WDL 101: Running WDL workflows at Fred Hutch using Cromwell

October, 2022

Table of Contents

# About this Course

## 0.1 Available course formats

This course is available in multiple formats which allows you to take it in the way that best suites your needs. You can take it for certificate which can be for free or fee.

* The material for this course can be viewed without login requirement on this [Bookdown website](LINK%20HERE). This format might be most appropriate for you if you rely on screen-reader technology.
* This course can be taken for [free certification through Leanpub](LINK%20HERE).
* This course can be taken on [Coursera for certification here](LINK%20HERE) (but it is not available for free on Coursera).
* Our courses are open source, you can find the [source material for this course on GitHub](LINK%20HERE).

# 1 Introduction

## 1.1 Motivation

## 1.2 Target Audience

The course is intended for …

## 1.3 Content

A repo containing instructions for running a Cromwell server on gizmo at the Fred Hutch. These instructions were created and tested by Amy Paguirigan, so drop her a line if they don’t work for you or you need help (Fred Hutch username is apaguiri) or you can tag ([**vortexing?**](#ref-vortexing)) in Issues filed here in the GitHub repository. Note if you do this, please be sure that you do not post sensitive information in your troubleshooting information like passwords and such, but the more info you can provide about errors, the better. Alternatively, join the discussion in the Fred Hutch Bioinformatics and Computational Research Community Slack in the [#question-and-answer channel](https://fhbig.slack.com/archives/CD3HGJHJT) using your Fred Hutch, UW, SCHARP or Sagebase email.

## 1.4 Cromwell Resources

Cromwell is a workflow manager developed by the Broad which manages the individual tasks involved in multi-step workflows, tracks job metadata, provides an API/Swagger UI interface and allows users to manage multiple workflows simultaneously. Cromwell currently uses either CWL or WDL workflow languages, and we will focus on WDL workflows for now. [Learn more on the Fred Hutch wiki about using Cromwell at Fred Hutch.](https://sciwiki.fredhutch.org/compdemos/Cromwell/)

You can see what is currently available in the FredHutch GitHub institution by using [this link to search results](https://github.com/FredHutch?utf8=%E2%9C%93&q=wdl+OR+cromwell&type=&language=).

We have also been building other places to find information about both workflow managers being supported at the Fred Hutch via [this GitHub Project](https://github.com/orgs/FredHutch/projects/8).

Amy also made a shiny app you can use to monitor your own Cromwell server workflows when you have a Cromwell server running on gizmo that can be found [here](https://cromwellapp.fredhutch.org/).

## 1.5 Steps to prepare

If you have questions about these steps, feel free to contact Amy Paguirigan (apaguiri) or scicomp.

### 1.5.1 Rhino Access (one time)

Currently, to run your own Cromwell server you’ll need to know how to connect to rhino at the Fred Hutch. If you have never used the local cluster (rhino/gizmo), you may need to file a ticket by emailing fredhutch email scicomp and requesting your account be set up. To do this you’ll need to specify which PI you are sponsored by/work for. You also may want to read a bit more about the use of our cluster over at [SciWiki](https://sciwiki.fredhutch.org/) in the Scientific Computing section about Access Methods, and Technologies.

### 1.5.2 AWS S3 Access (optional as of version 1.3)

Prior to release 1.3, this setup required that you have a set of AWS credentials for access to an AWS S3 bucket, and specifically to the S3 bucket(s) from which you pull workflow inputs. Refer to [SciWiki](https://sciwiki.fredhutch.org/scicomputing/compute_cloud/#get-aws-credentials) or email scicomp to request credentials.

As of version 1.3 if you have credentials, then the Cromwell server will be configured to allow input files to directly specified using their AWS S3 url. However if you do not have AWS credentials, then the server will now successfully start up, simply without the ability to localize files from S3, thus all test workflows that use files in S3 will not work for you, but everything else should.

### 1.5.3 Database Setup (one time)

These instructions let you stand up a Cromwell server for 7 days at a time. If you have workflows that run longer than that or you want to be able to get metadata for or restart jobs even after the server goes down, you’ll want an external database to keep track of your progress even if your server goes down (for whatever reason). It also will allow your future workflows to use cached copies of data when the exact task has already been done (and recorded in the database). We have found as well that by using a MySQL database for your Cromwell server, it will run faster and be better able to handle simultaneous workflows while also making all the metadata available to you during and after the run.

