하둡2.7.2 설치하기 - (1) Protocol Buffer

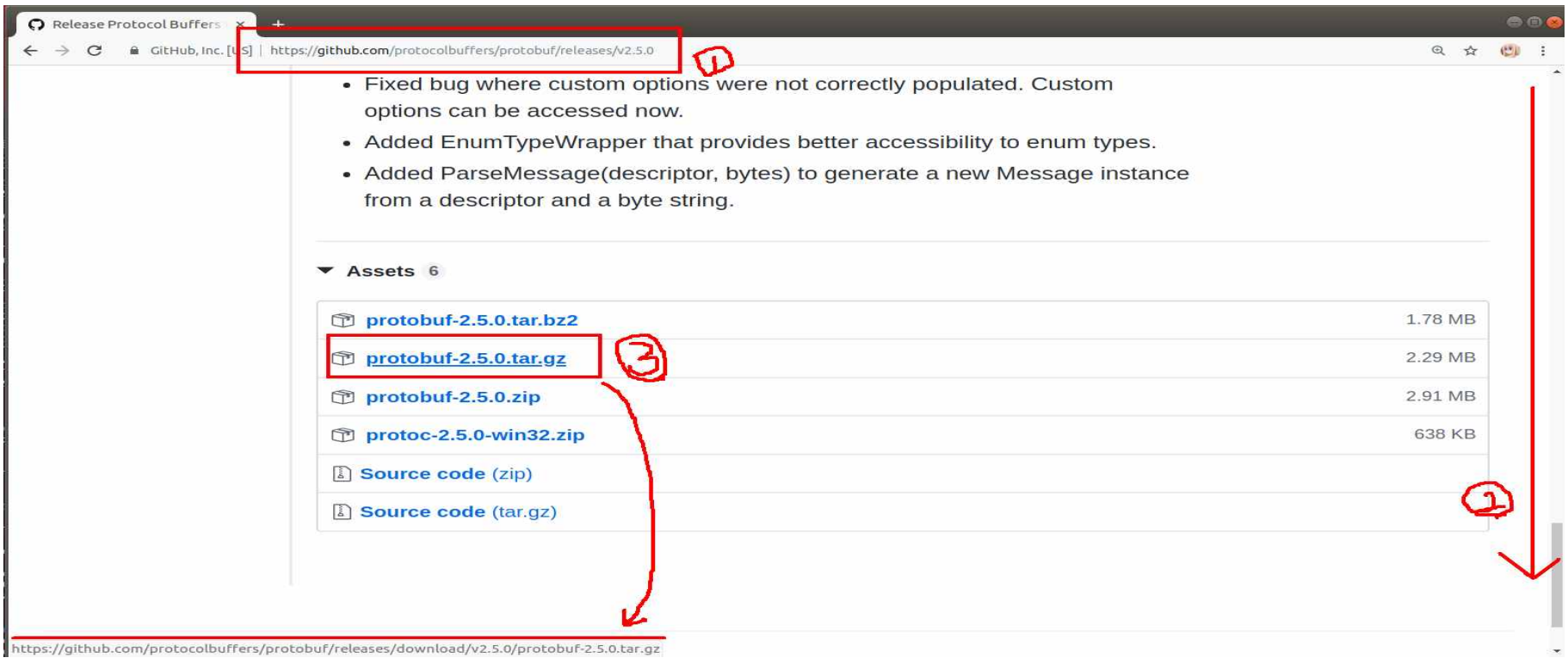
>>>>> 설치 과정은 로그가 너무 길어서 사진으로 대체하겠습니다.

1. 프로토콜 버퍼 설치

원래 먼저 환경설정을 해야하지만 이미 하둡1 실습할 때 완료했으므로 생략하겠습니다.

프로토콜 버퍼는 구글에서 공개한 오픈소스 직렬화 라이브러리입니다.
프로토콜 버퍼는 데이터를 연속된 비트로 만들고, 만들어진 비트를 해석해 원래의 데이터로 만들 수도 있습니다.
하둡2도 내부 데몬간의 데이터 통신을 위해 프로토콜 버퍼를 적용했기 때문에 하둡2를 설치하기 전에 프로토콜 버퍼를 설치했습니다.

