# CLM5.0 Tutorial: Creating and Running a CESM Model

We will be using the Community Earth System Model (CESM) is a set of models that can be run independently or together to simulate the Earth global climate. CLM is one of those and our focus for this tutorial will be Community Land Model.

Link to various CESM Releases : <http://www.cesm.ucar.edu/models/>

## Introduction

In this tutorial we will be discussing how to create a new model using CESM. We will be discussing about the following topics :

1. **Downloading CLM-CESM Scripts**

In this Step we will be downloading a few scripts which will aid us through the tutorial.

1. **Creating a new case**

This step sets up a new simulation. It is the most complicated of these four steps because it involves making choices to set up the model configuration.

1. **Setting up a new case**

This step configures the model so that it can compile.

1. **Build the Case**

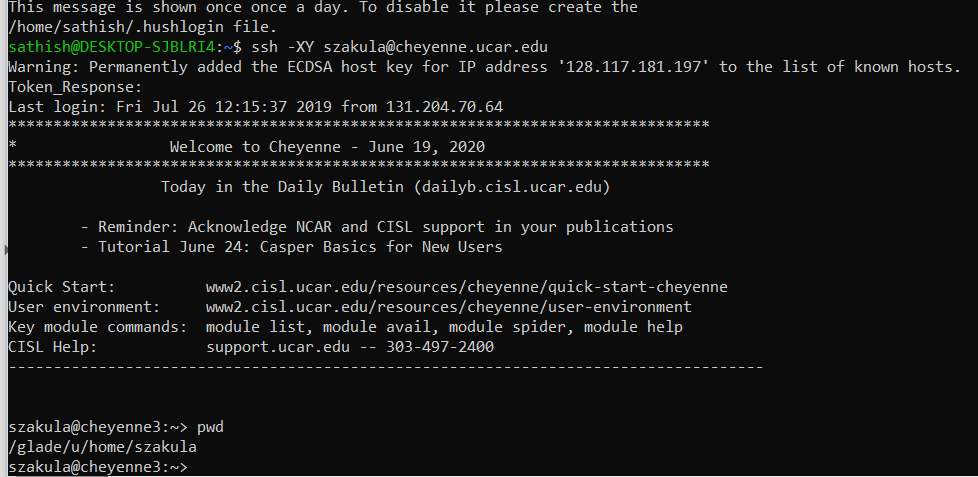
This step compiles the model.

1. **Submitting your Run**

This step submits the model simulation to the supercomputer queue.

## Part 1: Downloading CLM Code

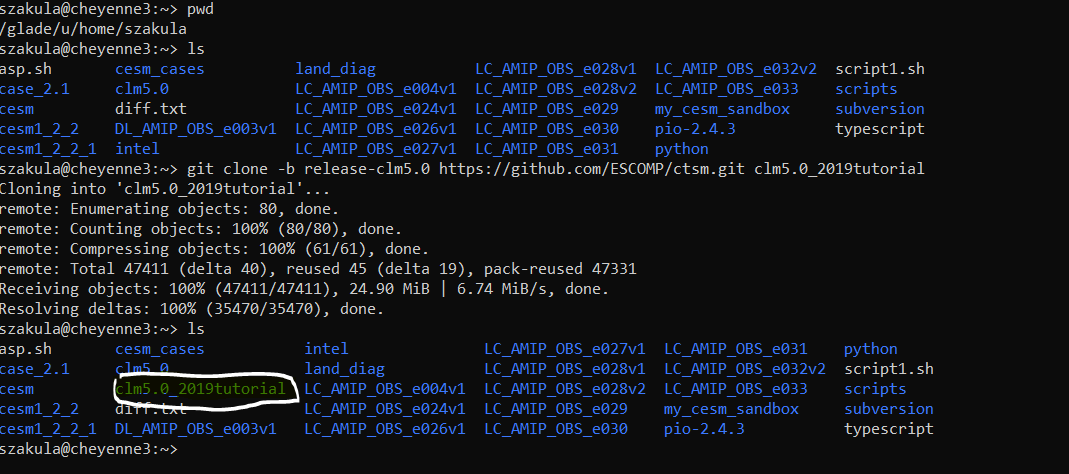
* Go to your home directory in glade (It is located at glade/u/home/username\*)