We currently suggest you go to [DB4Sci](https://mydb.fredhutch.org/login) and see the Wiki entry for DB4Sci [here](https://sciwiki.fredhutch.org/scicomputing/store_databases/#db4sci--previously-mydb). There, you will login using Fred Hutch credentials, choose Create DB Container, and choose the MariaDB option. The default database container values are typically fine, EXCEPT you likely need either weekly or no backups (no backups preferred) for this database. Save the DB/Container Name, DB Username and DB Password as you will need them for the configuration step. Once you click submit, a confirmation screen will appear (hopefully), and you’ll need to note which Port is specified. This is a 5 digit number currently.

Once this is complete, you can file a ticket to scicomp to request the database to be set up in that container. In the future it might be possible to have this step be part of the DB4Sci setup process but until then, you’ll need to do this additional step. Or if you are able to get onto rhino, you can do the following to do this process yourself:

ml MariaDB/10.5.1-foss-2019b  
mysql --host mydb --port <Port> --user <username> --password

It will then prompt you to enter the DB password you specified during setup. Once you are are a “mysql>” prompt, you can do the following. >Note, we suggest you name the database inside the container the same as the container, but you cannot include dashes in your database name. In the future, DB4Sci may also set up the database inside the container for you, in which case you would be provided a database name as well during setup.

MariaDB [(none)]> create database <DB Name>;  
# It should do it's magic  
MariaDB [(none)]> exit

Then you’re ready to go and never have to set up the database part again and you can use this database to manage all your work over time.

## 1.6 Server setup instructions

1. Decide where you want to keep your Cromwell configuration files. This must be a place where rhino can access them, such as in your Home directory, which is typically the default directory when you connect to the rhinos. We suggest you create a cromwell-home folder (or whatever you want to call it) and follow these git instructions to clone it directly.
2. First set up the customizations per user that you’re going to want for your server(s) by making user configuration file(s) in your cromwell-home or wherever you find convenient. You can manage mulitple Cromwell profiles this way by just maintaining different files full of credentials and configuration variables that you want.

To get started, do the following on rhino:

mkdir -p cromwell-home  
cd cromwell-home  
git clone --branch main https://github.com/FredHutch/diy-cromwell-server.git  
  
  
cp ./diy-cromwell-server/cromUserConfig.txt .  
## When you are first setting up Cromwell, you'll need to put all of your User Customizations into this `cromUserConfig.txt` which can serve as a template.   
## After you've done this once, you just need to keep the path to the file(s) handy for the future.

1. Tailor your cromUserConfig.txt file to be specific to your various directories and resources (see notes in the version of the file in this repo).

Note: For this server, you will want multiple cores to allow it to multi-task. Memory is less important when you use an external database. If you notice issues, the particular resource request for the server job itself might be a good place to start adjusting, in conjunction with some guidance from SciComp or the Slack [Question and Answer channel](https://fhbig.slack.com/archives/CD3HGJHJT) folks.

1. Kick off your Cromwell server: > Note: for version 1.2 and later, this script, cromwell.sh includes the version name in it, such as cromwellv1.2.sh.

## You'll want to put `cromwellv1.3.sh` somewhere handy for future use, we suggest:  
cp ./diy-cromwell-server/cromwellv1.3.sh .  
chmod +x cromwellv1.3.sh  
  
# Then simply start up Cromwell:  
./cromwellv1.3.sh cromUserConfig.txt

1. Much like the grabnode command you may have used previously, the script will run and print back to the console instructions once the resources have been provisioned for the server. You should see something like this:

Your configuration details have been found...  
Getting an updated copy of Cromwell configs from GitHub...  
Setting up all required directories...  
Detecting existence of AWS credentials...  
Credentials found, setting appropriate configuration...  
Requesting resources from SLURM for your server...  
Submitted batch job 50205062  
Your Cromwell server is attempting to start up on \*\*node/port gizmok30:2020\*\*. If you encounter errors, you may want to check your server logs at /home/username/cromwell-home/server-logs to see if Cromwell was unable to start up.  
Go have fun now.

