

实验一、

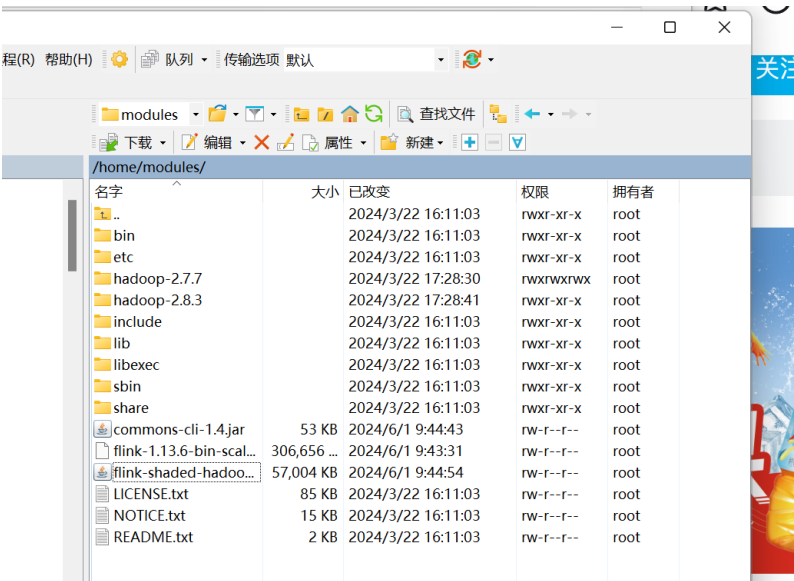
1. 打开 8088、8081、8032 端口：

安全组——入方向规则——添加规则



2. 下载 flink 和依赖包

下载好后，上传到/home/modules



解压缩：

```

etc          hadoop-2.8.3 lib NOTICE.txt sha
[root@cpy-2021211138 modules]# tar -zxvf flink-1.13.6-bin-scala_2.11.tgz
flink-1.13.6/
flink-1.13.6/lib/
flink-1.13.6/lib/flink-shaded-zookeeper-3.4.14.jar
flink-1.13.6/lib/flink-json-1.13.6.jar
flink-1.13.6/lib/log4j-1.2-api-2.17.1.jar
flink-1.13.6/lib/flink-table-blink_2.11-1.13.6.jar
flink-1.13.6/lib/flink-csv-1.13.6.jar
flink-1.13.6/lib/flink-dist_2.11-1.13.6.jar
flink-1.13.6/lib/log4j-slf4j-impl-2.17.1.jar
flink-1.13.6/lib/flink-table_2.11-1.13.6.jar

```

重命名

```

cp: omitting directory 'flink-1.13.6'
[root@cpy-2021211138 modules]# mv flink-1.13.6 flink
[root@cpy-2021211138 modules]# ls
bin          flink          hadoop-2.8.3 lib
commons-cli-1.4.jar flink-1.13.6-bin-scala_2.11.tgz hadoop-2.8.3 libexec
etc          flink-shaded-hadoop-3-uber-3.1.1.7.2.1.0-327-9.0.jar include LICENSE.txt
[root@cpy-2021211138 modules]# |

```

3. 修改环境变量（四台服务器都要改）：

/etc/profile

添加：

```

# Flink
export FLINK_HOME=/home/modules/flink
export PATH=$FLINK_HOME/bin:$PATH

```

再刷新一下环境变量：

```

commons-cli-1.4.jar flink-1.13.6-bin-scala_2.11.tgz
etc          flink-shaded-hadoop-3-uber-3.1.1.7.2.1.0-327-9.0.jar
[root@cpy-2021211138 modules]# source /etc/profile
[root@cpy-2021211138 modules]# |

```

4. 脚本启动 Flink

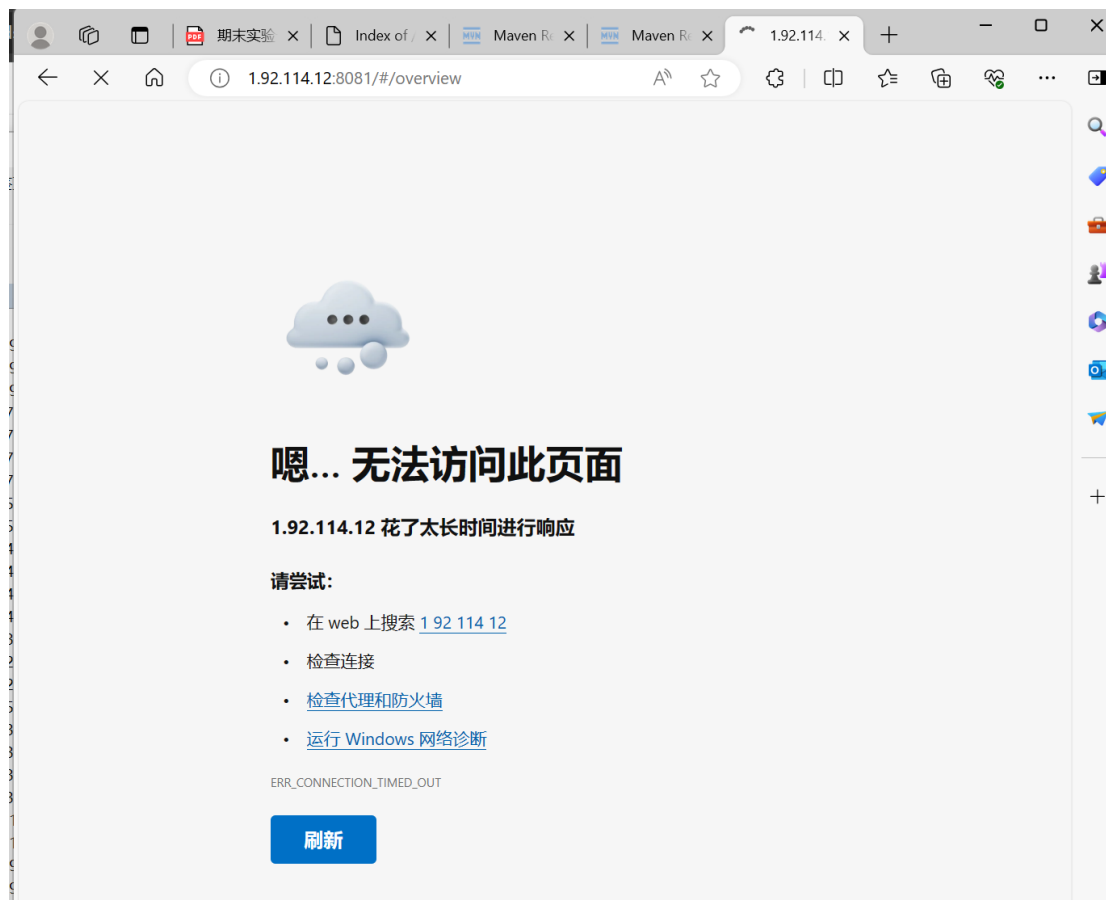
```
[root@cpy-2021211138 modules]# source /etc/profile
[root@cpy-2021211138 modules]# cd ~
[root@cpy-2021211138 ~]# start-cluster.sh
Starting cluster.
Starting standaloneSession daemon on host cpy-2021211138.
Starting taskexecutor daemon on host cpy-2021211138.
[root@cpy-2021211138 ~]# jps
2208 StandaloneSessionClusterEntrypoint
2529 Jps
2473 TaskManagerRunner
[root@cpy-2021211138 ~]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.30 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::f816:3eff:fe5:4d6b prefixlen 64 scopeid 0x20<link>
    ether fa:16:3e:f5:4d:6b txqueuelen 1000 (Ethernet)
    RX packets 259332 bytes 387920380 (369.9 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 156329 bytes 10485200 (9.9 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 107 bytes 86181 (84.1 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 107 bytes 86181 (84.1 KiB)
```

5. Web 界面访问

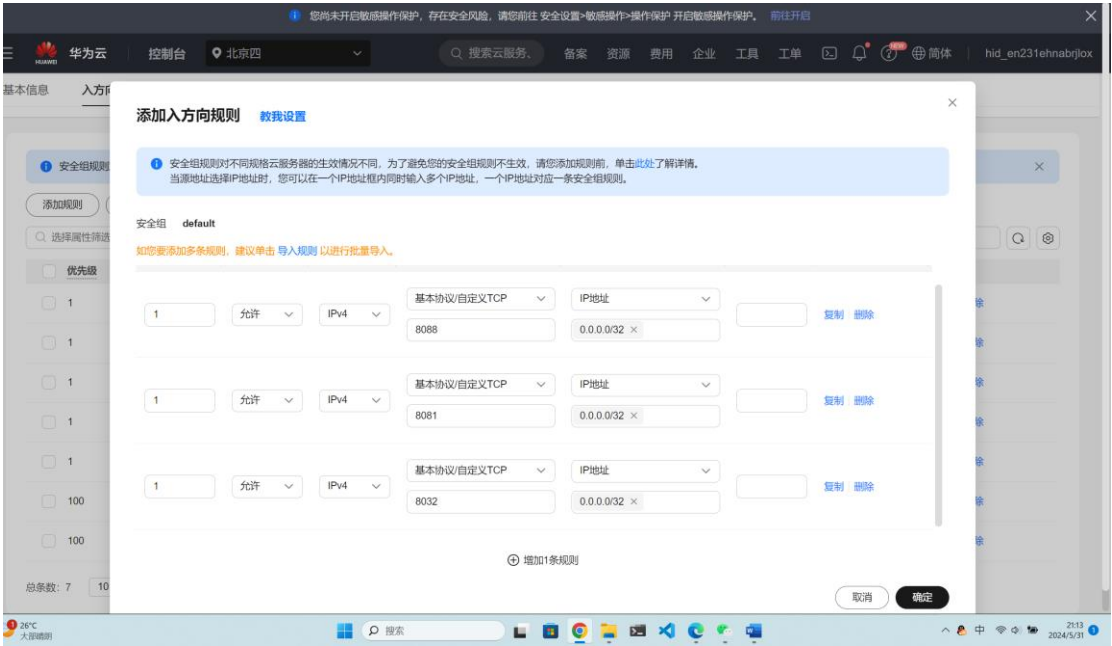
<http://1.92.114.12:8081/#/overview>

问题:



访问不了。

原因如下：

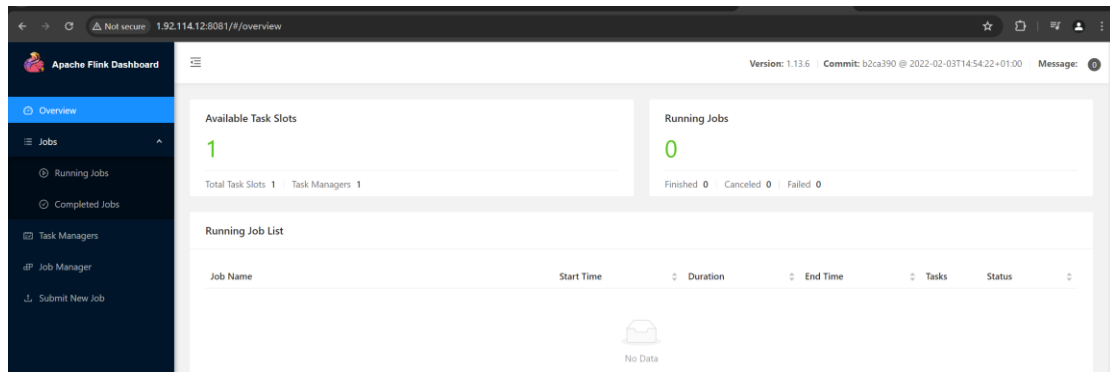


我的允许访问的 ip 地址网关设置为了/32, 但如果是要让所有 ip 都能访问到这台服务器的 8081 端口的话, 应该让 IP 地址和网关设置为 0.0.0.0/0, 才可以。

修改如下：



之后就能正常访问了



运行 Flink 自带的测试用例：

安装 nc 工具：

```
sudo yum -y install nc
```

遇到问题：

```
-bash: http://node1:8081/#/overview: No such file or directory
[root@cpy-2021211138 ~]# http://1.92.114.12:8081/#/overview
-bash: http://1.92.114.12:8081/#/overview: No such file or directory
[root@cpy-2021211138 ~]# sudo yum -y install nc
Loaded plugins: fastestmirror
```

```
Metadata Cache Created
[root@cpy-2021211138 ~]# sudo yum -y install nc
Loaded plugins: fastestmirror
[root@cpy-2021211138 ~]# sudo yum -y install nc
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
Resolving Dependencies
--> Running transaction check
---> Package nmap-ncat.aarch64 2:6.40-19.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved
```

原因是，从 word 复制的命令 `-y` 前面这个 `'-'` 是破折号，而命令行里需要用到的是连字符，所以要自己敲命令。

```
[root@cpy-2021211138 flink]# bin/flink run examples/batch/WordCount.jar
Executing WordCount example with default input data set.
Use --input to specify file input.
Printing result to stdout. Use --output to specify output path.
Job has been submitted with JobID 74a8986939c84f26d1eb419825689cf3
Program execution finished
Job with JobID 74a8986939c84f26d1eb419825689cf3 has finished.
Job Runtime: 505 ms
Accumulator Results:
- b8746b582ec419ff68478106de4bc4b1 (java.util.ArrayList) [170 elements]

(a,5)
(action,1)
(after,1)
(against,1)
(all,2)
(and,12)
(arms,1)
(arrows,1)
(awry,1)
(ay,1)
(bare,1)
(be,4)
(bear,3)
(bodkin,1)
(bourn,1)
(but,1)
(by,2)
(calamity,1)
```

打开 9000 端口

```
[root@cpy-2021211138 ~]# http://1.92.114.12:8081/#/overview
-bash: http://1.92.114.12:8081/#/overview: No such file or directory
[root@cpy-2021211138 ~]# nc -lk 9000
```

在另一个进程里，运行自带的程序

```
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

安装最新的 PowerShell，了解新功能和改进！https://aka.ms/PSWindows

PS C:\Users\20531> ssh root@1.92.114.12
root@1.92.114.12's password:
Last login: Sat Jun  1 09:46:54 2024 from 117.129.58.146

Welcome to Huawei Cloud Service

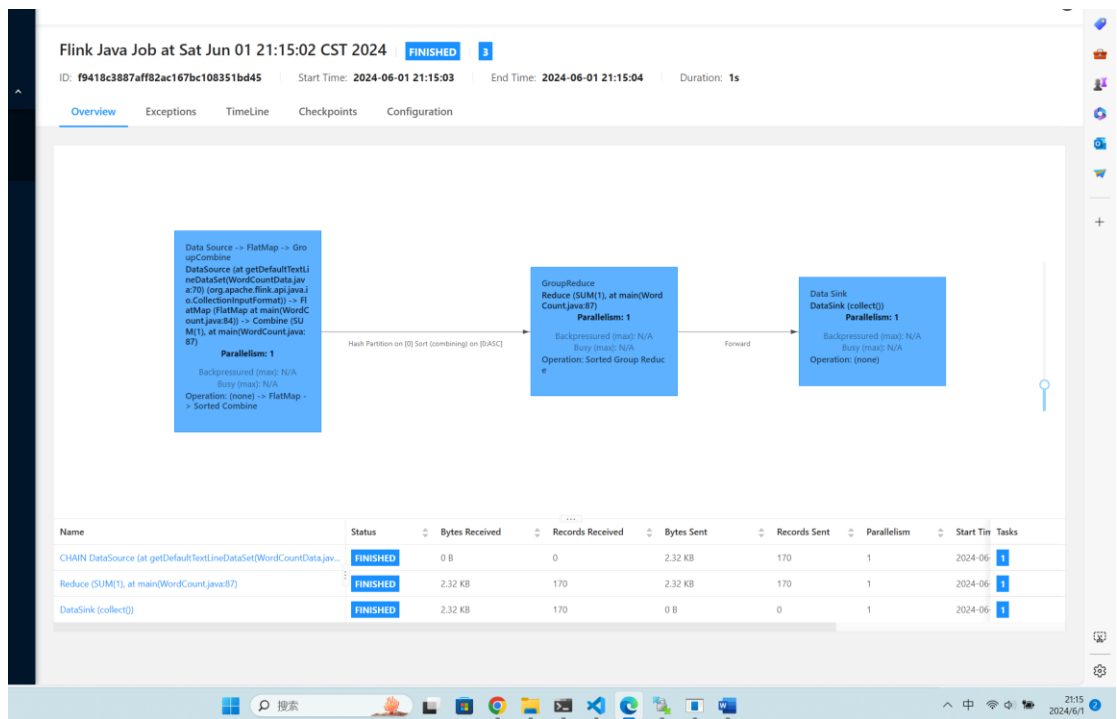
[root@cpy-2021211138 ~]# cd /home/modules/flink
[root@cpy-2021211138 flink]# bin/flink run examples/streaming/SocketWindowWordCount.jar --hostname localhost --port 9000
```

输入单次

```
[root@cpy-2021211138 ~]# http://1.92.114.12:8081/#/overview
-bash: http://1.92.114.12:8081/#/overview: No such file or directory
[root@cpy-2021211138 ~]# nc -lk 9000
cpy cpy cpy pcy pcy pyc pyc abc 12456 3454564 123456 123456 123456 cpy ypc cyp pyypypypyp 2021211138 2021211138 138 138 1
38
```

再打开一个终端，查看结果：

```
flink-root-standalone-session-0-cpy-2021211138.out
[root@cpy-2021211138 log]# tail -200f flink-root-taskexecutor-0-cpy-2021211138.out
cpy : 4
: 1
138 : 3
2021211138 : 2
pyypypy : 1
cyp : 1
ypc : 1
123456 : 3
3454564 : 1
12456 : 1
abc : 1
pyc : 2
pcy : 2
: 3
```

