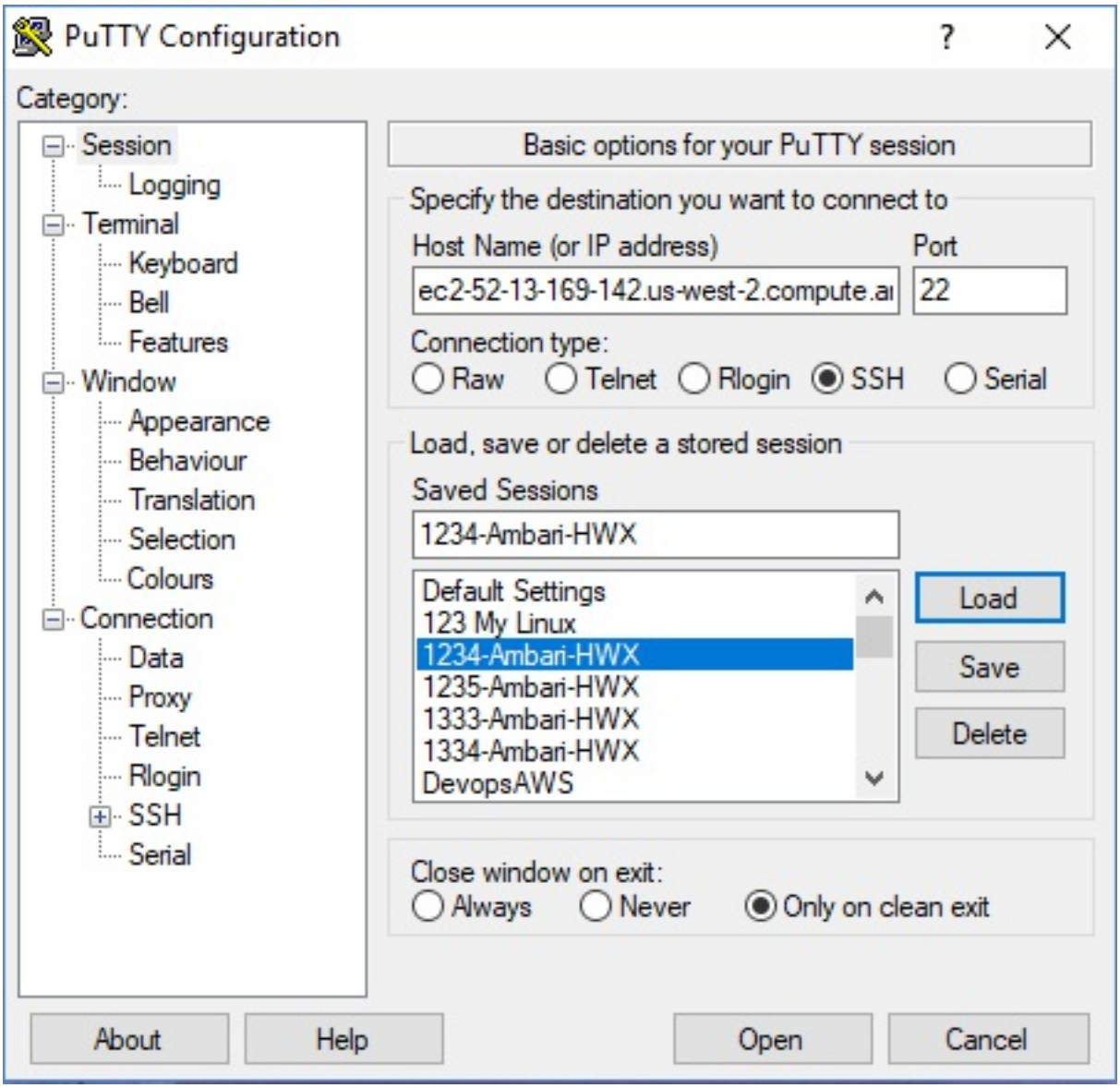
Set Up Your Terminals