NOTE: Please write down the node and port it specifies here. This is the only place where you will be able to find the particular node/port for this instance of your Cromwell server, and you’ll need that to be able to send jobs to the Crowmell server. If you forget it, scancel the Cromwell server job and start a new one.

1. This node host and port is what you use to submit and monitor workflows with the Shiny app at [cromwellapp.fredhutch.org](https://cromwellapp.fredhutch.org/). After you click the “Connect to Server” button, you’ll put gizmok30:20201 (or whatever your node:port is) where it says “Current Cromwell host:port”.
2. While your server will normally stop after 7 days (the default), at which point if you have jobs still running you can simply restart your server and it will reconnect to existing jobs/workflows. However, if you need to take down your server for whatever reason before that point, you can go to rhino and do:

# Here `username` is your Fred Hutch username  
squeue -u username  
## Or if you want to get fancy:  
squeue -o '%.18i %.9P %j %.8T %.10M %.9l %.6C %R' -u username  
  
## You'll see a jobname "cromwellServer". Next to that will be a JOBID. In this example the JOBID of the server is 50062886.  
  
scancel 50062886

## 1.7 Cromwell Server Customization

In cromUserConfig.txt there are some variables that allow users to share a similar configuration file but tailor the particular behavior of their Cromwell server to best suit them. The following text is also in this repo but these are the customizations you’ll need to decide on for your server.

################## WORKING DIRECTORY AND PATH CUSTOMIZATIONS ###################  
## Where do you want the working directory to be for Cromwell (note: this process will create a subdirectory here called "cromwell-executions")?   
### Suggestion: /fh/scratch/delete90/pilastname\_f/username/  
SCRATCHDIR=/fh/scratch/delete90/...  
  
## Where do you want logs about individual workflows (not jobs) to be written?  
## Note: this is a default for the server and can be overwritten for a given workflow in workflow-options.  
### Suggestion: /fh/fast/pilastname\_f/cromwell/workflow-logs  
WORKFLOWLOGDIR=~/cromwell-home/workflow-logs  
  
## Where do you want to save Cromwell server logs for troubleshooting Cromwell itself?  
### Suggestion: ~/home/username/cromwell-home/server-logs  
SERVERLOGDIR=~./cromwell-home/server-logs  
  
################ DATABASE CUSTOMIZATIONS #################  
## DB4Sci MariaDB details (remove < and >, and use unquoted text):  
  
CROMWELLDBPORT=...  
CROMWELLDBNAME=...  
CROMWELLDBUSERNAME=...  
CROMWELLDBPASSWORD=...  
  
## Number of cores for your Cromwell server itself - usually 4 is sufficient.   
###Increase if you want to run many, complex workflows simultaneously or notice your server is slowing down.  
NCORES=4  
  
## Length of time you want the server to run for.   
### Note: when servers go down, all jobs they'd sent will continue. When you start up a server the next time  
### using the same database, the new server will pick up whereever the previous workflows left off. "7-0" is 7 days, zero hours.  
SERVERTIME="7-0"

Contact Amy Paguirigan about these issues for some advice or file an issue on this repo.

# 2 FAQ and Reference

## 2.1 Test Workflows

1. See our [Test Workflow folder](https://github.com/FredHutch/diy-cromwell-server/tree/main/testWorkflows) once your server is up and run through the tests specified in the markdown there. > NOTE: For those test workflows that use Docker containers, know that the first time you run them, you may notice that jobs aren’t being sent very quickly. That is because for our cluster, we need to convert those Docker containers to something that can be run by Singularity. The first time a Docker container is used, it must be converted, but in the future Cromwell will used the cached version of the Docker container and jobs will be submitted more quickly.

## 2.2 Fred Hutch Customization

### 2.2.1 Task Defaults and Runtime Variables

For the gizmo backend, the following runtime variables are available that are customized to our configuration. What is specified below is the current default as written, you can edit these in the config file if you’d like OR you can specify these variables in your runtime block in each task to change only the variables you want to change from the default for that particular task.