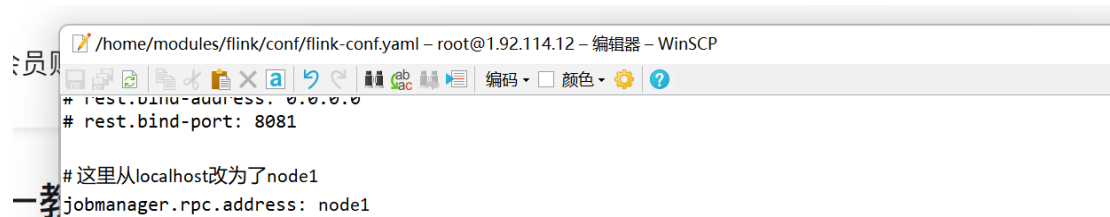


实验二

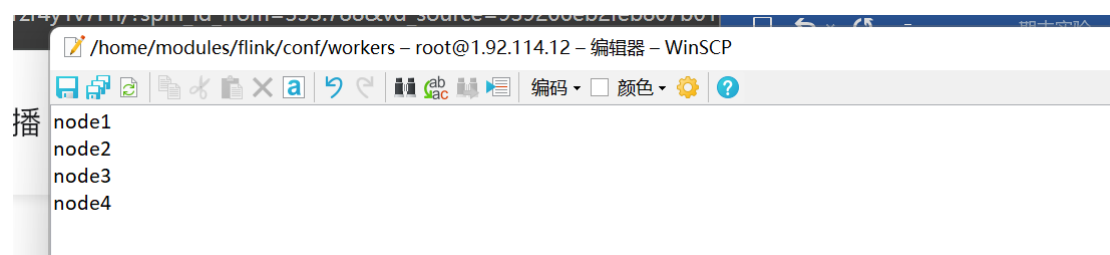
注释掉 host 文件的 127.0.0.1

只需要注释掉主服务器的 host

修改 flink-conf.yaml 文件，如下：



修改 workers 文件:



再将配置好的 flink 传给其他三个节点

```
cd /home/modules/  
scp -r flink root@node2:$PWD/  
scp -r flink root@node3:$PWD/  
scp -r flink root@node4:$PWD/
```

```
etc flink-shaded-hadoop-3-uber-3.11.1-2.1.0-927-9.0.jar include LICENSE.txt 501h  
[root@cpy-2021211138 modules]# scp -r flink root@node2:$PWD/  
flink-s3-fs-presto-1.13.6.jar 100% 32MB 120.0MB/s 00:00  
cloudpickle-1.2.2-src.zip 100% 23KB 15.8MB/s 00:00  
py4j-0.10.8.1-src.zip 100% 81KB 38.5MB/s 00:00  
pyflink.zip 100% 546KB 85.3MB/s 00:00  
flink-oss-fs-hadoop-1.13.6.jar 100% 16MB 122.2MB/s 00:00  
flink-cep_2.11-1.13.6.jar 100% 185KB 48.2MB/s 00:00  
flink-gelly-scala_2.11-1.13.6.jar 100% 743KB 90.3MB/s 00:00  
flink-shaded-zookeeper-3.5.6.jar 100% 7838KB 119.3MB/s 00:00  
flink-s3-fs-hadoop-1.13.6.jar 100% 19MB 112.8MB/s 00:00  
flink-python_2.11-1.13.6.jar 100% 37MB 115.7MB/s 00:00
```

启动:

```
README.txt 100% 1309  
[root@cpy-2021211138 modules]# start-cluster.sh  
Starting cluster.  
Starting standalone-session daemon on host cpy-2021211138.  
Starting taskexecutor daemon on host cpy-2021211138.  
Starting taskexecutor daemon on host cpy-2021211138.  
Starting taskexecutor daemon on host cpy-2021211138.  
Starting taskexecutor daemon on host cpy-2021211138.
```

主节点的 jps:


```
Starting taskexecutor daemon on host cpy-2021211138.
[root@cpy-2021211138 modules]# jps
12123 Jps
12013 TaskManagerRunner
11713 StandaloneSessionClusterEntrypoint
[root@cpy-2021211138 modules]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.30 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::f816:3eff:fe5:4d6b prefixlen 64 scopeid 0x20<link>
    ether fa:16:3e:f5:4d:6b txqueuelen 1000 (Ethernet)
    RX packets 69459 bytes 5297091 (5.0 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 59855 bytes 1077058398 (1.0 GiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

从节点的 jps:

node2:

```
[root@cpy-2021211138 ~]# jps
1968 Jps
1888 TaskManagerRunner
[root@cpy-2021211138 ~]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.213 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::f816:3eff:fe5:4d22 prefixlen 64 scopeid 0x20<link>
    ether fa:16:3e:f5:4d:22 txqueuelen 1000 (Ethernet)
    RX packets 259244 bytes 372410061 (355.1 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 22928 bytes 1798836 (1.7 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

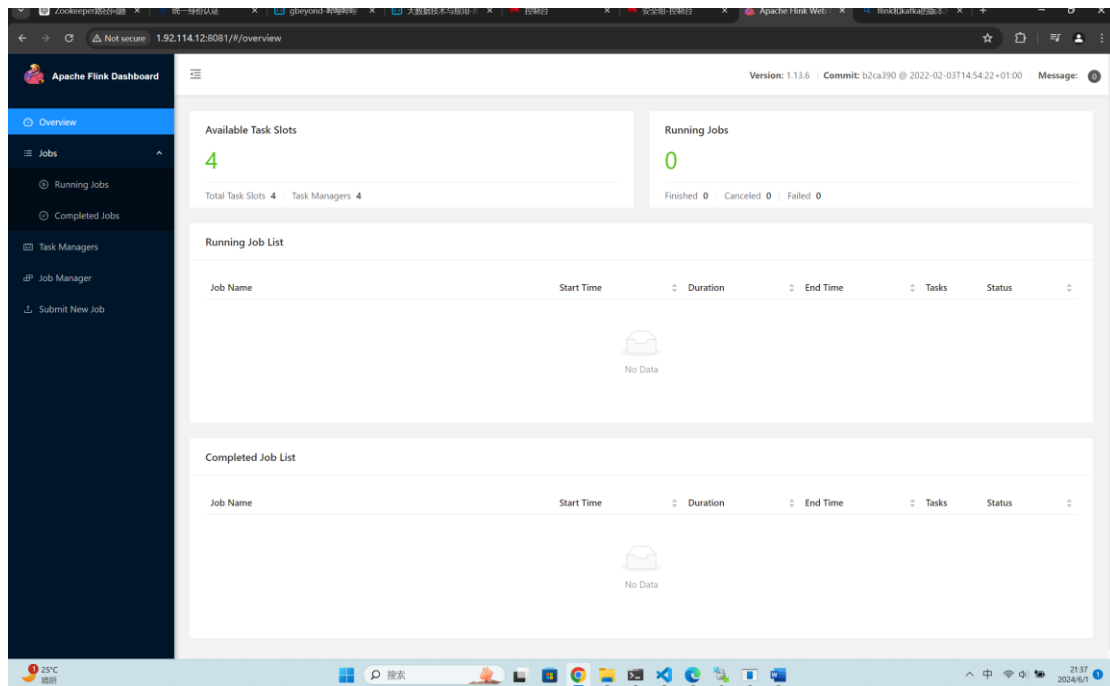
node3:

```
[root@cpy-2021211138 ~]# jps
1898 TaskManagerRunner
1978 Jps
[root@cpy-2021211138 ~]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.161 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::f816:3eff:fe5:4dee prefixlen 64 scopeid 0x20<link>
    ether fa:16:3e:f5:4d:ee txqueuelen 1000 (Ethernet)
    RX packets 256712 bytes 372242342 (354.9 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 20954 bytes 1702853 (1.6 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

node4:

```
[root@cpy-2021211138 ~]# jps
1906 TaskManagerRunner
1986 Jps
[root@cpy-2021211138 ~]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.135 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::f816:3eff:fe5:4dd4 prefixlen 64 scopeid 0x20<link>
    ether fa:16:3e:f5:4d:d4 txqueuelen 1000 (Ethernet)
    RX packets 255926 bytes 372193478 (354.9 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 20321 bytes 1668116 (1.5 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