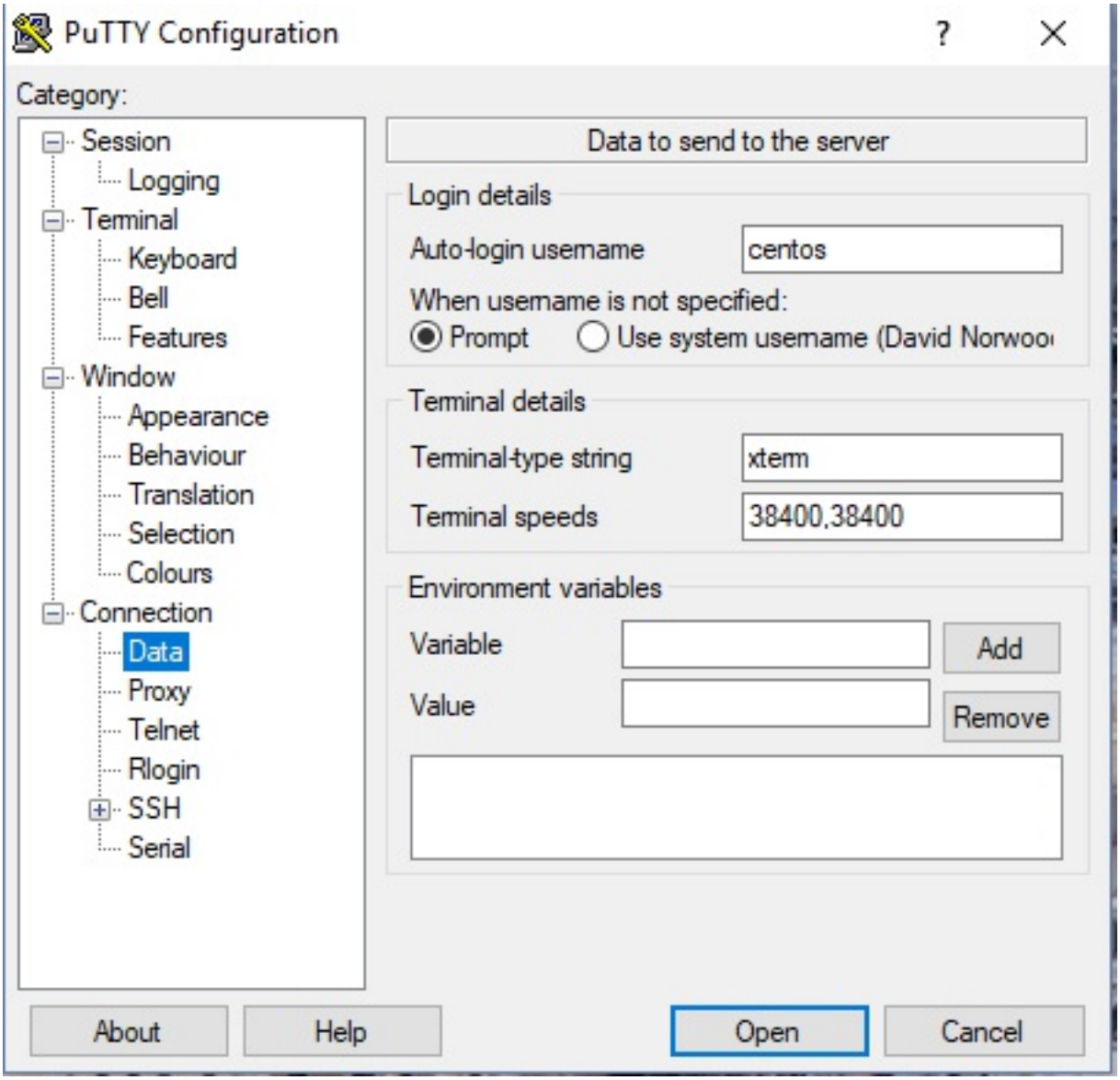
You will receive 1 to X AWS terminals:

DEV-343-del831-DN-Cognizant-120	AmbariNode	52.13.169.142	ec2-52-13-169-142.us-west-2.compute.amazonaws.com	172.30.10.248	ip-1
DEV-343-del831-DN-Cognizant-120	AdditionalNodes	54.190.61.151	ec2-54-190-61-151.us-west-2.compute.amazonaws.com	172.30.9.71	ip-1
DEV-343-del831-DN-Cognizant-120	AdditionalNodes	35.164.184.26	ec2-35-164-184-26.us-west-2.compute.amazonaws.com	172.30.15.197	ip-1
DEV-343-del831-DN-Cognizant-120	AdditionalNodes	54.202.147.67	ec2-54-202-147-67.us-west-2.compute.amazonaws.com	172.30.13.149	ip-1

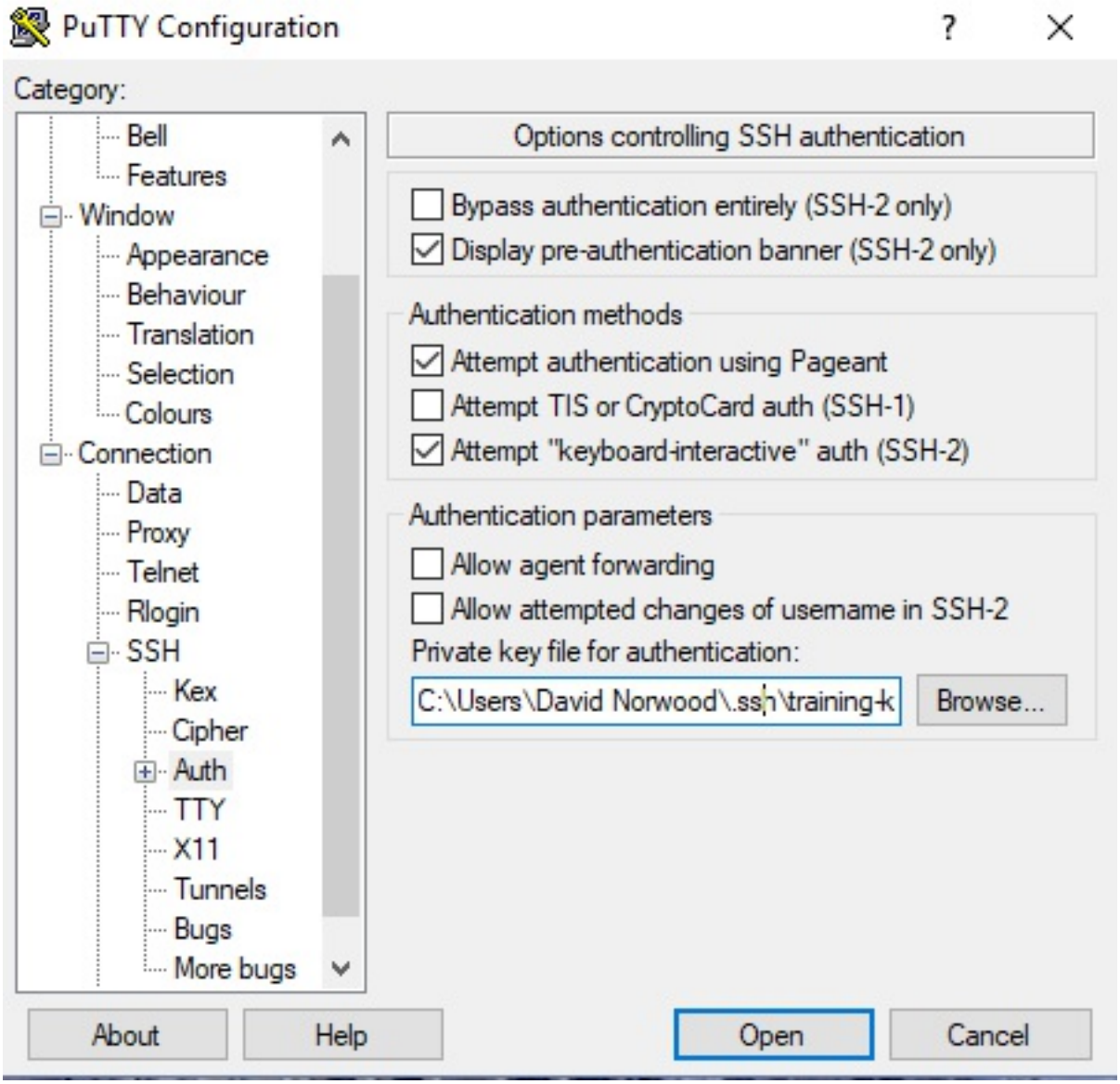
So using Putty, set up the appropriate connections:



And the user page:



And the private key (should be on the share drive):



Now open a connection (to a non-Ambari node):

```
[root@ip-172-30-9-71 ~]# cd /usr/hdp/
2.6.5.0-292/ current/
[root@ip-172-30-9-71 ~]# cd /usr/hdp/current/
hadoop-client/                  hadoop-mapreduce-client/
   hive-webhcat/                   spark2-thriftserver/
hadoop-hdfs-client/             hadoop-mapreduce-historyserve
r/ kafka-broker/                   spark-client/
hadoop-hdfs-datanode/           hadoop-yarn-client/
```

```
   livy2-client/                    spark-historyserver/
hadoop-hdfs-journalnode/         hadoop-yarn-nodemanager/
   livy2-server/                    spark_llap/
hadoop-hdfs-namenode/            hadoop-yarn-resourcemanager/
   livy-client                      spark-thriftserver/
hadoop-hdfs-nfs3/                hadoop-yarn-timelineserver/
   pig-client/                      storm-slider-client/
hadoop-hdfs-portmap/             hive-client/
   shc/                             tez-client/
hadoop-hdfs-secondarynamenode/   hive-metastore/
   slider-client/                   zeppelin-server/
hadoop-hdfs-zkfc/                hive-server2/
   spark2-client/                   zookeeper-client/
hadoop-httpfs                    hive-server2-hive2/
   spark2-historyserver/            zookeeper-server/
```

Then go to the `kafka-broker':

```
[root@ip-172-30-9-71 ~]# cd /usr/hdp/current/kafka-broker/
[root@ip-172-30-9-71 kafka-broker]# ll
total 52
drwxr-xr-x. 3 root root  4096 May 30 18:35 bin
lrwxrwxrwx. 1 root root    24 May 30 18:35 conf -> /etc/kafka
/2.6.5.0-292/0
lrwxrwxrwx. 1 root root    31 May 30 18:35 config -> /usr/hdp
/2.6.5.0-292/kafka/conf
drwxr-xr-x. 4 root root    35 May 30 18:35 doc
```

```
drwxr-xr-x. 3 root root    16 May 30 18:35 examples
drwxr-xr-x. 3 root root  8192 May 30 18:35 libs
-rw-r--r--. 1 root root 28824 May 11 07:59 LICENSE
lrwxrwxrwx. 1 root root    14 May 30 18:35 logs -> /var/log/k
afka
-rw-r--r--. 1 root root   336 May 11 07:59 NOTICE
lrwxrwxrwx. 1 root root    14 May 30 18:35 pids -> /var/run/k
afka
```

```
[root@ip-172-30-9-71 kafka-broker]#  bin/kafka-topics.sh  --c
reate  --zookeeper  localhost:2181  --replication-factor  1
--partitions  1  --topic  Hello-Kafka
Created topic "Hello-Kafka".
[root@ip-172-30-9-71 kafka-broker]#
```

You’re all set!

Mac

There are several alternatives for Putty on the Mac:

1. Alternatives
2. Cyberduck

Troubleshooting

Instructs in this course may change from time to time, or based on your

particular settings. See below for help.