* cpu = 1
  + An integer number of cpus you want for the task
* walltime = "18:00:00"
  + A string of date/time that specifies how many hours/days you want to request for the task
* memory = 2000
  + An integer number of MB of memory you want to use for the task
* partition = "campus-new"
  + Which partition you want to use, the default is campus-new but whatever is in the runtime block of your WDL will overrride this.
* modules = ""
  + A space-separated list of the environment modules you’d like to have loaded (in that order) prior to running the task.
* docker = "ubuntu:latest"
  + A specific Docker container to use for the task. For the custom Hutch configuration, docker containers can be specified and the necessary conversions (to Singularity) will be performed by Cromwell (not the user). Note: when docker is used, soft links cannot be used in our filesystem, so workflows using very large datasets may run slightly slower due to the need for Cromwell to copy files rather than link to them.
* dockerSL= "ubuntu:latest"
  + This is a custom configuration for the Hutch that allows users to use docker and softlinks only to specific locations in Scratch. It is helpful when working with very large files.

## 2.3 Guidance and Support

### 2.3.1 Monitoring your workflows at Fred Hutch:

I made a shiny app you can use to monitor your own Cromwell server workflows when you have a Cromwell server running on gizmo that can be found [here](https://cromwellapp.fredhutch.org/). If you’d like to roll your own, you can find my shiny app code [here](https://github.com/FredHutch/shiny-cromwell).

### 2.3.2 Design Recommendations for WDL workflows at Fred Hutch

See our [SciWiki page](https://sciwiki.fredhutch.org/compdemos/Cromwell/) on Cromwell for more about guidance for how to start structuring and building your workflows as well as how to share them with others on campus in a findable way.

### 2.3.3 R support

I have a basic R package that wraps the Cromwell API allowing you to submit, monitor and kill workflow jobs on gizmo from R directly. The package is [fh.wdlR](https://github.com/FredHutch/fh.wdlR).

### 2.3.4 Other Fred Hutch based resources

While additional development is going on to make Cromwell work better in AWS (currently it works well in Google and SLURM among other backends), we are anticipating that it will be more widely available for use with AWS based computing. To support that there is a growing public data set AWS S3 bucket at fh-ctr-public-reference-data. Contact Amy Paguirigan if you’d like something to be added here and we can help you do that.

# About the Authors

These credits are based on our [course contributors table guidelines](https://www.ottrproject.org/more_features.html#giving-credits-to-contributors).

| Credits | Names |
| --- | --- |
| **Pedagogy** |  |
| Lead Content Instructor(s) | [FirstName LastName](link%20to%20personal%20website) |
| Lecturer(s) (include chapter name/link in parentheses if only for specific chapters) - make new line if more than one chapter involved | Delivered the course in some way - video or audio |
| Content Author(s) (include chapter name/link in parentheses if only for specific chapters) - make new line if more than one chapter involved | If any other authors besides lead instructor |
| Content Contributor(s) (include section name/link in parentheses) - make new line if more than one section involved | Wrote less than a chapter |
| Content Editor(s)/Reviewer(s) | Checked your content |
| Content Director(s) | Helped guide the content direction |
| Content Consultants (include chapter name/link in parentheses or word “General”) - make new line if more than one chapter involved | Gave high level advice on content |
| Acknowledgments | Gave small assistance to content but not to the level of consulting |
| **Production** |  |
| Content Publisher(s) | Helped with publishing platform |
| Content Publishing Reviewer(s) | Reviewed overall content and aesthetics on publishing platform |
| **Technical** |  |
| Course Publishing Engineer(s) | Helped with the code for the technical aspects related to the specific course generation |
| Template Publishing Engineers | [Candace Savonen](https://www.cansavvy.com/), [Carrie Wright](https://carriewright11.github.io/) |
| Publishing Maintenance Engineer | [Candace Savonen](https://www.cansavvy.com/) |
| Technical Publishing Stylists | [Carrie Wright](https://carriewright11.github.io/), [Candace Savonen](https://www.cansavvy.com/) |
| Package Developers ([ottrpal](https://github.com/jhudsl/ottrpal)) [Candace Savonen](https://www.cansavvy.com/), [John Muschelli](https://johnmuschelli.com/), [Carrie Wright](https://carriewright11.github.io/) |  |
| **Art and Design** |  |
| Illustrator(s) | Created graphics for the course |
| Figure Artist(s) | Created figures/plots for course |
| Videographer(s) | Filmed videos |
| Videography Editor(s) | Edited film |
| Audiographer(s) | Recorded audio |
| Audiography Editor(s) | Edited audio recordings |
| **Funding** |  |
| Funder(s) | Institution/individual who funded course including grant number |
| Funding Staff | Staff members who help with funding |