web 管理页面：

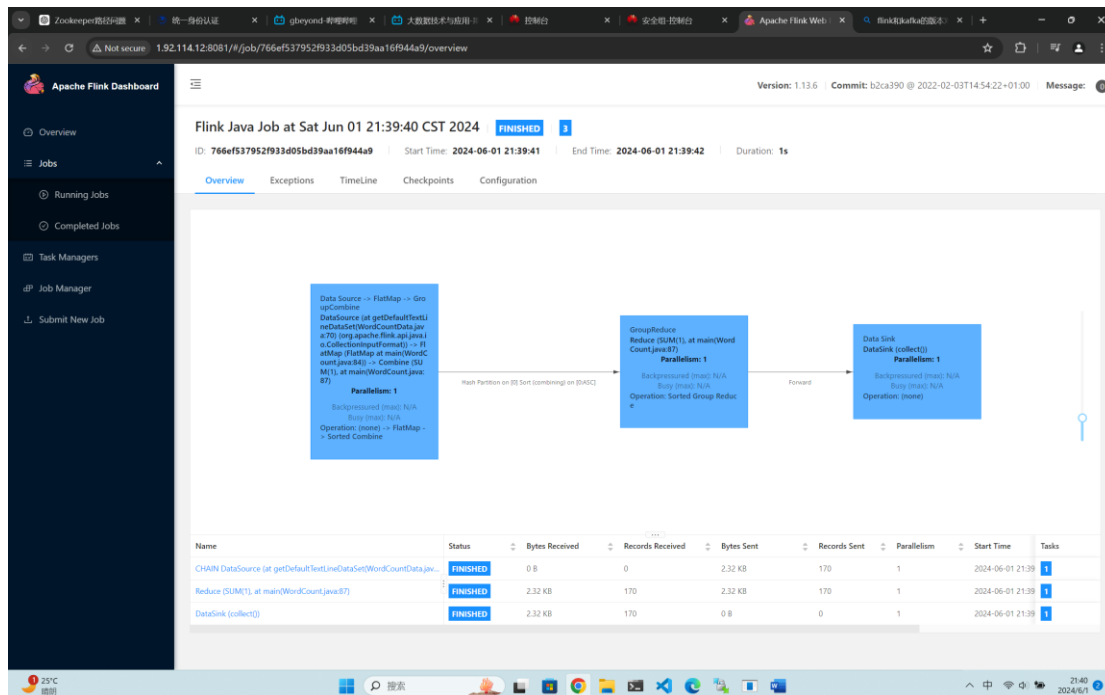


运行测试用例：

```
[root@cpy-2021211138 modules]# cd flink
[root@cpy-2021211138 flink]# bin/flink run examples/batch/WordCount.jar
Executing WordCount example with default input data set.
Use --input to specify file input.
Printing result to stdout. Use --output to specify output path.
Job has been submitted with JobID 766ef537952f933d05bd39aa16f944a9
Program execution finished
Job with JobID 766ef537952f933d05bd39aa16f944a9 has finished.
Job Runtime: 1072 ms
Accumulator Results:
- f96fa2439761acf45d1db4a102d1fe57 (java.util.ArrayList) [170 elements]

(a,5)
(action,1)
(after,1)
(against,1)
(all,2)
(and,12)
(arms,1)
(arrows,1)
(awry,1)
(ay,1)
```

查看 web 管理页面的 Completed job：



实验三、Flink on Yarn

首先停掉所有服务

```
(you,1)
[root@cpy-2021211138 flink]# stop-cluster.sh
Stopping taskexecutor daemon (pid: 12013) on host cpy-2021211138.
Stopping taskexecutor daemon (pid: 1888) on host cpy-2021211138.
Stopping taskexecutor daemon (pid: 1898) on host cpy-2021211138.
Stopping taskexecutor daemon (pid: 1906) on host cpy-2021211138.
Stopping standalone session daemon (pid: 11713) on host cpy-2021211138.
[root@cpy-2021211138 flink]# stop-all.sh
This script is deprecated. Instead use stop-dfs.sh and stop-yarn.sh
24/06/01 21:58:51 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform...
```

修改 hadoop 路径下的 yarn-site.xml

```
<property>
  <name>yarn.resourcemanager.am.max-attempts</name>
  <value>4</value>
  <description>The maximum number of application master execution
attempts.</description>
</property>
<property>
  <name>yarn.application.classpath</name>
  <value>/home/modules/hadoop-2.7.7/etc/hadoop,/home/modules/hadoop-
2.7.7/share/hadoop/common/*,/home/modules/hadoop-
2.7.7/share/hadoop/common/lib/*,/home/modules/hadoop-
2.7.7/hdfs/*,/home/modules/hadoop-
2.7.7/share/hadoop/hdfs/lib/*,/home/modules/hadoop-
2.7.7/share/hadoop/yarn/*,/home/modules/hadoop-
```

```
2.7.7/share/hadoop/yarn/lib/*,/home/modules/hadoop-
2.7.7/mapreduce/*,/home/modules/hadoop-
2.7.7/share/hadoop/mapreduce/lib/*</value>
</property>
```

```
</property>
<property>
  <name>yarn.nodemanager.pmem-check-enabled</name>
  <value>>false</value>
</property>
<property>
  <name>yarn.nodemanager.vmem-check-enabled</name>
  <value>>false</value>
</property>
<property>
  <name>yarn.resourcemanager.am.max-attempts</name>
  <value>4</value>
  <description>The maximum number of application master execution attempts.</description>
</property>
<property>
  <name>yarn.application.classpath</name>
  <value>/home/modules/hadoop-2.7.7/etc/hadoop,/home/modules/hadoop-2.7.7/share/hadoop/common/*,/home/modules/hadoop-2.7.7/share/hadoop/common/lib/*,/home/mod
</property>
</configuration>
```

将修改好后的 yarn-site.xml 送给其他节点：

```
cd /home/modules/hadoop-2.7.7/etc/hadoop/
scp yarn-site.xml root@node2:$PWD
scp yarn-site.xml root@node3:$PWD
scp yarn-site.xml root@node4:$PWD
```

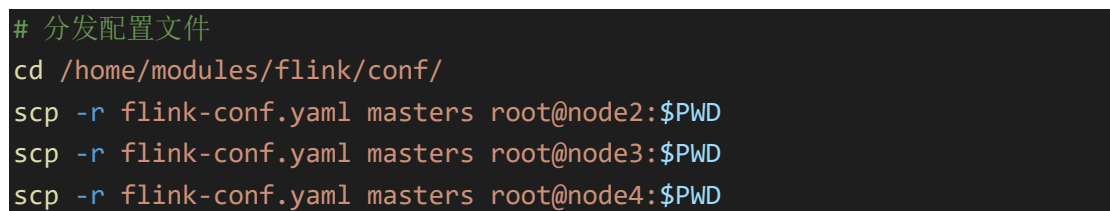
```
drwxr-xr-x - root supergroup 0 2024-05-25 20:23 /user
[root@cpy-2021211138 ~]# cd /home/modules/hadoop-2.7.7/etc/hadoop/
[root@cpy-2021211138 hadoop]# scp yarn-site.xml root@node2:$PWD
yarn-site.xml 100% 3251 7.7MB/s 00:00
[root@cpy-2021211138 hadoop]# scp yarn-site.xml root@node3:$PWD
yarn-site.xml 100% 3251 8.0MB/s 00:00
[root@cpy-2021211138 hadoop]# scp yarn-site.xml root@node4:$PWD
yarn-site.xml 100% 3251 8.2MB/s 00:00
[root@cpy-2021211138 hadoop]# |
```


再打开 hadoop 和 zookeeper 服务









```
[root@cpy-2021211138 flink]# cd ~
[root@cpy-2021211138 ~]# start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
24/06/01 22:09:17 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-j
va classes where applicable
Starting namenodes on [node1]
node1: starting namenode, logging to /home/modules/hadoop-2.7.7/logs/hadoop-root-namenode-cpy-2021211138.out
node3: starting datanode, logging to /home/modules/hadoop-2.7.7/logs/hadoop-root-datanode-cpy-2021211138.out
node2: starting datanode, logging to /home/modules/hadoop-2.7.7/logs/hadoop-root-datanode-cpy-2021211138.out
node4: starting datanode, logging to /home/modules/hadoop-2.7.7/logs/hadoop-root-datanode-cpy-2021211138.out
Starting secondary namenodes [node1]
node1: starting secondarynamenode, logging to /home/modules/hadoop-2.7.7/logs/hadoop-root-secondarynamenode-cpy-20212111
38.out
24/06/01 22:09:33 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-j
va classes where applicable
starting yarn daemons
starting resourcemanager, logging to /home/modules/hadoop-2.7.7/logs/yarn-root-resourcemanager-cpy-2021211138.out
node4: starting nodemanager, logging to /home/modules/hadoop-2.7.7/logs/yarn-root-nodemanager-cpy-2021211138.out
node2: starting nodemanager, logging to /home/modules/hadoop-2.7.7/logs/yarn-root-nodemanager-cpy-2021211138.out
node3: starting nodemanager, logging to /home/modules/hadoop-2.7.7/logs/yarn-root-nodemanager-cpy-2021211138.out
[root@cpy-2021211138 ~]# zkServer.sh start
ZooKeeper JMX enabled by default
Using config: /usr/local/zookeeper/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
[root@cpy-2021211138 ~]# |
```

修改 node1 的 flink-conf.yaml 文件

添加四行：



 /home/modules/flink/conf/masters – root@1.92.114.12 – 编辑

       编码 ▾ 

```
node1:8081
node2:8081
```

其他四个节点也是相同修改

将下面两个 jar 包放入 lib 中，并分发给其他四个节点：

```
[root@cpy-2021211138 lib]# scp commons-cli-1.4.jar root@node2:$PWD
s-cli-1.4.jar root@node3:$PWD
commons-cli-1.4.jar                                100% 53KB 23.5MB/s 00:00
[root@cpy-2021211138 lib]# scp commons-cli-1.4.jar root@node3:$PWD
commons-cli-1.4.jar                                100% 53KB 50.8MB/s 00:00
[root@cpy-2021211138 lib]# scp commons-cli-1.4.jar root@node4:$PWD
commons-cli-1.4.jar                                100% 53KB 52.1MB/s 00:00
[root@cpy-2021211138 lib]# scp flink-shaded-hadoop-3-uber-3.1.1.7.2.1.0-327-9.0.jar root@node2:$PWD
1.0-327-9.0.jar root@node3:$PWD
flink-shaded-hadoop-3-uber-3.1.1.7.2.1.0-327-9.0.jar          100% 56MB 124.6MB/s 00:00
[root@cpy-2021211138 lib]# scp flink-shaded-hadoop-3-uber-3.1.1.7.2.1.0-327-9.0.jar root@node3:$PWD
flink-shaded-hadoop-3-uber-3.1.1.7.2.1.0-327-9.0.jar          100% 56MB 122.6MB/s 00:00
[root@cpy-2021211138 lib]# scp flink-shaded-hadoop-3-uber-3.1.1.7.2.1.0-327-9.0.jar root@node4:$PWD
flink-shaded-hadoop-3-uber-3.1.1.7.2.1.0-327-9.0.jar          100% 56MB 125.6MB/s 00:00
```