1) 아래 사이트에서 protobuf-2.5.0.tar.gz 파일을 다운로드 할 수 있습니다.
https://github.com/protocolbuffers/protobuf/releases/v2.5.0



프로토콜 버퍼를 설치하기 위해 root계정으로 접속했습니다.
실습때 사용했던 wget 명령어를 사용해서 protobuf-2.5.0.tar.gz파일을 다운로드 했습니다.
이 때, /usr/local 디렉터리에서 tar파일을 작업할 것이기 때문에 local디렉터리에서 tar파일을 다운로드 했습니다.

```
root@chosun-VirtualBox:~#
root@chosun-VirtualBox:~# cd /usr/local
root@chosun-VirtualBox:/usr/local# wget https://github.com/google/protobuf/releases/download/v2.5.0/protobuf-2.5.0.tar.gz
--2019-05-24 21:44:10--  https://github.com/google/protobuf/releases/download/v2.5.0/protobuf-2.5.0.tar.gz
Resolving github.com (github.com)... 15.164.81.167
접속 github.com (github.com)|15.164.81.167|:443... 접속됨.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://github.com/protocolbuffers/protobuf/releases/download/v2.5.0/protobuf-2.5.0.tar.gz [following]
--2019-05-24 21:44:10--  https://github.com/protocolbuffers/protobuf/releases/download/v2.5.0/protobuf-2.5.0.tar.gz
Reusing existing connection to github.com:443.
HTTP request sent, awaiting response... 302 Found
Location: https://github-production-release-asset-2e65be.s3.amazonaws.com/23357588/09f5cfca-d24e-11e4-9840-20d894b9ee09?X-Amz
-Amz-Expires=300&X-Amz-Signature=bc6015161b2c9e9b4027de9e54cdc24165754a449103ef85fdb2f8c3dc953514&X-Amz-SignedHeaders=host&ac
ollowing]
--2019-05-24 21:44:10--  https://github-production-release-asset-2e65be.s3.amazonaws.com/23357588/09f5cfca-d24e-11e4-9840-20d
90524T124410Z&X-Amz-Expires=300&X-Amz-Signature=bc6015161b2c9e9b4027de9e54cdc24165754a449103ef85fdb2f8c3dc953514&X-Amz-Signed
octet-stream
Resolving github-production-release-asset-2e65be.s3.amazonaws.com (github-production-release-asset-2e65be.s3.amazonaws.com)..
접속 github-production-release-asset-2e65be.s3.amazonaws.com (github-production-release-asset-2e65be.s3.amazonaws.com)|52.216
HTTP request sent, awaiting response... 200 OK
Length: 2401901 (2.3M) [application/octet-stream]
Saving to: 'protobuf-2.5.0.tar.gz'

protobuf-2.5.0.tar.gz                                100%[=======================================================

2019-05-24 21:44:14 (857 KB/s) - 'protobuf-2.5.0.tar.gz' saved [2401901/2401901]
```

2) protobuf-2.5.0.tar.gz파일을 확인하고 tar 명령어를 사용해 압축 해제했습니다.

```
root@chosun-VirtualBox:/usr/local#
root@chosun-VirtualBox:/usr/local# ls
bin  etc  games  include  lib  man  protobuf-2.5.0.tar.gz  sbin  share  src
root@chosun-VirtualBox:/usr/local#
root@chosun-VirtualBox:/usr/local# tar xvfz protobuf-2.5.0.tar.gz
protobuf-2.5.0/
protobuf-2.5.0/editors/
protobuf-2.5.0/editors/README.txt
protobuf-2.5.0/editors/proto.vim
protobuf-2.5.0/editors/protobuf-mode.el
protobuf-2.5.0/examples/
protobuf-2.5.0/examples/README.txt
protobuf-2.5.0/examples/Makefile
protobuf-2.5.0/examples/addressbook.proto
protobuf-2.5.0/examples/add_person.cc
protobuf-2.5.0/examples/list_people.cc
protobuf-2.5.0/examples/AddPerson.java
```

3) 작업이 완료되면 protobuf-2.5.0 디렉터리가 생성됩니다.

protobuf-2.5.0 디렉터리로 이동해서 configure명령어를 사용해 make 파일을 생성했습니다.

```
root@chosun-VirtualBox:/usr/local# cd protobuf-2.5.0
root@chosun-VirtualBox:/usr/local/protobuf-2.5.0# ./configure
checking whether to enable maintainer-specific portions of Makefiles... yes
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking target system type... x86_64-unknown-linux-gnu
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
```

4) 작업이 완료되면 make파일이 생성됩니다.