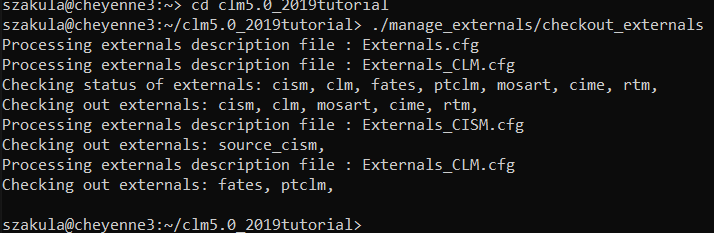


* Download the CLM 5.0 code to your directory.
  + We do this by cloning the GitHub repository using the following code line

>> git clone -b release-clm5.0 https://github.com/ESCOMP/ctsm.git clm5.0\_2019tutorial

* Breakdown
  + **git** – Calling the git operation
  + **Clone**- used to copy the contents from source to destination
  + **-b release-clm5.0** – Specifies the version of clm to download.
  + [**https://github.com/ESCOMP/ctsm.git**](https://github.com/ESCOMP/ctsm.git) - Location of the clm files.
  + **clm5.0\_2019tutorial** – Local file name where the files downloaded are saved.
* After you run the git command on your home directory you should see the file in your home directory.



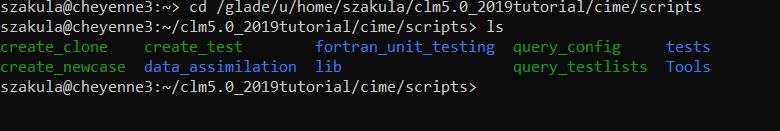
* Open the code-directory by using “cd clm5.0\_2019tutorial” and now check whether you’ve correctly downloaded all the parts of the code by running “./manage\_externals/checkout\_externals”
* After you finish running the command given above your output should look something like this : 

# Step 2: Creating a new case

* To create a new case we use one of the clm scripts which we downloaded by going to the following directory:

>> cd /glade/u/home/<username>/clm5.0\_2019tutorial/cime/scripts

Once you open up the directory you should be able to see the following scripts by using the command “ls” as shown in the image below :



* Run the following command

>> ./create\_newcase --case ~/clm\_tutorial\_cases/I1850CLM50\_001 --res f19\_g17 --compset I1850Clm50Sp --project UOLA002

Breakdown:

“create\_newcase” The name of the script which you need to run

“--case” is used to specify the location of the new case

“~” Is used to specify home directory

“~/clm\_tutorial\_cases” is used to refer the directory where the new case has to be created (please note make a directory using the following command :

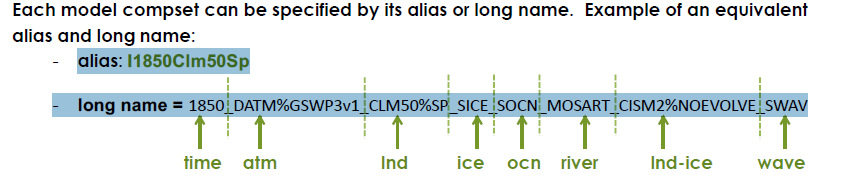
>>mkdir ~/clm\_tutorial\_cases/

“I1850CLM50\_001” is the name of the case which we are creating

“--res f19\_g17” is the resolution of the model (f19\_g17 is a 2-degree grid)