在 hdfs 文件系统中添加路径：

```
Did you mean -mkdir? This command begins with a dash.
[root@cpy-2021211138 ~]# hadoop fs -mkdir -p /flink
24/06/01 22:38:05 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-ja
va classes where applicable
[root@cpy-2021211138 ~]# hadoop fs -ls /
24/06/01 22:38:16 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-ja
va classes where applicable
Found 8 items
drwxr-xr-x - root supergroup 0 2024-04-22 21:12 /experiment2
drwxr-xr-x - root supergroup 0 2024-06-01 22:38 /flink
drwxr-xr-x - root supergroup 0 2024-05-26 08:56 /hbase
drwxr-xr-x - root supergroup 0 2024-05-25 22:23 /spark-test
drwxr-xr-x - root supergroup 0 2024-05-25 22:20 /spark-test1
drwx----- - root supergroup 0 2024-05-25 16:19 /tmp
-rw-r--r-- 3 root supergroup 37 2024-03-22 22:08 /upload_2021211138.txt
drwxr-xr-x - root supergroup 0 2024-05-25 20:23 /user
```

各个节点都启动 zookeeper：

解除 hadoop 的安全模式：

hadoop dfsadmin -safemode leave

```
156.out
[root@cpy-2021211138 ~]# hadoopdfsadmin-safemode leave
-bash: hadoopdfsadmin-safemode: command not found
[root@cpy-2021211138 ~]# hadoop dfsadmin -safemode leave
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.

24/06/01 23:16:28 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... u
sing builtin-java classes where applicable
Safe mode is OFF
[root@cpy-2021211138 ~]# |
```

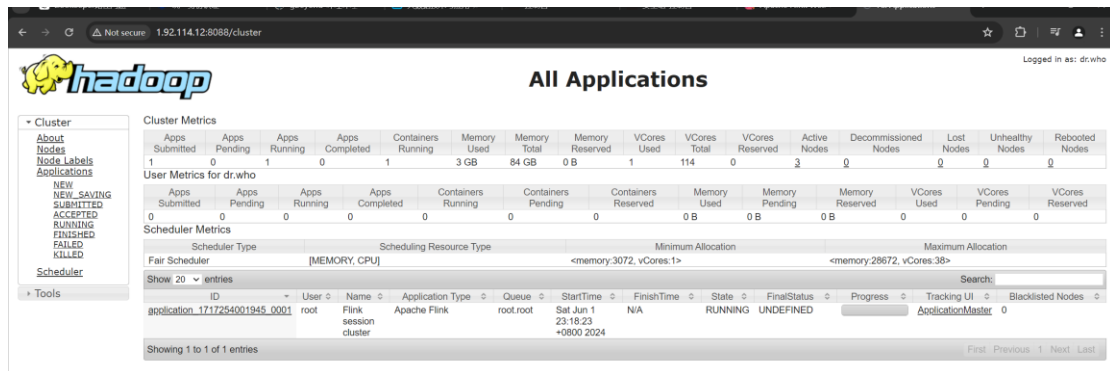
输入以下命令：

yarn-session.sh -n 2 -jm 1024 -tm 1024 -d

```
2024-06-01 23:18:07 INFO org.apache.flink.yarn.YarnClusterDescriptor: [] - The configured taskmanager memory is 1728 MB. Yarn will allocate 3072 MB to
an integer multiple of its minimum allocation memory (3072 MB, configured via 'yarn.scheduler.minimum-allocation-mb'). The extra 1344 MB may not be used by Flink.
2024-06-01 23:18:19,087 INFO org.apache.flink.yarn.YarnClusterDescriptor: [] - Cluster specification: ClusterSpecification{masterMemoryMB=3072, taskManagerM
8=1728, slotsPerTaskManager=1}
2024-06-01 23:18:23,465 INFO org.apache.flink.runtime.util.config.memory.ProcessMemoryUtils: [] - The derived from fraction jvm overhead memory (160.000mb (167772162 bytes))
is than its min value 192.000mb (201326592 bytes), min value will be used instead
2024-06-01 23:18:23,478 INFO org.apache.flink.yarn.YarnClusterDescriptor: [] - Submitting application master application_1717254001945_0001
2024-06-01 23:18:23,903 INFO org.apache.hadoop.yarn.client.api.impl.YarnClientImpl: [] - Submitted application application_1717254001945_0001
2024-06-01 23:18:23,822 INFO org.apache.flink.yarn.YarnClusterDescriptor: [] - Waiting for the cluster to be allocated
2024-06-01 23:18:33,903 INFO org.apache.flink.yarn.YarnClusterDescriptor: [] - YARN application has been deployed successfully.
2024-06-01 23:18:33,904 INFO org.apache.flink.yarn.YarnClusterDescriptor: [] - Found Web Interface node3:34447 of application 'application_1717254001945_0001'
2024-06-01 23:18:33,965 INFO org.apache.flink.shaded.curator4.org.apache.curator.utils.Compatibility: [] - Running in ZooKeeper 3.4.x compatibility mode
2024-06-01 23:18:33,966 INFO org.apache.flink.shaded.curator4.org.apache.curator.utils.Compatibility: [] - Using emulated InjectSessionExpiration
2024-06-01 23:18:33,995 INFO org.apache.flink.shaded.curator4.org.apache.curator.framework.imps.CuratorFrameworkImpl: [] - Starting
2024-06-01 23:18:34,017 WARN org.apache.flink.shaded.zookeeper3.org.apache.zookeeper.ClientCnxn: [] - SASL configuration failed: javax.security.auth.login.LoginException: No
configuration section named 'Client' was found in specified JAAS configuration file: '/tmp/jaas-1545829289241262353.conf'. Will continue connection to Zookeeper server witho
ut authentication, if Zookeeper server allows it.
2024-06-01 23:18:34,019 ERROR org.apache.flink.shaded.curator4.org.apache.curator.ConnectionState: [] - Authentication failed
2024-06-01 23:18:34,028 INFO org.apache.flink.shaded.curator4.org.apache.curator.framework.imps.CuratorFrameworkImpl: [] - Default schema
2024-06-01 23:18:34,066 INFO org.apache.flink.shaded.curator4.org.apache.curator.framework.state.ConnectionStateManager: [] - State change: CONNECTED
2024-06-01 23:18:34,330 INFO org.apache.flink.runtime.leaderretrieval.DefaultLeaderRetrievalService: [] - Starting DefaultLeaderRetrievalService with ZookeeperLeaderRetrieval
r{retrievalPath=/leader/rest_server.lock'}
JobManager Web Interface: http://node3:34447
```

之后卡在上面这个地方，并且不要关闭。

接下来访问 yarn 的 8088 管理界面 <http://1.92.114.12:8088/cluster>



再开一个终端连接主节点，进入 flink 安装目录，输入以下命令

```
bin/flink run examples/batch/WordCount.jar
```