소스코드를 컴파일 해서 바이너리 파일을 생성했습니다.

```
root@chosun-VirtualBox:/usr/local/protobuf-2.5.0#
root@chosun-VirtualBox:/usr/local/protobuf-2.5.0# make
make  all-recursive
make[1]: 디렉터리 '/usr/local/protobuf-2.5.0' 들어감
Making all in .
make[2]: 디렉터리 '/usr/local/protobuf-2.5.0' 들어감
make[2]: 디렉터리 '/usr/local/protobuf-2.5.0' 나감
Making all in src
make[2]: 디렉터리 '/usr/local/protobuf-2.5.0/src' 들어감
g++ -DHAVE_CONFIG_H -I. -I..     -pthread -Wall -Wwrite-strings -Woverloaded-virtual -Wno-sign-compare -O2 -g -DNDEB
mv -f .deps/main.Tpo .deps/main.Po
/bin/bash ../libtool  --tag=CXX   --mode=compile g++ -DHAVE_CONFIG_H -I. -I..     -pthread -Wall -Wwrite-strings -Wo
cops_internals_x86_gcc.lo `test -f 'google/protobuf/stubs/atomicops_internals_x86_gcc.cc' || echo './'`google/proto
libtool: compile:  g++ -DHAVE_CONFIG_H -I. -I.. -pthread -Wall -Wwrite-strings -Woverloaded-virtual -Wno-sign-compa
_x86_gcc.cc  -fPIC -DPIC -o .libs/atomicops_internals_x86_gcc.o
libtool: compile:  g++ -DHAVE_CONFIG_H -I. -I.. -pthread -Wall -Wwrite-strings -Woverloaded-virtual -Wno-sign-compa
_x86_gcc.cc -o atomicops_internals_x86_gcc.o >/dev/null 2>&1
mv -f .deps/atomicops_internals_x86_gcc.Tpo .deps/atomicops_internals_x86_gcc.Plo
```

5) 작업이 완료되면 바이너리 파일이 생성됩니다.

생성된 바이너리 파일을 protobuf에서 지정한 디렉토리로 옮겼습니다.

```
root@chosun-VirtualBox:/usr/local/protobuf-2.5.0#
root@chosun-VirtualBox:/usr/local/protobuf-2.5.0# make install
Making install in .
make[1]: 디렉터리 '/usr/local/protobuf-2.5.0' 들어감
make[2]: 디렉터리 '/usr/local/protobuf-2.5.0' 들어감
make[2]: 'install-exec-am'을(를) 위해 할 일이 없습니다.
test -z "/usr/local/lib/pkgconfig" || /bin/mkdir -p "/usr/local/lib/pkgconfig"
 /usr/bin/install -c -m 644 protobuf.pc protobuf-lite.pc '/usr/local/lib/pkgconfig'
make[2]: 디렉터리 '/usr/local/protobuf-2.5.0' 나감
make[1]: 디렉터리 '/usr/local/protobuf-2.5.0' 나감
Making install in src
make[1]: 디렉터리 '/usr/local/protobuf-2.5.0/src' 들어감
make  install-am
make[2]: 디렉터리 '/usr/local/protobuf-2.5.0/src' 들어감
make[3]: 디렉터리 '/usr/local/protobuf-2.5.0/src' 들어감
test -z "/usr/local/lib" || /bin/mkdir -p "/usr/local/lib"
 /bin/bash ../libtool   --mode=install /usr/bin/install -c   libprotobuf-lite.la libprotobuf.la libprotoc.la '/usr/l
libtool: install: /usr/bin/install -c .libs/libprotobuf-lite.so.8.0.0 /usr/local/lib/libprotobuf-lite.so.8.0.0
```

6) 작업이 완료되면 프로토콜 버퍼 설치가 완료됩니다.
protoc —version 명령어를 사용해 버전을 확인할 수 있습니다.
아래 사진에서는 프로토콜 버퍼 설치가 완료되었음에도 버전 확인이 되지 않습니다.

