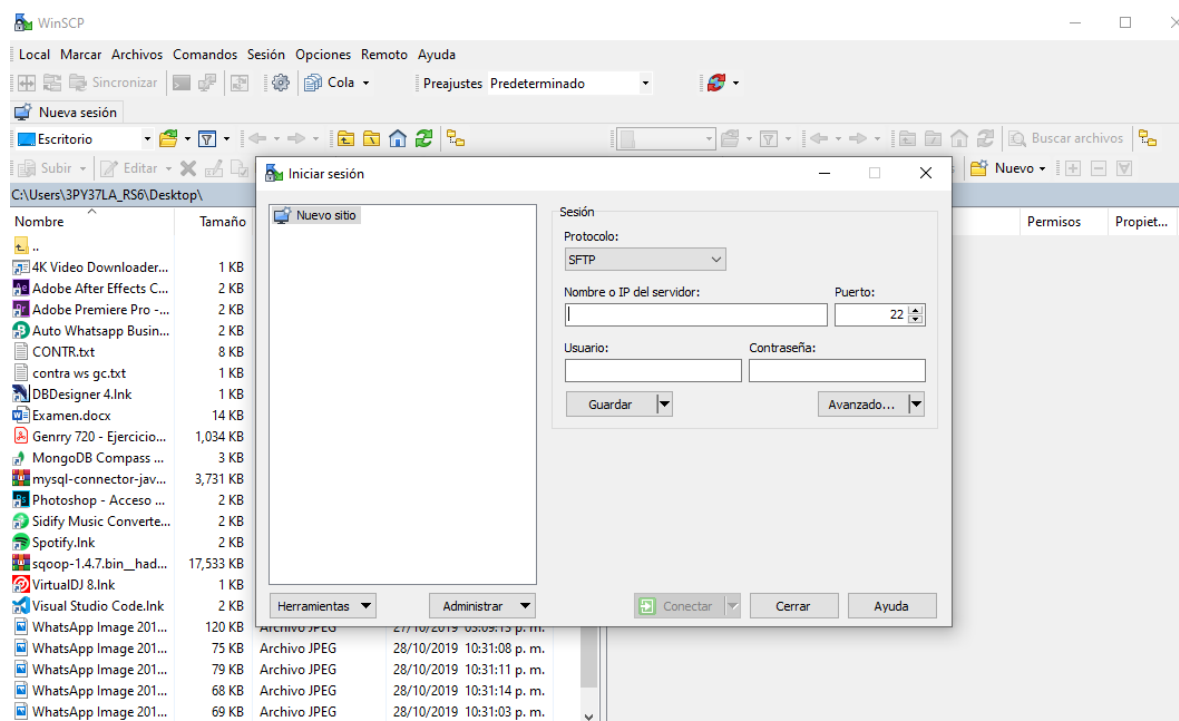




Para la instalación de flume primero tenemos que descargar el archivo con la extensión.tar.gz el cual nos dejara un archivo comprimido en nuestro escritorio



Después usamos el siguiente programa para la conexión entre la maquina nativa y la virtual, con ello copiaremos el archivo comprimido



Seguido lo listamos con el comando -LS y veremos que exista para despues extraerlo en esa carpeta con el comando

TAR -XVF 'NOMBRE DEL ARCHIVO

Despues movemos el archivo que nos da despues de extraer a la carpeta U01 con el siguiente comando

```
mv apache-flume-1.7.0-bin /u01/
```

LUIS HEBERTO BARRERA SALAS #15050915



```
[hdc@m1 u01]$ ls
apache-flume-1.9.0-bin      jdk-8u221-linux-x64.tar.gz
apache-flume-1.9.0-bin.tar.gz  lost+found
hadoop-2.6.5               sqoop-1.4.7.bin_hadoop-2.6.0
hadoop-2.6.5.tar.gz        sqoop-1.4.7.bin_hadoop-2.6.0.tar.gz
jdk1.8.0_221
[hdc@m1 u01]$
```

Con esto procedemos a la instalación de este y su configuración;

Creamos en link con el siguiente comando y después actualizamos el path

Create softlink;

```
$ sudo ln -s /u01/apache-flume-1.7.0-bin /usr/local/flume
```