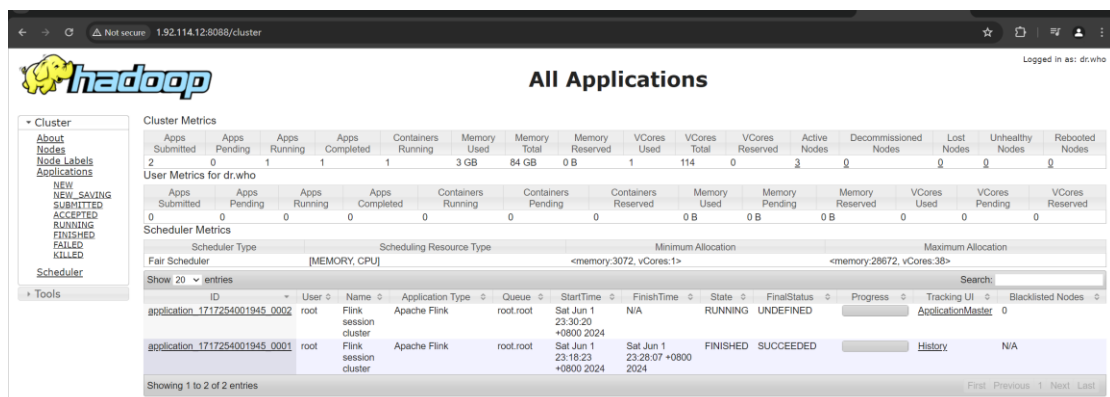
结果：

```
bin bash examples (10) LICENSE licenses log NOTICE opt plugins README.txt
[root@copy-202111138 flink]# bin/flink run examples/batch/WordCount.jar
2024-06-01 23:31:01,875 INFO org.apache.flink.yarn.cli.FlinkYarnSessionCli
2024-06-01 23:31:01,875 INFO org.apache.flink.yarn.cli.FlinkYarnSessionCli
Executing WordCount example with default input data set.
Use --input to specify file input.
Printing result to stdout. Use --output to specify output path.
2024-06-01 23:31:01,549 WARN org.apache.flink.yarn.configuration.YarnLogConfigUtil
ntains a LOG4J config file.If you want to use logback, then please delete or rename the log configuration file.
2024-06-01 23:31:01,644 INFO org.apache.hadoop.yarn.client.RMPProxy
[] - Found Yarn properties file under /tmp/.yarn-properties-root.
[] - Found Yarn properties file under /tmp/.yarn-properties-root.
2024-06-01 23:31:01,795 INFO org.apache.flink.yarn.YarnClusterDescriptor
[] - The configuration directory ('/home/modules/flink/conf') already co
[] - Connecting to ResourceManager at node1/192.168.0.30:8032
[] - No path for the flink jar passed. Using the location of class org.a
pache.flink.yarn.YarnClusterDescriptor to locate the jar
2024-06-01 23:31:01,864 INFO org.apache.flink.yarn.YarnClusterDescriptor
[] - Found Web Interface node2:44611 of application 'application_1717254
001945_0002'.
Job has been submitted with JobID 8049d44977529ffd49f5a85774aceecf
Program execution finished
Job with JobID 8049d44977529ffd49f5a85774aceecf has finished.
Job Runtime: 12869 ms
Accumulator Results:
- 70db26b72f0673f0edf77d3edaf7dbf9 (java.util.ArrayList) [170 elements]

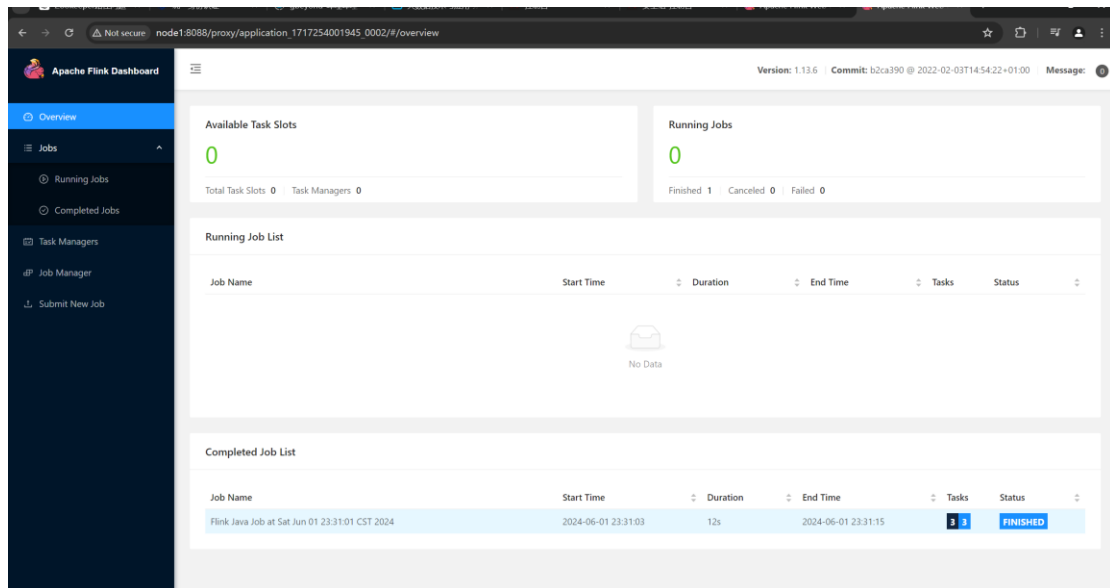
(a,5)
(action,1)
(after,1)
(against,1)
(all,2)
(and,12)
(are,1)
(arrows,1)

(turn,1)
(under,1)
(undiscover,1)
(unworthy,1)
(us,3)
(we,4)
(weary,1)
(what,1)
(when,2)
(whether,1)
(whips,1)
(who,2)
(whose,1)
(will,1)
(wish,1)
(with,3)
(would,2)
(wrong,1)
(you,1)
```

在 yarn 中有对应的任务：



点击 ApplicationMaster, 可以看到 Job Name



实验四、

下载 kafka_1.12.1.0.2.tgz 到/home/modules 目录下, 并解压缩

换名成 kafka

然后将 kafka 分发给其他三个从节点

```
scp -r kafka root@node2:$PWD
scp -r kafka root@node3:$PWD
scp -r kafka root@node4:$PWD
```

更改四个节点的/etc/profile 环境路径:

```
# Kafka
export KAFKA_HOME=/home/modules/kafka
export PATH=$KAFKA_HOME/bin:$PATH
```

再重新刷新一下

```
source /etc/profile
```

进入 kafka 安装包的 config 目录, 在 server.properties 添加内容:


```
broker.id=1
host.name=node1
zookeeper.connect=node1:2181,node2:2181,node3:2181,node4:2181
```

```
# The id of the broker. This must be set to a unique integer for each broker.
broker.id=1
host.name=node1
zookeeper.connect=node1:2181,node2:2181,node3:2181,node4:2181
```

```
# The id of the broker. This must be set to a unique integer for each broker.
broker.id=2
host.name=node2
zookeeper.connect=node1:2181,node2:2181,node3:2181,node4:2181
```

```
# The id of the broker. This must be set to a unique integer for each broker.
broker.id=3
host.name=node3
zookeeper.connect=node1:2181,node2:2181,node3:2181,node4:2181
```

```
broker.id=4
host.name=node4
zookeeper.connect=node1:2181,node2:2181,node3:2181,node4:2181
```

```
##### Socket Server Settings #####
```

配置完成，验证是否安装成功，前提要先启动 zookeeper

然后在各个节点分别启动 kafka

先进入 kafka 目录

然后输入以下命令：

```
./bin/kafka-server-start.sh ./config/server.properties
```

再启动一个终端，查看 jps

这是主节点的 jps：

```

PS C:\Users\20531> ssh root@1.92.114.12
root@1.92.114.12's password:
Last login: Sun Jun  2 08:59:35 2024 from 117.129.58.146

Welcome to Huawei Cloud Service

[root@cpy-2021211138 ~]# jps
18164 QuorumPeerMain
18585 Jps
18242 Kafka
[root@cpy-2021211138 ~]# |

```

node2 的 jps:

```

PS C:\Users\20531> ssh root@120.46.149.118
root@120.46.149.118's password:
Last login: Sun Jun  2 09:37:37 2024 from 117.129.58.146

Welcome to Huawei Cloud Service

[root@cpy-2021211138 ~]# jps
4742 Kafka
4633 QuorumPeerMain
5099 Jps

```

node3 的 jps:

```

PS C:\Users\20531> ssh root@1.92.86.3
root@1.92.86.3's password:
Last login: Sun Jun  2 08:59:04 2024 from 117.129.58.146

Welcome to Huawei Cloud Service

[root@cpy-2021211138 ~]# jps
4577 QuorumPeerMain
4979 Jps
4641 Kafka

```

node4 的 jps:

```

PS C:\Users\20531> ssh root@120.46.87.42
root@120.46.87.42's password:
Last login: Sun Jun  2 08:59:16 2024 from 1.92.114.12

Welcome to Huawei Cloud Service

[root@cpy-2021211138 ~]# jps
4454 QuorumPeerMain
4511 Kafka
4849 Jps

```

检查完毕后, 关闭各个节点的 kafka 进程,

编写代码: pom.xml 文件:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>org.example</groupId>
  <artifactId>MyWordCount</artifactId>
  <version>1.0-SNAPSHOT</version>

  <properties>

<maven.compiler.source>8</maven.compiler.source>

<maven.compiler.target>8</maven.compiler.target>
    <project.build.sourceEncoding>UTF-
8</project.build.sourceEncoding>
    <hadoop.version>2.7.7</hadoop.version>
  </properties>

  <dependencies>
    <dependency>
      <groupId>org.javatuples</groupId>
      <artifactId>javatuples</artifactId>
      <version>1.2</version>
    </dependency>

    <dependency>
      <groupId>log4j</groupId>
      <artifactId>log4j</artifactId>
      <version>1.2.17</version>
      <scope>provided</scope>
    </dependency>
    <dependency>
      <groupId>org.slf4j</groupId>
      <artifactId>slf4j-api</artifactId>
      <version>1.7.30</version>
      <scope>runtime</scope>
    </dependency>
    <dependency>
      <groupId>org.apache.flink</groupId>
```