>>> 해결 : vi 편집기로 /etc/ld.so.conf.d위치에서 libprotobuf.conf파일을 생성한 다음,
두 번째 사진과 같이 /usr/local/lib 경로를 입력해주면 정상적으로 프로토콜 버퍼의 버전이 출력됩니다.

```
root@chosun-VirtualBox:/usr/local/protobuf-2.5.0#
root@chosun-VirtualBox:/usr/local/protobuf-2.5.0# protoc --version
protoc: error while loading shared libraries: libprotoc.so.8: cannot open shared object file: No such file or directory
root@chosun-VirtualBox:/usr/local/protobuf-2.5.0# cd ..
root@chosun-VirtualBox:/usr/local# protoc --version
protoc: error while loading shared libraries: libprotoc.so.8: cannot open shared object file: No such file or directory
root@chosun-VirtualBox:/usr/local#
root@chosun-VirtualBox:/usr/local# ls
bin  etc  games  include  lib  man  protobuf-2.5.0  protobuf-2.5.0.tar.gz  sbin  share  src
root@chosun-VirtualBox:/usr/local# protoc --version
protoc: error while loading shared libraries: libprotoc.so.8: cannot open shared object file: No such file or directory
root@chosun-VirtualBox:/usr/local# cd protobuf-2.5.0/
root@chosun-VirtualBox:/usr/local/protobuf-2.5.0# ls
CHANGES.txt       INSTALL.txt  Makefile.in  autogen.sh    config.h.in    config.sub    depcomp   generate_descriptor_proto.sh
CONTRIBUTORS.txt  Makefile     README.txt   config.guess  config.log     configure     editors   gtest
COPYING.txt       Makefile.am  aclocal.m4   config.h      config.status  configure.ac  examples  install-sh
root@chosun-VirtualBox:/usr/local/protobuf-2.5.0# cd ..
root@chosun-VirtualBox:/usr/local# ls
bin  etc  games  include  lib  man  protobuf-2.5.0  protobuf-2.5.0.tar.gz  sbin  share  src
root@chosun-VirtualBox:/usr/local# cd etc
root@chosun-VirtualBox:/usr/local/etc# ls
root@chosun-VirtualBox:/usr/local/etc# ls
root@chosun-VirtualBox:/usr/local/etc#
root@chosun-VirtualBox:/usr/local/etc# cd ..
root@chosun-VirtualBox:/usr/local# ls
bin  etc  games  include  lib  man  protobuf-2.5.0  protobuf-2.5.0.tar.gz  sbin  share  src
root@chosun-VirtualBox:/usr/local# vi /etc/ld.so.conf.d/libprotobuf.conf
root@chosun-VirtualBox:/usr/local# ls
bin  etc  games  include  lib  man  protobuf-2.5.0  protobuf-2.5.0.tar.gz  sbin  share  src
root@chosun-VirtualBox:/usr/local# vi /etc/ld.so.conf.d
root@chosun-VirtualBox:/usr/local# vi /etc/ld.so.conf.d/libprotobuf.conf
root@chosun-VirtualBox:/usr/local# ls
bin  etc  games  include  lib  man  protobuf-2.5.0  protobuf-2.5.0.tar.gz  sbin  share  src
root@chosun-VirtualBox:/usr/local# cd etc
root@chosun-VirtualBox:/usr/local/etc# ls
root@chosun-VirtualBox:/usr/local/etc# cd ..
root@chosun-VirtualBox:/usr/local# sudo ldconfig
root@chosun-VirtualBox:/usr/local# protoc --version
libprotoc 2.5.0
root@chosun-VirtualBox:/usr/local#
```

```
chosun@chosun-VirtualBox: ~
파일(F)  편집(E)  보기(V)  검색(S)  터미널(T)  도움말(H)
/usr/local/lib
~
~
~
~
~
~
~
~
~
~
~
:wq
```