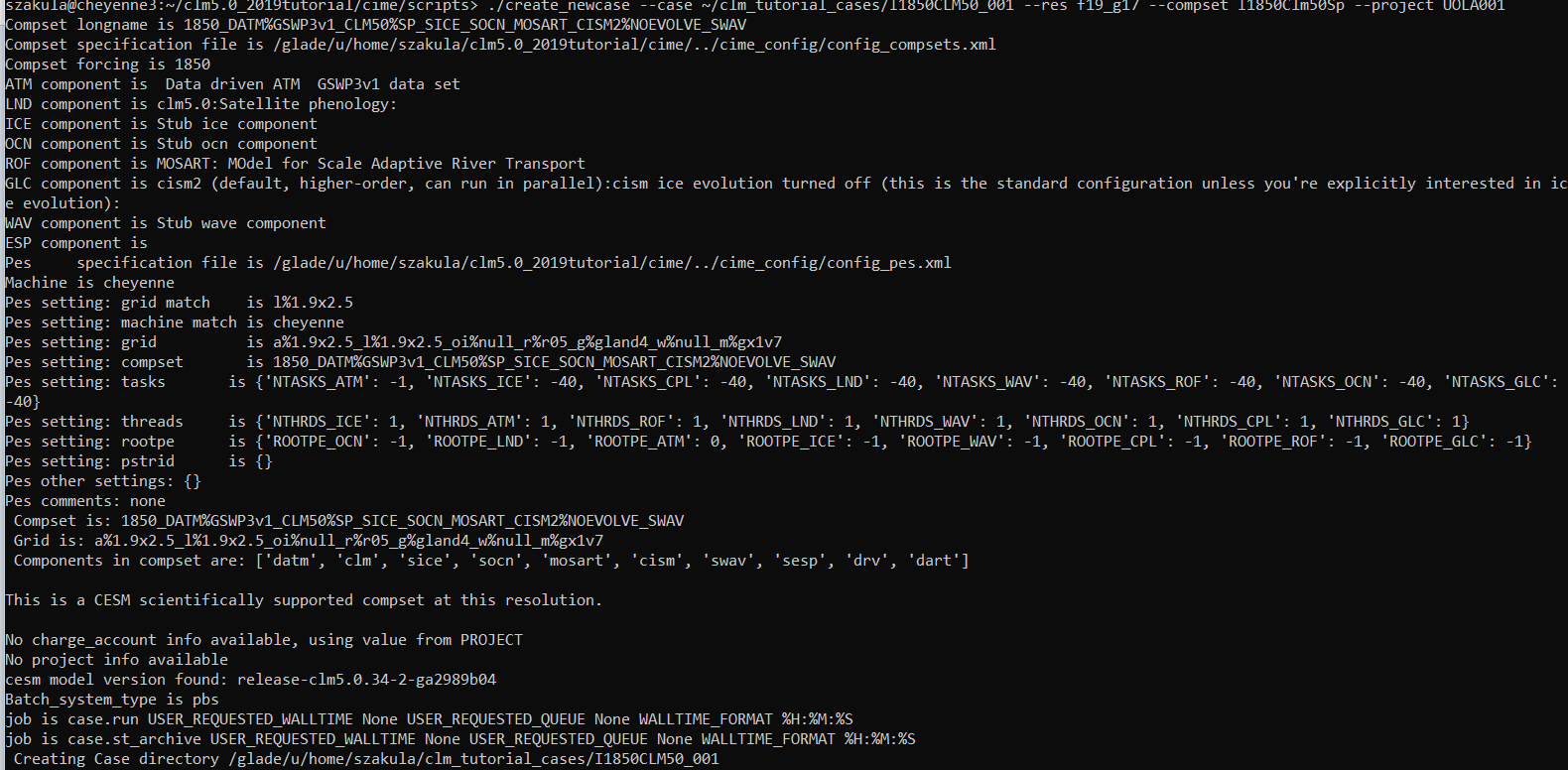
“--compset I1850Clm50Sp” compset specifies the “component set”

“--project UOLA002” is the project ID which you are working on



* More info about the configuration can be found at : <http://www.cesm.ucar.edu/models/cesm2/> under the Configurations and Grids section.

After you run the command you should be able to find your case in the clm\_tutorial\_case directory.



Your output for the create case should look something like this.