```

        <artifactId>flink-java</artifactId>
        <version>1.13.6</version>
    </dependency>
    <dependency>
        <groupId>org.apache.kafka</groupId>
        <artifactId>kafka_2.12</artifactId>
        <version>1.0.2</version>
    </dependency>
    <dependency>
        <groupId>org.apache.flink</groupId>
        <artifactId>flink-clients_2.11</artifactId>
        <version>1.13.6</version>
    </dependency>
    <dependency>
        <groupId>org.apache.kafka</groupId>
        <artifactId>kafka-clients</artifactId>
        <version>1.0.2</version>
    </dependency>
    <dependency>
        <groupId>org.apache.flink</groupId>
        <artifactId>flink-streaming-
scala_2.12</artifactId>
        <version>1.13.6</version>
    </dependency>
    <dependency>
        <groupId>org.apache.flink</groupId>
        <artifactId>flink-connector-
kafka_2.11</artifactId>
        <version>1.13.6</version>
        <scope>provided</scope>
    </dependency>
    <dependency>
        <groupId>org.apache.flink</groupId>
        <artifactId>flink-connector-
base</artifactId>
        <version>1.13.6</version>
    </dependency>
    <dependency>
        <groupId>org.apache.hadoop</groupId>
        <artifactId>hadoop-client</artifactId>
        <version>${hadoop.version}</version>
        <scope>provided</scope>
    </dependency>
    <dependency>

```

```

        <groupId>org.apache.hadoop</groupId>
        <artifactId>hadoop-common</artifactId>
        <version>${hadoop.version}</version>
    </dependency>
    <dependency>
        <groupId>org.apache.hadoop</groupId>
        <artifactId>hadoop-hdfs</artifactId>
        <version>${hadoop.version}</version>
    </dependency>
    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>RELEASE</version>
        <scope>test</scope>
    </dependency>
</dependencies>

</project>

```

Flink_Kafka.java 文件:

```

import
org.apache.flink.api.common.functions.FlatMapFunction
;
import
org.apache.flink.api.common.serialization.SimpleStringSchema;
import
org.apache.flink.streaming.api.datastream.DataStream;
import
org.apache.flink.streaming.api.environment.StreamExecutionEnvironment;
import
org.apache.flink.streaming.connectors.kafka.FlinkKafkaConsumer;
import org.apache.flink.util.Collector;
import org.apache.flink.api.java.tuple.Tuple2;

import java.util.Properties;

/**
 * Flink_Kafka - A simple Flink application that
 * reads messages from a Kafka topic,
 * splits each message into words, and counts the

```

```
occurrences of each word.
 *
 * @author 陈朴炎
 * @version 1.0
 */
public class Flink_Kafka {
    public static void main(String[] args) throws
Exception {

        // 获取 Flink 运行环境

        StreamExecutionEnvironment env =
StreamExecutionEnvironment.getExecutionEnvironment();

        // 配置 Kafka 连接属性

        Properties properties = new Properties();
        properties.setProperty("bootstrap.servers",
"node1:9092"); // Kafka 服务器地址
        properties.setProperty("zookeeper.connect",
"node1:2181"); // Zookeeper 地址（可选，通常用于旧版
Kafka）

        properties.setProperty("group.id", "1"); // 消
费者组 ID

        // 创建 FlinkKafkaConsumer 来读取 Kafka 中的数据

        FlinkKafkaConsumer<String> myConsumer = new
FlinkKafkaConsumer<>("test", new
SimpleStringSchema(), properties);

        myConsumer.setStartFromGroupOffsets(); // 从消
费者组的偏移量开始消费

        DataStream<String> dataStream =
env.addSource(myConsumer); // 将 Kafka 消费者添加为数据源
```

```

        // 处理数据流：将每行文本拆分成单词，并计算每个单词的出
        现次数

        DataStream<Tuple2<String, Integer>> result =
        dataStream

            .flatMap(new MyFlatMapper()) // 将每行文
        本拆分成单词

            .keyBy(0) // 按照单词分组

            .sum(1); // 计算每个单词的出现次数


        // 打印结果到控制台，设置并行度为 1
        result.print().setParallelism(1);


        // 执行 Flink 作业
        env.execute();
    }

    /**
     * MyFlatMapper - A FlatMapFunction that splits
     lines of text into words.
     * 每行文本拆分成单词，并将每个单词作为 Tuple2<单词, 1>输
     出
     */
    public static class MyFlatMapper implements
    FlatMapFunction<String, Tuple2<String, Integer>> {
        @Override
        public void flatMap(String s,
        Collector<Tuple2<String, Integer>> collector) throws
        Exception {

            // 将行文本拆分成单词
            String[] words = s.split(" ");

            // 输出每个单词，附带初始计数 1

```

```

        for (String word : words) {
            collector.collect(new Tuple2<>(word,
1)) );
        }
    }
}
}
}

```

打包、上传 jar 包

停掉 zookeeper

开启 flink on yarn 集群

打开 hadoop、关闭 hadoop 的安全模式：

cd /home/modules/flink, 输入：

bin/yarn-session.sh -n 2 -jm 1024 -tm 1024 -d

```

ction to Zookeeper server without SASL authentication, if Zookeeper server al
2024-06-02 11:05:05,872 ERROR org.apache.flink.shaded.curator4.org.apache.cura
entification failed
2024-06-02 11:05:05,890 INFO  org.apache.flink.shaded.curator4.org.apache.cura
nStateManager [] - State change: CONNECTED
2024-06-02 11:05:06,152 INFO  org.apache.flink.runtime.leaderretrieval.Default
tarting DefaultLeaderRetrievalService with ZookeeperLeaderRetrievalDrive
r_lock'}.
JobManager Web Interface: http://node3:35679
)

```

看到卡住，说明成功

再开一个终端，进入 kafka 的 config 目录下启动 kafka：

```

[root@cpy-2021211138 kafka]# cd config
[root@cpy-2021211138 config]# kafka-server-start.sh -daemon server.properties
[root@cpy-2021211138 config]# jps
21435 FlinkYarnSessionCli
19489 NameNode
19891 ResourceManager
21830 Kafka
21021 QuorumPeerMain
21866 Jps
19706 SecondaryNameNode
[root@cpy-2021211138 config]#

```

停止 node1 的 zookeeper，然后启动 kafka 自带的 zookeeper

./bin/zookeeper-server-start.sh config/zookeeper.properties

进入到 kafka 的安装目录，输入下列指令：


```
./bin/kafka-topics.sh --create --zookeeper node1:2181 --replication-factor 1 --partitions 1 --topic test

kafka-console-producer.sh --broker-list node1:9092 --topic test
```

由于我输入的是 --zookeeper 选项，所以没有指导书中的那些提示信息

```
[root@cpy-2021211138 kafka]# ./bin/kafka-topics.sh --create --zookeeper node1:2181 --replication-factor 1 -
-partitions 1 --topic wordsendertest
Created topic "wordsendertest".
[root@cpy-2021211138 kafka]# ./bin/kafka-topics.sh --create --zookeeper node1:2181 --replication-factor 1 -
-partitions 1 --topic test
Created topic "test".
[root@cpy-2021211138 kafka]# kafka-console-producer.sh --broker-list node1:9092 --topic test
>
>
```

再启动一个终端，输入 `flink run -c Flink_Kafka WordCount.jar`

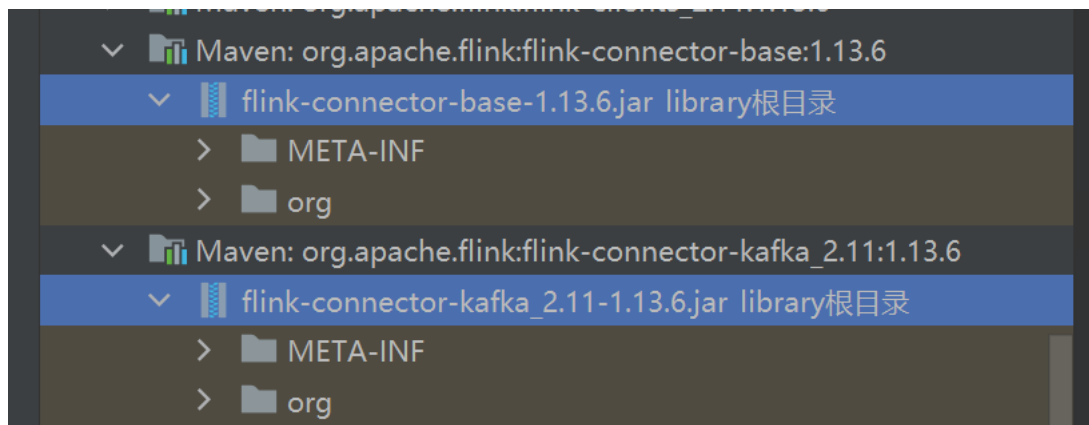
```
[root@cpy-2021211138-0001 ~]# flink run -c Flink_Kafka WordCount.jar
2024-06-02 11:39:25,597 INFO org.apache.flink.yarn.cli.FlinkYarnSessionCli
2024-06-02 11:39:25,597 INFO org.apache.flink.yarn.cli.FlinkYarnSessionCli
2024-06-02 11:39:25,597 INFO org.apache.flink.yarn.cli.FlinkYarnSessionCli
2024-06-02 11:39:25,609 INFO org.apache.flink.yarn.cli.FlinkYarnSessionCli
YARN properties set default parallelism to 2
2024-06-02 11:39:25,123 INFO org.apache.hadoop.yarn.client.RMProxy
2024-06-02 11:39:25,228 INFO org.apache.flink.yarn.cli.FlinkYarnSessionCli
arnClusterDescriptor to locate the jar
2024-06-02 11:39:25,228 INFO org.apache.flink.yarn.cli.FlinkYarnSessionCli
arnClusterDescriptor to locate the jar
2024-06-02 11:39:25,281 INFO org.apache.flink.yarn.AbstractYarnClusterDescriptor
plied application id 'application_1717389318704_0001'
Starting execution of program
- Found Yarn properties file under /tmp/yarn-properties-root.
- Found Yarn properties file under /tmp/yarn-properties-root.
- YARN properties set default parallelism to 2
- YARN properties set default parallelism to 2
- Connecting to ResourceManager at cpy-2021211138-0003/192.168.0.128:8032
- No path for the flink jar passed. Using the location of class org.apache.flink.yarn.
- No path for the flink jar passed. Using the location of class org.apache.flink.yarn.
- Found application JobManager host name 'cpy-2021211138-0003' and port '44047' from s
```

出现报错：

