**##### Set environmental variables in AWS WRF #####################**

$ csh

setenv DIR /home/ubuntu/Build\_WRF/LIBRARIES/  
setenv CC gcc  
setenv CXX g++  
setenv FC gfortran  
setenv FCFLAGS -m64  
setenv F77 gfortran  
setenv FFLAGS -m64

setenv PATH $DIR/netcdf/bin:$PATH  
setenv NETCDF $DIR/netcdf

setenv PATH $DIR/mpich/bin:$PATH

setenv PATH $DIR/mpich/bin:$PATH

setenv LDFLAGS -L$DIR/grib2/lib   
setenv CPPFLAGS -I$DIR/grib2/include

setenv JASPERLIB $DIR/grib2/lib  
setenv JASPERINC $DIR/grib2/include

setenv WRF\_CHEM=1

# bash

setenv EM\_CORE=1

setenv WRF\_KPP=1

export CC=gcc

export CXX=g++

export FC=gfortran

export FCFLAGS=-m64  
export F77=gfortran  
export FFLAGS=-m64

export PATH=/home/ubuntu/Build\_WRF/LIBRARIES/netcdf/bin:$PATH  
export NETCDF=/home/ubuntu/Build\_WRF/LIBRARIES/netcdf

export PATH=/home/ubuntu/Build\_WRF/LIBRARIES/mpich/bin:$PATH

export LDFLAGS=-L/home/ubuntu/Build\_WRF/LIBRARIES/grib2/lib

export CPPFLAGS=-I/home/ubuntu/Build\_WRF/LIBRARIES/grib2/include

export JASPERLIB=/home/ubuntu/Build\_WRF/LIBRARIES/grib2/lib  
export JASPERINC=/home/ubuntu/Build\_WRF/LIBRARIES/grib2/include

export FC=gfortran

export NETCDF\_DIR=/home/ubuntu/Build\_WRF/LIBRARIES/netcdf

# to erase previous compilation

./clean –a

check folder size in LINUX

**du -h --max-depth=1 | sort -hr**

Download GFS data for a today date date:

**./download\_GFS0.25\_today.sh**

!!!!! Remove and entire directory !!!!!

**rm -rf /path/to/directory**

**# resize partition after having increased an EBS volume**

lsblk

sudo growpart /dev/xvda 1

**# resize filesystem**

sudo resize2fs /dev/xvda1

**# ncl, nco, and ncvew can be easly installe via “sudo apt-install ncl” etc…**

**#### WRF Met ###########################################**

$ csh

setenv DIR /home/ubuntu/Build\_WRF/LIBRARIES/  
setenv CC gcc  
setenv CXX g++  
setenv FC gfortran  
setenv FCFLAGS -m64  
setenv F77 gfortran  
setenv FFLAGS -m64

setenv PATH $DIR/netcdf/bin:$PATH  
setenv NETCDF $DIR/netcdf

setenv PATH $DIR/mpich/bin:$PATH

setenv PATH $DIR/mpich/bin:$PATH

setenv LDFLAGS -L$DIR/grib2/lib   
setenv CPPFLAGS -I$DIR/grib2/include

setenv JASPERLIB $DIR/grib2/lib  
setenv JASPERINC $DIR/grib2/include

# bash

$

To include erodibility map (EROD)

*If you run WRF-Chem, link the GEOGRIB table to GEOGRIB.TBL\_ARW\_CHEM*

cd /home/ubuntu/Build\_WRF/WPS/geogrid/

**ln -svf GEOGRID.TBL.ARW\_CHEM GEOGRID.TBL**

**without chem use:**

**ln -svf GEOGRID.TBL.ARW GEOGRID.TBL**

**to check links:**

**ls -la**

**#######################################**

cd /home/ubuntu/Build\_WRF/WPS

**# only when required**

./geogrid.exe

**# Link Met data ###############################**

cd /home/ubuntu/Build\_WRF/WPS/

./link\_grib.csh /home/ubuntu/Build\_WRF/forcing\_data/2017111200/gfs\*

# updated the **Vtable** (or place a Vtable file directly in the WPS folder)

ln -sf ungrib/Variable\_Tables/Vtable.GFS\_new Vtable

./ungrib.exe >ungrib.log

./metgrid.exe >metgrid.log

**# WRF MET #####################################################**

cd /home/ubuntu/Build\_WRF/WRFV3/test/em\_real/

**# start WRF MET**

**./real.exe**

**./wrf.exe**

**############ WRF-Chem #############################################**

**library Flex**

sudo apt-get install bison flex

The files are in the libfl-dev package, which is a dependency of flex. The package is Multi-Arch enabled, so the files are under the /usr/lib/i386-linux-gnu/ or /usr/lib/x86\_64-linux-gnu/ directory, depending on whether you are on 32 or 64 bit. Or possibly under /usr/lib/armhf-linux-gnu/ if you are on ARM