Update the Path;

```
$ vi .bashrc
```

```
export FLUME_HOME=/usr/local/flume
```

```
export FLUME_CONF_DIR=$FLUME_HOME/conf
```

```
export FLUME_CLASSPATH=$FLUME_CONF_DIR
```

```
export PATH=$PATH:$FLUME_HOME/bin
```

```
# .bashrc
export JAVA_HOME=/u01/jdk1.8.0_221/jre
export HADOOP_INSTALL=/usr/local/hadoop
export SQOOP_HOME=/usr/local/sqoop
export PATH=/u01/jdk1.8.0_221/bin:$PATH:$HADOOP_INSTALL/bin:$HADOOP_INSTALL/sbin
:$SQOOP_HOME/bin
export FLUME_HOME=/usr/local/flume
export FLUME_CONF_DIR=$FLUME_HOME/conf
export FLUME_CLASSPATH=$FLUME_CONF_DIR
export PATH=$PATH:$FLUME_HOME/bin
# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi

# Uncomment the following line if you don't like systemctl's auto-paging feature
:
# export SYSTEMD_PAGER=

# User specific aliases and functions

~
".bashrc" 20L, 589C          5,1          Todo
```

despues actualizamos el entorno de FLUME con VI



Update the Flume Environment;

```
$ cd $FLUME_CONF_DIR
```

```
$ mv flume-env.sh.template flume-env.sh
```

```
$ vi flume-env.sh
```

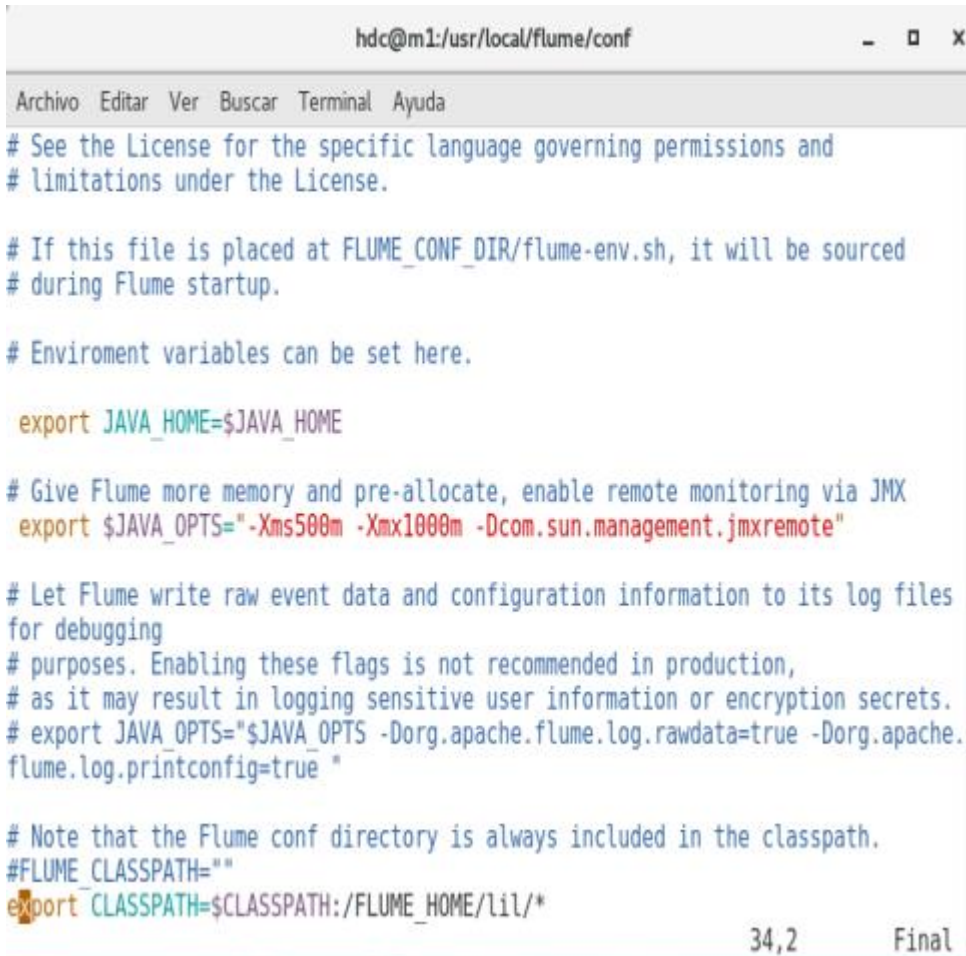
```
export JAVA_HOME=$JAVA_HOME
```

```
$JAVA_OPTS="-Xms500m -Xmx1000m -
```

```
Dcom.sun.management.jmxremote"
```

```
export CLASSPATH=$CLASSPATH:/FLUME_HOME/lib/*
```

```
<save and exit>
```



```
hdc@m1:usr/local/flume/conf
```