Wrong node

You may be running Kafka commands on the wrong node, usually this will be the Ambari node:

```
[root@ip-172-30-10-248 ~]# cd /usr/hdp/current
[root@ip-172-30-10-248 current]# ls -al | grep kafka
[root@ip-172-30-10-248 current]#
```

So go to another node. What you shuld see is something like this:

```
[root@ip-172-30-9-71 current]# ls -al|grep kafka
lrwxrwxrwx. 1 root root   26 May 30 18:35 kafka-broker -> /us
r/hdp/2.6.5.0-292/kafka
[root@ip-172-30-9-71 current]#
```

Then you can `cd` to the kafta-broker directory:

```
[root@ip-172-30-9-71 current]# cd kafka-broker/
[root@ip-172-30-9-71 kafka-broker]# ls -al
total 52
drwxr-xr-x. 3 root root  4096 May 30 18:35 bin
lrwxrwxrwx. 1 root root    24 May 30 18:35 conf -> /etc/kafka
/2.6.5.0-292/0
lrwxrwxrwx. 1 root root    31 May 30 18:35 config -> /usr/hdp
```

```
/2.6.5.0-292/kafka/conf
drwxr-xr-x. 4 root root    35 May 30 18:35 doc
drwxr-xr-x. 3 root root    16 May 30 18:35 examples
drwxr-xr-x. 3 root root  8192 May 30 18:35 libs
-rw-r--r--. 1 root root 28824 May 11 07:59 LICENSE
lrwxrwxrwx. 1 root root    14 May 30 18:35 logs -> /var/log/k
afka
-rw-r--r--. 1 root root   336 May 11 07:59 NOTICE
lrwxrwxrwx. 1 root root    14 May 30 18:35 pids -> /var/run/k
afka
[root@ip-172-30-9-71 kafka-broker]#
```

Ports

The following table lists the default ports used by Kafka:

Servers	Default Port	Default Ambari Port	Protocol
Kafka Server	9092	6667	TCP

Most of these labs list `localhost:6667` as the `host:port` combination. If that doesn't work you may want to check to see just who's listening to the port:

```
[root@ip-172-30-9-71 kafka-broker]# netstat -tulpn
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address
       State       PID/Program name
```

```
tcp        0      0 127.0.0.1:55300         0.0.0.0:*
       LISTEN      5932/java
tcp        0      0 0.0.0.0:2181            0.0.0.0:*
       LISTEN      5780/java
tcp        0      0 0.0.0.0:8999            0.0.0.0:*
       LISTEN      25469/java
tcp        0      0 0.0.0.0:8040            0.0.0.0:*
       LISTEN      25629/java
tcp        0      0 172.30.9.71:2888        0.0.0.0:*
       LISTEN      5780/java
tcp        0      0 0.0.0.0:35464           0.0.0.0:*
       LISTEN      5780/java
tcp        0      0 0.0.0.0:4200            0.0.0.0:*
       LISTEN      11213/shellinaboxd
tcp        0      0 0.0.0.0:7337            0.0.0.0:*
       LISTEN      25629/java
tcp        0      0 0.0.0.0:8042            0.0.0.0:*
       LISTEN      25629/java
tcp        0      0 0.0.0.0:8010            0.0.0.0:*
       LISTEN      5932/java
tcp        0      0 0.0.0.0:9995            0.0.0.0:*
       LISTEN      24985/java
tcp        0      0 172.30.9.71:6667        0.0.0.0:*
       LISTEN      6347/java
tcp        0      0 0.0.0.0:45454           0.0.0.0:*
       LISTEN      25629/java
tcp        0      0 172.30.9.71:3888        0.0.0.0:*
       LISTEN      5780/java
```

```
tcp        0      0 0.0.0.0:22                  0.0.0.0:*
       LISTEN      14373/sshd
tcp        0      0 0.0.0.0:7447                0.0.0.0:*
       LISTEN      25629/java
tcp        0      0 127.0.0.1:25                0.0.0.0:*
       LISTEN      1691/master
tcp        0      0 0.0.0.0:13562               0.0.0.0:*
       LISTEN      25629/java
tcp        0      0 0.0.0.0:50010               0.0.0.0:*
       LISTEN      5932/java
tcp        0      0 0.0.0.0:50075               0.0.0.0:*
       LISTEN      5932/java
tcp        0      0 0.0.0.0:8670                0.0.0.0:*
       LISTEN      15789/python
tcp        0      0 0.0.0.0:60928               0.0.0.0:*
       LISTEN      6347/java
tcp        0      0 0.0.0.0:18081               0.0.0.0:*
       LISTEN      6885/java
tcp6       0      0 :::22                       :::*
       LISTEN      14373/sshd
tcp6       0      0 ::1:25                      :::*
       LISTEN      1691/master
udp        0      0 0.0.0.0:68                  0.0.0.0:*
                   754/dhclient
udp        0      0 172.30.9.71:123             0.0.0.0:*
                   14471/ntpd
udp        0      0 127.0.0.1:123               0.0.0.0:*
                   14471/ntpd
```

```
udp        0      0 0.0.0.0:123             0.0.0.0:*
                    14471/ntpd
udp        0      0 0.0.0.0:13113           0.0.0.0:*
                    754/dhclient
udp6       0      0 :::123                  :::*
                    14471/ntpd
udp6       0      0 :::34373                :::*
                    754/dhclient
```

