.k1.hdfs.batchSize = 20

# flume操作hdfs的线程数（包括新建，写入等）

a1.sinks.k1.hdfs.threadsPoolSize=10

# 操作hdfs超时时间

a1.sinks.k1.hdfs.callTimeout=30000

# 组装source、channel、sink

a1.sources.r1.channels = c1

a1.sinks.k1.channel = c1

自定义拦截器详情看https://blog.csdn.net/jinywum/article/details/82598947