Note: create\_newcase creates case directory that contains:

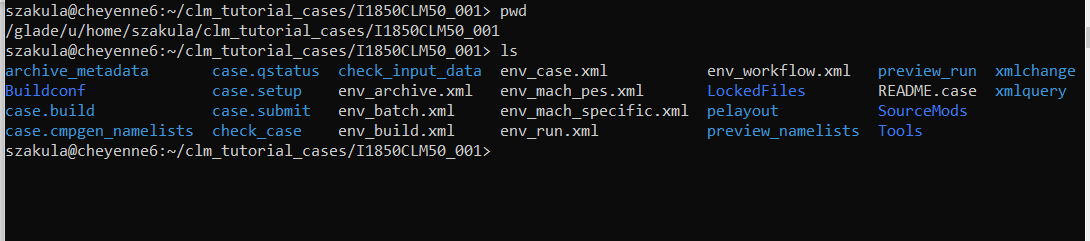
case\_setup: script used in the next step

User customizable case xml files script to edit env\_\*xml files

# Step 3: Setting up a new case

You will be working with the case.setup which is found in the directory where you created your case.

pwd is the command used to know the current working directory



Location where my case has been created.

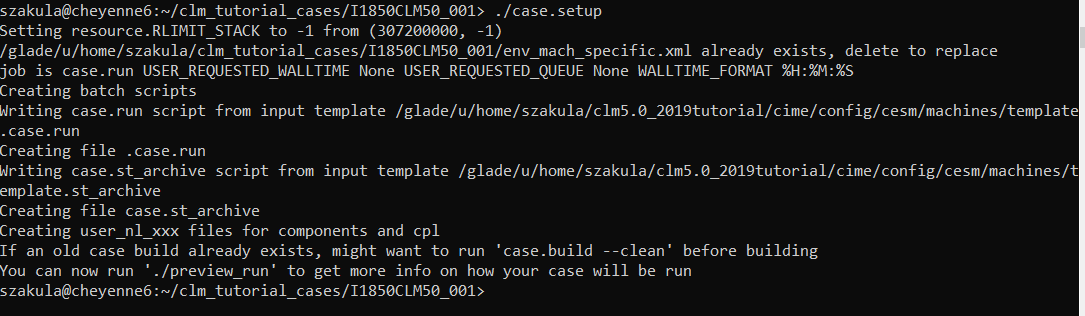
Navigate to the directory where the case is created by using

>> cd /glade/u/home/<username>/clm\_tutorial\_cases/I1850CLM50\_001

After navigating to the directory where your case is created setup your case by running

>> ./case.setup

One the case.setup script is executed you should see an output like this



# Step 4 : Build the Case

In the same directory i.e. in your case directory, run the following command to build the case.

>> qcmd -- ./case.build

Note that “qcmd --” is specific for cheyenne and runs the command on a computing node, reducing the load on the login node.

Note that it might take up to 20 mins for the case.build script to finish building the case.

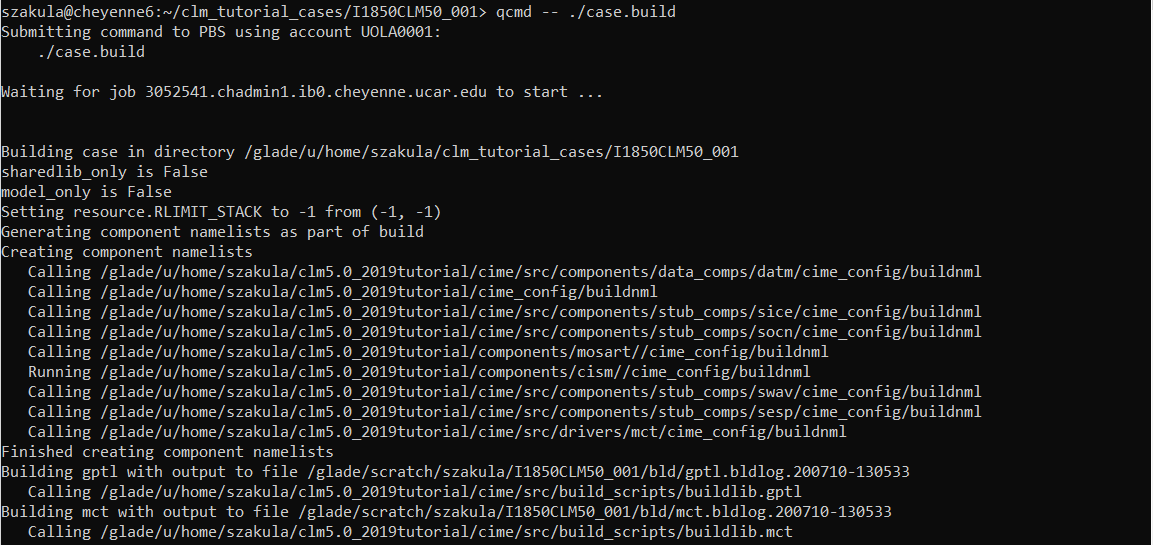
Please refer to the section below step 4 to go over the various ways in which we can edit the case to suit our needs.