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
```

```
# See the License for the specific language governing permissions and
# limitations under the License.

# If this file is placed at FLUME_CONF_DIR/flume-env.sh, it will be sourced
# during Flume startup.

# Enviroment variables can be set here.

export JAVA_HOME=$JAVA_HOME

# Give Flume more memory and pre-allocate, enable remote monitoring via JMX
export $JAVA_OPTS="-Xms500m -Xmx1000m -Dcom.sun.management.jmxremote"

# Let Flume write raw event data and configuration information to its log files
for debugging
# purposes. Enabling these flags is not recommended in production,
# as it may result in logging sensitive user information or encryption secrets.
# export JAVA_OPTS="$JAVA_OPTS -Dorg.apache.flume.log.rawdata=true -Dorg.apache.
flume.log.printconfig=true "

# Note that the Flume conf directory is always included in the classpath.
#FLUME_CLASSPATH=""
export CLASSPATH=$CLASSPATH:/FLUME_HOME/lib/*
```

34,2 Final

Guardamos y salimos de VI con los cambios en la configuración.

Luego entramos con vi a las siguientes propiedades, haciendo uso de este comando:

```
vi log4j.properties
```

dando resultado lo siguiente mostrado en pantalla



```
hdc@m1:/usr/local/flume/conf
Archivo Editar Ver Buscar Terminal Ayuda
flume.log.file=flume.log

log4j.logger.org.apache.flume.lifecycle = INFO
log4j.logger.org.jboss = WARN
log4j.logger.org.mortbay = INFO
log4j.logger.org.apache.avro.ipc.NettyTransceiver = WARN
log4j.logger.org.apache.hadoop = INFO
log4j.logger.org.apache.hadoop.hive = ERROR

# Define the root logger to the system property "flume.root.logger".
log4j.rootLogger=${flume.root.logger}

# Stock log4j rolling file appender
# Default log rotation configuration
log4j.appender.LOGFILE=org.apache.log4j.RollingFileAppender
log4j.appender.LOGFILE.MaxFileSize=100MB
log4j.appender.LOGFILE.MaxBackupIndex=10
log4j.appender.LOGFILE.File=${flume.log.dir}/${flume.log.file}
log4j.appender.LOGFILE.layout=org.apache.log4j.PatternLayout
log4j.appender.LOGFILE.layout.ConversionPattern=%d{dd MMM yyyy HH:mm:ss,SSS} %-5p [%t] (%C.%M:%L) %x - %m%n

"log4j.properties" 68L, 3107C 39,0-1 58%
```

Antes de finalizar recargamos el .bash con el comando:
Reload BashRc; \$ source .bashrc

Y finalmente ejecutamos lo siguiente para verificar la instalación correcta de flume:

\$ flume-ng --help



```
--conf-file, -f <file>    specify a config file (required if -z missing)
--zkConnString, -z <str>  specify the ZooKeeper connection to use (required if
-f missing)
--zkBasePath, -p <path>   specify the base path in ZooKeeper for agent configs
--no-reload-conf          do not reload config file if changed
--help, -h               display help text

avro-client options:
--rpcProps, -P <file>    RPC client properties file with server connection param
s
--host, -H <host>        hostname to which events will be sent
--port, -p <port>        port of the avro source
--dirname <dir>          directory to stream to avro source
--filename, -F <file>    text file to stream to avro source (default: std input)
--headerFile, -R <file>  File containing event headers as key/value pairs on eac
h new line
--help, -h               display help text
```

Either --rpcProps or both --host and --port must be specified.

Note that if <conf> directory is specified, then it is always included first in the classpath.

```
[hdc@m1 conf]$
```