```
[root@cpy-2021211138 ~]# flink run -c Flink_Kafka WordCount.jar
2024-06-02 11:39:25,308 INFO org.apache.flink.yarn.cli.FlinkYarnSessionCli [] - Found Yarn
properties file under /tmp/.yarn-properties-root.
2024-06-02 11:39:25,308 INFO org.apache.flink.yarn.cli.FlinkYarnSessionCli [] - Found Yarn
properties file under /tmp/.yarn-properties-root.
java.lang.NoClassDefFoundError: org.apache.flink.streaming.connectors.kafka.FlinkKafkaConsumer
    at Flink_Kafka.main(Flink_Kafka.java:31)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:498)
    at org.apache.flink.client.program.PackagedProgram.callMainMethod(PackagedProgram.java:355)
    at org.apache.flink.client.program.PackagedProgram.invokeInteractiveModeForExecution(PackagedProgra
m.java:222)
    at org.apache.flink.client.ClientUtils.executeProgram(ClientUtils.java:114)
    at org.apache.flink.client.cli.CliFrontend.executeProgram(CliFrontend.java:812)
    at org.apache.flink.client.cli.CliFrontend.run(CliFrontend.java:246)
    at org.apache.flink.client.cli.CliFrontend.parseAndRun(CliFrontend.java:1054)
    at org.apache.flink.client.cli.CliFrontend.lambda$main$10(CliFrontend.java:1132)
```

解决：

在 maven 工程中的外部库中找到 `FlinkKafkaConsumer` 类和 `FlinkKafkaConsumerBase` 类，并将这两个 jar 包上传到服务器的 flink 下的 lib 中



上传:

/home/modules/flink/lib/		
名字	大小	已改变
..		2024/6/1 22:40:55
commons-cli-1.4.jar	53 KB	2024/6/1 9:44:43
flink-connector-base-1.13.6.jar	75 KB	2024/6/2 10:03:21
flink-connector-kafka_2.11-1.13.6.jar	351 KB	2024/6/2 10:03:20
flink-csv-1.13.6.jar	91 KB	2022/2/4 17:11:26
flink-dist_2.11-1.13.6.jar	112,721 ...	2022/2/4 17:15:16
flink-json-1.13.6.jar	145 KB	2022/2/4 17:11:06
flink-shaded-hadoop-3-uber-3.1.1.7.2.1.0-327-9.0.jar	57,004 KB	2024/6/1 9:44:54
flink-shaded-zookeeper-3.4.14.jar	7,530 KB	2021/5/7 15:53:16
flink-table_2.11-1.13.6.jar	35,601 KB	2022/2/4 17:14:12
flink-table-blink_2.11-1.13.6.jar	40,115 KB	2022/2/4 17:14:21
log4j-1.2-api-2.17.1.jar	204 KB	2022/1/13 19:06:26
log4j-api-2.17.1.jar	295 KB	2022/1/7 18:07:50
log4j-core-2.17.1.jar	1,749 KB	2022/1/7 18:07:50
log4j-slf4j-impl-2.17.1.jar	24 KB	2022/1/7 18:07:50

重新运行

```
mysql80-community-release-el7-3.noarch.rpm wordCount.jar
[root@cpy-2021211138 ~]# flink run -c Flink_Kafka WordCount.jar
2024-06-02 11:51:52,695 INFO org.apache.flink.yarn.cli.FlinkYarnSessionCli [] - Found Yarn
properties file under /tmp/.yarn-properties-root.
2024-06-02 11:51:52,695 INFO org.apache.flink.yarn.cli.FlinkYarnSessionCli [] - Found Yarn
properties file under /tmp/.yarn-properties-root.
java.lang.NoClassDefFoundError: org.apache.kafka.common.serialization.ByteArrayDeserializer
    at org.apache.flink.streaming.connectors.kafka.FlinkKafkaConsumer.setDeserializer(FlinkKafkaConsumer.java:322)
    at org.apache.flink.streaming.connectors.kafka.FlinkKafkaConsumer.<init>(FlinkKafkaConsumer.java:223)
```

出现上述错误,

再次找到相关的 jar 包, 传给 kafka 的 lib 下:

但是 kafka 的 lib 下已经有这个包了

但是 flink 下面没有这个包，于是把这个包放到 flink 的 lib 下面。

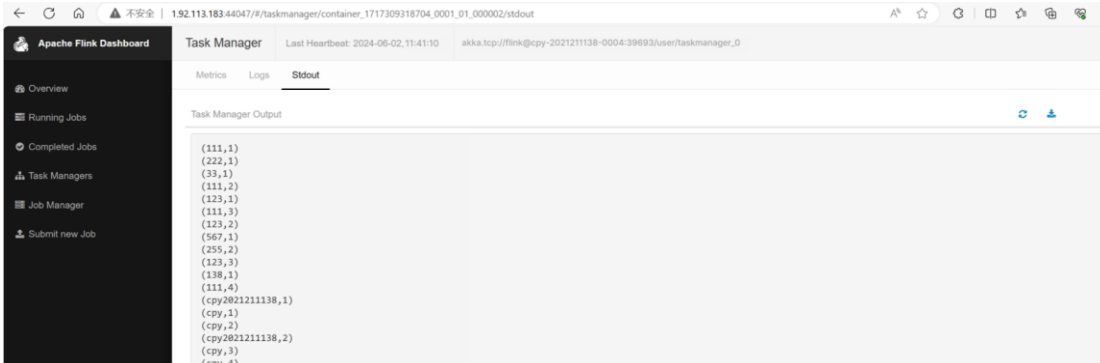
重新运行，没有这个报错了。

但是有了新的报错：

```
... 24 more
[root@cpy-2021211138 ~]# flink run -c Flink_Kafka WordCount.jar
2024-06-02 15:51:30,582 INFO  org.apache.flink.yarn.cli.FlinkYarnSessionCli      [] - Found Yarn
properties file under /tmp/.yarn-properties-root.
2024-06-02 15:51:30,582 INFO  org.apache.flink.yarn.cli.FlinkYarnSessionCli      [] - Found Yarn
properties file under /tmp/.yarn-properties-root.
2024-06-02 15:51:31,473 WARN  org.apache.flink.yarn.configuration.YarnLogConfigUtil      [] - The configura
tion directory ('/home/modules/flink/conf') already contains a LOG4J config file.If you want to use logba
ck, then please delete or rename the log configuration file.
2024-06-02 15:51:31,577 INFO  org.apache.hadoop.yarn.client.RMPProxy      [] - Connecting
to ResourceManager at node1/192.168.0.30:8032
2024-06-02 15:51:31,761 INFO  org.apache.flink.yarn.YarnClusterDescriptor      [] - No path for
the flink jar passed. Using the location of class org.apache.flink.yarn.YarnClusterDescriptor to locate th
e jar
2024-06-02 15:51:31,829 INFO  org.apache.flink.yarn.YarnClusterDescriptor      [] - Found Web I
nterface node3:43311 of application 'application_1717297085377_0005'.
Job has been submitted with JobID fa45c3617f840169c95e34d415725525

-----
The program finished with the following exception:

org.apache.flink.client.program.ProgramInvocationException: The main method caused an error: org.apache.fli
nk.client.program.ProgramInvocationException: Job failed (JobID: fa45c3617f840169c95e34d415725525)
    at org.apache.flink.client.program.PackagedProgram.callMainMethod(PackagedProgram.java:372)
    at org.apache.flink.client.program.PackagedProgram.invokeInteractiveModeForExecution(PackagedProgram.java:222)
    at org.apache.flink.client.ClientUtils.executeProgram(ClientUtils.java:114)
    at org.apache.flink.client.cli.CliFrontend.executeProgram(CliFrontend.java:812)
    at org.apache.flink.client.cli.CliFrontend.run(CliFrontend.java:246)
    at org.apache.flink.client.cli.CliFrontend.parseAndRun(CliFrontend.java:1054)
```



```
[root@cpy-2021211138      flink]#      bin/flink      run
example/batch/WordCount.jar      -
inputhdfs://node1:8020/flink_input      -output
hdfs://node1:8020/flink_coutput/wordcount-result.txt
```