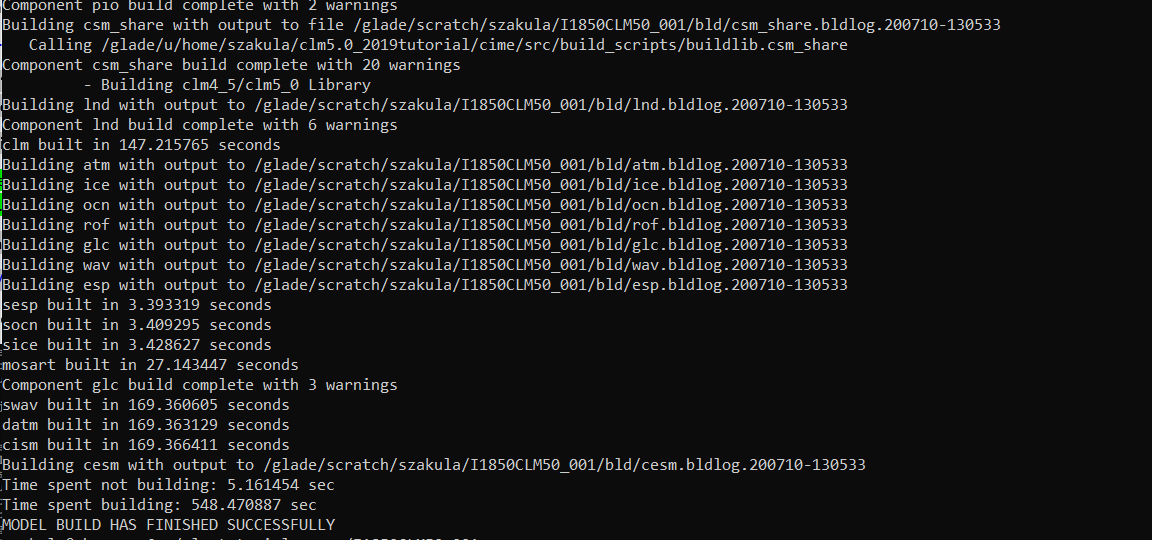
Note: If any input data is missing, build aborts and provides a list of missing files. You can run

>> ./check\_input\_data --download to acquire missing data

If your case builds successfully your output should look like this

Command used to build the case.





Look for this message to check for successful build

# Step 5: Submitting your Run

After you made the necessary changes in the required files run following command

>> ./case.submit

## **Case Customization Checks :**

Once the Build script finishes executing, the model is now compiled and ready to run! There are a few things we should check before submitting the run. For example:

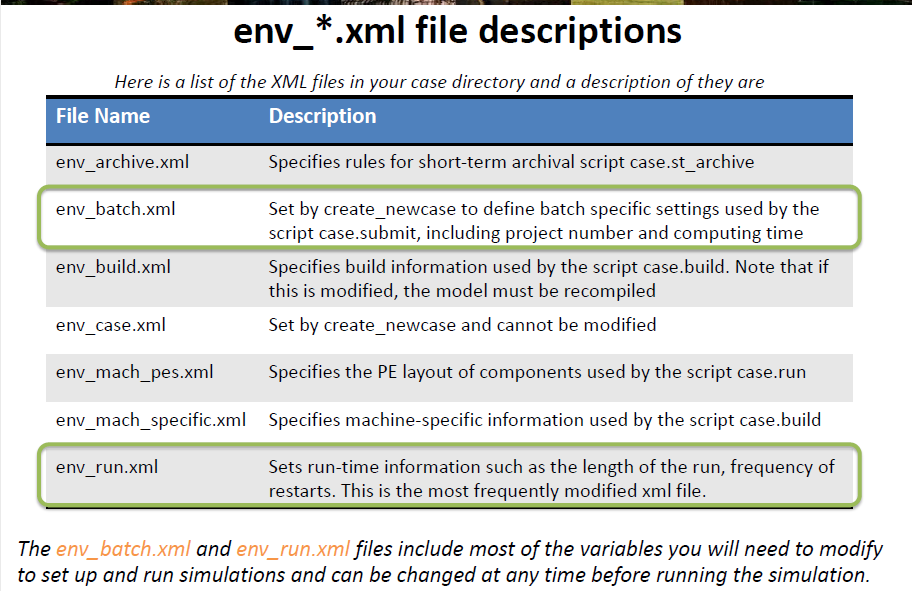
1) How many days or years will the model simulate?

2) How much time does the computer need for this simulation?

3) Which computing project account is the model charging to?

These options are specified in the “env\_\*.xml” files in your case directory

Note : Using the “xmlchange" script is the preferred method, but you can edit XML files by hand.



*Few Useful tips for using XML scripts*

*1. Use “./xmlquery --listall” to list variables and their values in the .xml files*

*2. Modify a variable in a .xml file, use “./xmlchange”*

*3. For help, type ./xmlchange --help*

*Example: editing env\_\*.xml via the xmlchange tool*

*./xmlchange {variable to be changed}={value to change to}*

## **Editing few important variables in XML files**

Many runtime variables are found in the env\_run.xml file. The variables in this file control the mechanics of the run (length, resubmits, and archiving).

Common variables in env\_run.xml to change include:

1. STOP\_OPTION: Sets the run-time interval type, i.e. nmonths, ndays, nyears

2. STOP\_N: Sets the number of run-time intervals to run the model during the specified wallclock\* time.

\* Wallclock time is set in the env\_batch.xml file and is a measure of the actual time.

3. RESUBMIT: Sets the number of times to resubmit the run

NOTE: By default, the model is set to run for 5 days\*. Let’s change the length of the simulation to 20 years. We will not see any model output from a 5-day model run because history files with model output are only recorded every month by default

### **Run the following commands to set the length to 20 years:**

>> ./xmlchange STOP\_OPTION=nyears

This changes the run-time interval from days to years. Verify that the change worked as you expected:

>> ./xmlchange STOP\_N = 20

Run these two commands if you’ve made the required changes

>> ./xmlquery STOP\_N \*checks for the value stored in STOP\_N which should be 20.

>> ./xmlquery STOP\_OPTION \*checks for the value stored in STOP\_OPTION which should be nyears.

Now that you have confirmed that the run-time interval is years, verify that the model will run for 20 years.

We also need to check that we are using the correct project code and giving the computer enough time to run the simulation. These are found in the env\_batch.xml file.

### **Run the following commands to check the wall time and project code:**

We also need to check that we are using the correct project code and giving the computer enough time to run the simulation. These are found in the env\_batch.xml file.

To Do: Use xmlquery to check the project id and wall clock

1. PROJECT : UOLA001

>> ./xmlquery PROJECT = UOLA001

2. JOB\_WALLCLOCK\_TIME : 20:00:00

>> ./xmlquery JOB\_WALLCLOCK\_TIME = 20:00:00

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Reference : <http://www.cesm.ucar.edu/events/tutorials/2016-clm/2016CLMTutorial_day1-practical-lombardozzi.pdf>