## ─ Session info ───────────────────────────────────────────────────────────────  
## setting value   
## version R version 4.0.2 (2020-06-22)  
## os Ubuntu 20.04.3 LTS   
## system x86\_64, linux-gnu   
## ui X11   
## language (EN)   
## collate en\_US.UTF-8   
## ctype en\_US.UTF-8   
## tz Etc/UTC   
## date 2022-10-18   
##   
## ─ Packages ───────────────────────────────────────────────────────────────────  
## package \* version date lib source   
## assertthat 0.2.1 2019-03-21 [1] RSPM (R 4.0.3)   
## bookdown 0.24 2022-02-15 [1] Github (rstudio/bookdown@88bc4ea)   
## callr 3.4.4 2020-09-07 [1] RSPM (R 4.0.2)   
## cli 2.0.2 2020-02-28 [1] RSPM (R 4.0.0)   
## crayon 1.3.4 2017-09-16 [1] RSPM (R 4.0.0)   
## desc 1.2.0 2018-05-01 [1] RSPM (R 4.0.3)   
## devtools 2.3.2 2020-09-18 [1] RSPM (R 4.0.3)   
## digest 0.6.25 2020-02-23 [1] RSPM (R 4.0.0)   
## ellipsis 0.3.1 2020-05-15 [1] RSPM (R 4.0.3)   
## evaluate 0.14 2019-05-28 [1] RSPM (R 4.0.3)   
## fansi 0.4.1 2020-01-08 [1] RSPM (R 4.0.0)   
## fs 1.5.0 2020-07-31 [1] RSPM (R 4.0.3)   
## glue 1.6.1 2022-01-22 [1] CRAN (R 4.0.2)   
## htmltools 0.5.0 2020-06-16 [1] RSPM (R 4.0.1)   
## knitr 1.33 2022-02-15 [1] Github (yihui/knitr@a1052d1)   
## lifecycle 1.0.0 2021-02-15 [1] CRAN (R 4.0.2)   
## magrittr 2.0.2 2022-01-26 [1] CRAN (R 4.0.2)   
## memoise 1.1.0 2017-04-21 [1] RSPM (R 4.0.0)   
## pkgbuild 1.1.0 2020-07-13 [1] RSPM (R 4.0.2)   
## pkgload 1.1.0 2020-05-29 [1] RSPM (R 4.0.3)   
## prettyunits 1.1.1 2020-01-24 [1] RSPM (R 4.0.3)   
## processx 3.4.4 2020-09-03 [1] RSPM (R 4.0.2)   
## ps 1.3.4 2020-08-11 [1] RSPM (R 4.0.2)   
## purrr 0.3.4 2020-04-17 [1] RSPM (R 4.0.3)   
## R6 2.4.1 2019-11-12 [1] RSPM (R 4.0.0)   
## remotes 2.2.0 2020-07-21 [1] RSPM (R 4.0.3)   
## rlang 0.4.10 2022-02-15 [1] Github (r-lib/rlang@f0c9be5)   
## rmarkdown 2.10 2022-02-15 [1] Github (rstudio/rmarkdown@02d3c25)  
## rprojroot 2.0.2 2020-11-15 [1] CRAN (R 4.0.2)   
## sessioninfo 1.1.1 2018-11-05 [1] RSPM (R 4.0.3)   
## stringi 1.5.3 2020-09-09 [1] RSPM (R 4.0.3)   
## stringr 1.4.0 2019-02-10 [1] RSPM (R 4.0.3)   
## testthat 3.0.1 2022-02-15 [1] Github (R-lib/testthat@e99155a)   
## usethis 2.1.5.9000 2022-02-15 [1] Github (r-lib/usethis@57b109a)   
## withr 2.3.0 2020-09-22 [1] RSPM (R 4.0.2)   
## xfun 0.26 2022-02-15 [1] Github (yihui/xfun@74c2a66)   
## yaml 2.2.1 2020-02-01 [1] RSPM (R 4.0.3)   
##   
## [1] /usr/local/lib/R/site-library  
## [2] /usr/local/lib/R/library

# 3 References