Install flex like the common way :  
cd /home/ubuntu/Build\_WRF/LIBRARIES/  
mkdir New\_flex

cd New\_flex

wget <http://www.ncl.ucar.edu/Download/files/flex.tar.gz>  
tar -xvzf flex.tar.gz

vi README

./configure --prefix=/home/ubuntu/Build\_WRF/LIBRARIES/New\_flex

make

make install  
  
Then edit your Makefile.defs within chem/KPP/kpp/kpp-2.1/, with :  
FLEX=/home/jlefevre/flex/bin/flex  
FLEX\_LIB\_DIR=/home/jlefevre/flex/lib  
  
Go to chem/KPP/kpp/kpp-2.1/src  
make clean  
make  
That should to be right. There are some warning, you can add some include like :  
scan.l :  
line 45 #include <string.h>  
gen.c :  
line 37 :  
#include <string.h>  
#include <math.h>

**Need** libfl.a from library flex

$ cd /usr/lib/x86\_64-linux-gnu/

ln -s libfl.a /usr/local/lib/libfl.a

**Download yacc from linux**

$ sudo apt-get install yacc

**Change lines in KPP**

<https://ruc.noaa.gov/wrf/wrf-chem/KPP_yacc_flex_problems.pdf>

**Make changes in the /home/ubuntu/WRFV3\_Chem/chem/KPP/configure.kpp file**

# Uncomment or set your path to yacc in your environment

YACC=/usr/bin/byacc -d

#YACC=/usr/bin/bison

FLEX=flex

# The complete pathname of the FLEX library (libfl.a).

# On many systems this is: /usr/local/util/lib/flex

FLEX\_LIB\_DIR = /usr/lib/x86\_64-linux-gnu

export zlib=/home/ubuntu/Build\_WRF/LIBRARIES/zlib-1.2.7

./configure --prefix=/home/ubuntu/Build\_WRF/LIBRARIES/zlib-1.2.7 --enable-fortran --enable-cxx --with-zlib=/home/ubuntu/Build\_WRF/LIBRARIES/zlib-1.2.7

make  
make test  
make install  
make check-install

LD\_LIBRARY\_PATH=/home/ubuntu/Build\_WRF/LIBRARIES/zlib-1.2.7/lib/:LD\_LIBRARY\_PATH

export HDF5=/home/ubuntu/Build\_WRF/LIBRARIES/hdf5-1.8.12

# export LD\_LIBRARY\_PATH=/usr/lib/x86\_64-linux-gnu:$LD\_LIBRARY\_PATH

export LD\_LIBRARY\_PATH=/home/ubuntu/Build\_WRF/LIBRARIES/New\_flex/lib:$LD\_LIBRARY\_PATH

# export LD\_LIBRARY\_PATH=/usr/local/lib:$LD\_LIBRARY\_PATH

export CC=gcc

export CXX=g++

export FC=gfortran

export FCFLAGS=-m64  
export F77=gfortran  
export FFLAGS=-m64

export PATH=/home/ubuntu/Build\_WRF/LIBRARIES/netcdf/bin:$PATH  
export NETCDF=/home/ubuntu/Build\_WRF/LIBRARIES/netcdf

export PATH=/home/ubuntu/Build\_WRF/LIBRARIES/mpich/bin:$PATH

export LDFLAGS=-L/home/ubuntu/Build\_WRF/LIBRARIES/grib2/lib

export CPPFLAGS=-I/home/ubuntu/Build\_WRF/LIBRARIES/grib2/include

export JASPERLIB=/home/ubuntu/Build\_WRF/LIBRARIES/grib2/lib  
export JASPERINC=/home/ubuntu/Build\_WRF/LIBRARIES/grib2/include

export FC=gfortran

export NETCDF\_DIR=/home/ubuntu/Build\_WRF/LIBRARIES/netcdf

export WRF\_CHEM=1

export EM\_CORE=1

export WRF\_KPP=1

# export FLEX\_LIB\_DIR=/usr/lib/x86\_64-linux-gnu

# export FLEX\_LIB\_DIR=/usr/local/lib

export FLEX\_LIB\_DIR=/home/ubuntu/Build\_WRF/LIBRARIES/New\_flex/lib

# export YACC=/usr/local/bin/yacc

# export PATH=/usr/local/bin:$PATH

export PATH=/home/ubuntu/Build\_WRF/LIBRARIES/New\_flex/bin:$PATH

export NMM\_CORE=0

export WRF\_EM\_CORE=1

**Build\_WRF\_Chem directory**

tar -xvzf WRFV3.8.1.TAR.gz

**untar the WRF-Chem code inside the WRFV3 directory. This will create a sub-directory called '/chem.'**

tar -xvzf WRFV3-Chem-3.8.1.TAR.gz

**./configure  
./compile em\_real 2>&1 |tee compile.log**

Use **./clean -a** to clean previous compilation

**### Download EGDAR/HTAP emission data ####################**

# Create directory EDGAR-HTAP in Build\_WRF\_Chem

# download emission inventory data

wget <http://www.acom.ucar.edu/webt/wrf-chem/processors/EDGAR-HTAP.tgz>

extract the EDGAR/HTAP data:

tar -xvzf EDGAR-HTAP.tgz

**########### anthro\_emiss ##########**

Download **anthro\_emiss** from this page into the Build\_WRF\_Chem:

<https://www2.acom.ucar.edu/wrf-chem/wrf-chem-tools-community>

<http://www.acom.ucar.edu/wrf-chem/download.shtml>

Untar “ANTHRO.tar”

**$ tar –xf ANTHRO.tar**

**$ cd /home/fkaragulian/ANTHRO/src/**

proceed with the compilation of anthro\_emiss

export FC=gfortran

export NETCDF\_DIR=/home/ubuntu/Build\_WRF/LIBRARIES/netcdf

modify the **make\_anthro** in /home/fkaragulian/ANTHRO/src/

*vi make\_anthro*

..make the following changes….

set ar\_libs = -lnetcdff

if( -e ${NETCDF\_DIR}/lib/libnetcdff.a ) then

set ar\_libs = "${ar\_libs} -lnetcdf"

endif

compile anthro\_emiss….

./make\_anthro

it will build the executable anthro\_emis

cd /home/fkaragulian/EDGAR-HTAP/

look at the file “**MOZCART.inp**” as example of input file for the emission (/home/ubuntu/Build\_WRF\_Chem/EDGAR-HTAP/INP-Examples/)

Execution DATE\_TIME of the WRFChem will be taken from the files:

wrfinput\_d01 in the /home/fkaragulian/WRF\_UAE/WRFV3/test/em\_real/ directory…that is the one indicated in the namelist.input (from GFS met data)

**make a copy of the MOZCART.inp in the ANTHRO/src/ directory**

**change date in the MOZCART.inp according to the date of the namelist.input (met)**

To run anthro\_emis, execute the command:

**$ cd /home/fkaragulian/ANTHRO/src/**

**./anthro\_emis < MOZCART.inp > MOZCART\_FK.out**

….it will generate files:

**wrfchemi\_00z\_d01**

**wrfchemi\_12z\_d01**

**##### wesely and exo\_coldens #######################**

download wes-coldens.tar into

**/home/ubuntu/Build\_WRF\_Chem/WRFV3/wesley**

untar wes-coldens.tar in a new directory:

**tar -xvf wes-coldens.tar**

compile the MOZCART source code in the /home/fkaragulian/WRFV3/wesely/

**change to gfortaran processor in the make\_util**

set OP\_SYS=`uname -s`

if( $OP\_SYS == "Linux" ) then

# setenv FC pgf90

setenv FC gfortran

else

setenv FC xlf90

endif

**and in the Makefile in the**

F90 = $(FC)

LIBS = -L$(NETCDF\_DIR)/lib -lnetcdff -lnetcdf

INCLUDE\_MODULES = -I$(NETCDF\_DIR)/include

FFLAGS = -g

**$ cd /home/ubuntu/Build\_WRF\_Chem/WRFV3/wesely/**

./make\_util wesely (will generate wesely)

./make\_util exo\_coldens (will generate exo\_coldens)

**update the namelist.input (in the wesely folder) as follow:**

&control  
 wrf\_dir = ' /home/fkaragulian/WRFV3/test/em\_real/'

wrf\_dir = '/home/fkaragulian/WRF\_UAE/WRFV3/test/em\_real/'  
 domains = 1,  
/

run exo\_coldens with the updated namelist.input

**./exo\_coldens < namelist.input**

Will generate ***exo\_coldens\_d01***

**link wrfchemi\_d01 and exo\_coldens\_d01 to /home/ubuntu/Build\_WRF\_Chem/WRFV3/test/em\_real/**

$ cd /home/ubuntu/Build\_WRF\_Chem/WRFV3/test/em\_real/

ln -sf /home/ubuntu/Build\_WRF\_Chem/ANTHRO/src/wrfchemi\_\* .

ln -sf /home/ubuntu/Build\_WRF\_Chem/WRFV3/wesely/exo\_coldens\_\* .

**link ungribbed met files to**

**/home/ubuntu/Build\_WRF\_Chem/WRFV3/test/em\_real/**

ln -sf /home/ubuntu/Build\_WRF/WRFV3/test/em\_real/met\_em\* .

(generated above with ./metgrid.exe above)

update the **NAMELIST.INPUT.CHEM** in /home/ubuntu/Build\_WRF\_Chem/WRFV3/test/em\_real/

namelist.input.chem MUST be renamed into namelist.input or just link the “name” of namelist.input to namelist.input.chem

ln -s namelist.input.chem namelist.input

**# start WRF\_Chem**

<https://ruc.noaa.gov/wrf/wrf-chem/wrf_tutorial_exercises_v35/compiling_code.html>

**./real.exe**

**./wrf.exe**