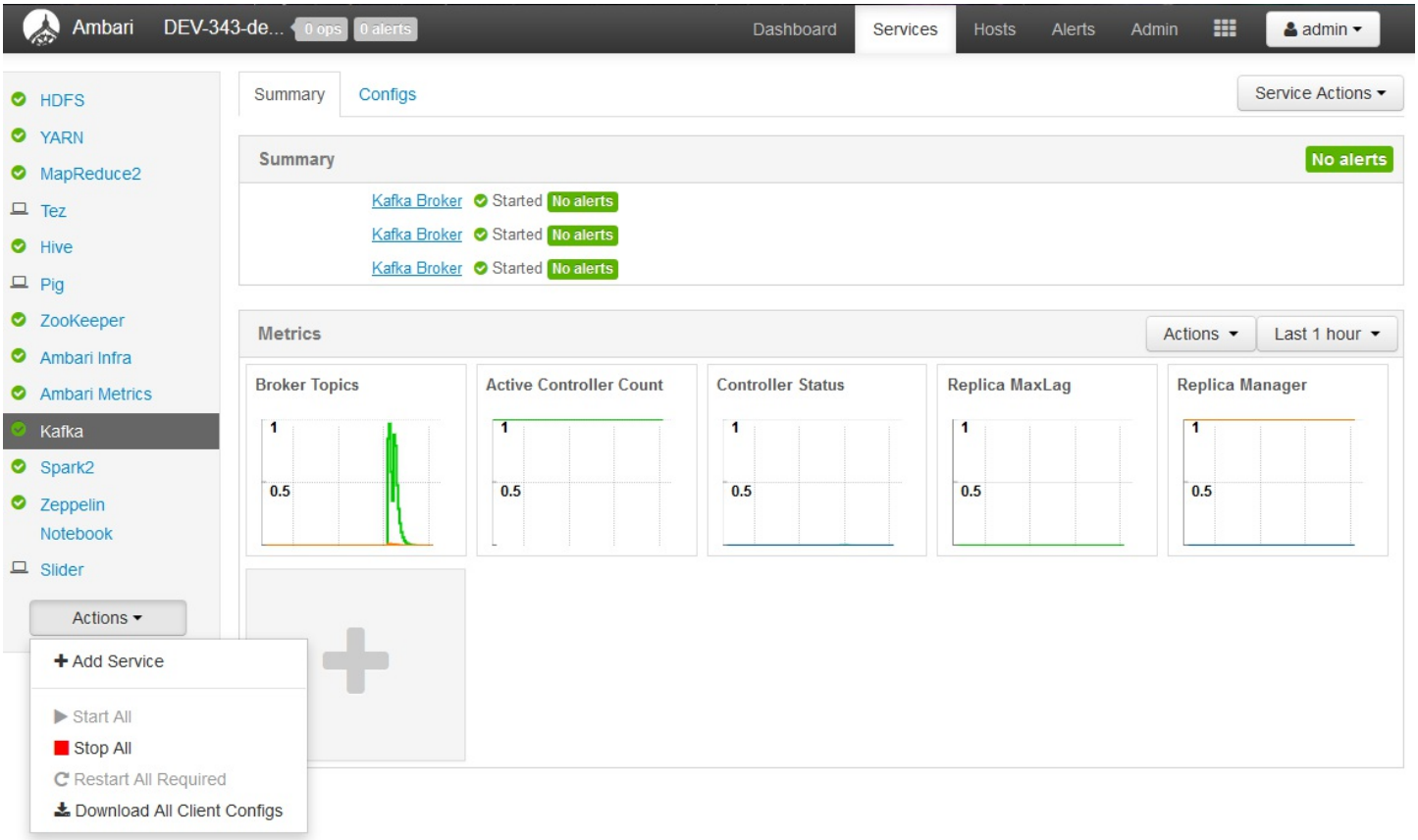
You can see that port 6667 is being used by another port than localhost. So let’s change your command to the host shown:

```
[root@ip-172-30-9-71 kafka-broker]# bin/kafka-console-produce
r.sh  --broker-list 172.30.9.71:6667  --topic  Hello-Kafka
>Hello
```

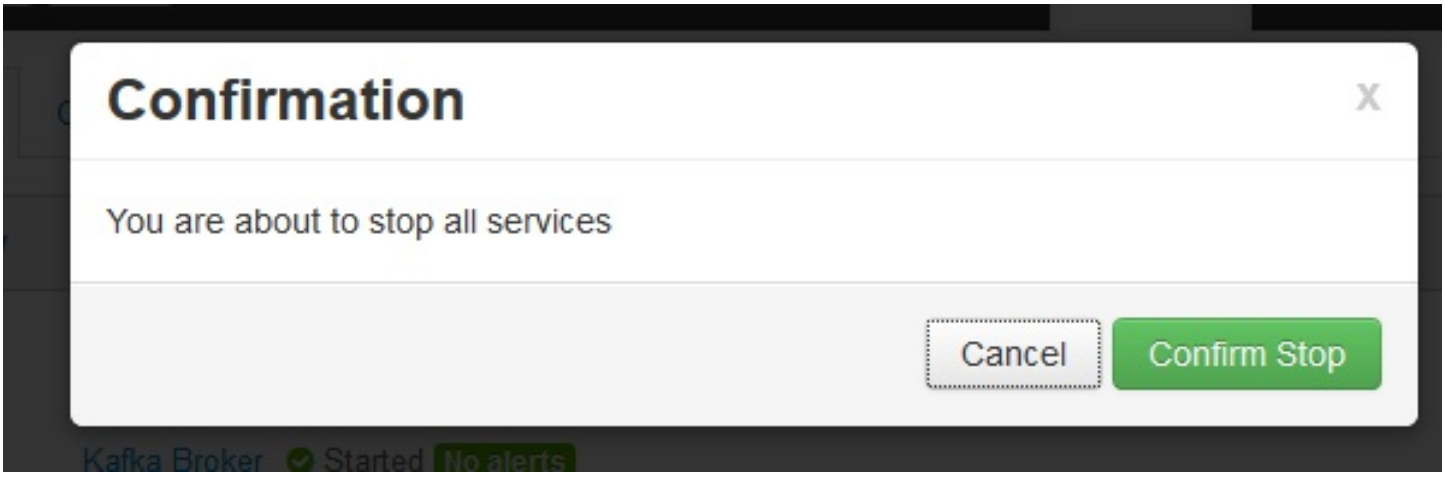
> *Note: if there is no port 6667 showing up, you may need to go to the next step.*

Restart All Services

When the above doesn’t work, then you may have to resort to this:



And then:



When completed, restart all.