

# JuliaPro

(v1.5.4-1)

## Installation Manual and Quickstart Guide

### Contents

1. *Objective*
2. *Prerequisites*
  - 2.1. *System Library Prerequisites*
    - 2.1.1 *Prerequisites for Installation on CentOS 7*
    - 2.1.2 *Prerequisites for Installation on Ubuntu*
3. *Installing JuliaPro*
4. *Using JuliaPro in a GUI environment*
  - 4.1. *Launching JuliaPro*
  - 4.2. *Getting Started with JuliaPro*
5. *Using JuliaPro in a non-GUI environment*
6. *Cloud computing with JuliaPro*
7. *Uninstalling JuliaPro*

## 1. Objective

This guide details the installation procedure and usage of the base JuliaPro package and JuliaPro's Juno IDE.

## 2. Prerequisites

- `JuliaPro-1.5.4-1.sh` file (This file can be downloaded from <https://juliacomputing.com/> )
- An installation of CentOS 7, Ubuntu 14.04, or Ubuntu 16.04, RHEL 7
- 5 GB of disk space
- Installation of required system libraries using Administrator or sudo privileges
- Active Internet connection

- LinkedIn or Gmail or GitHub or JuliaComputing account (Either one of these accounts are required for authentication)
- `token.toml` , this file is required only if you're using a Linux machine that does not have GUI, instructions to download this file can be found in following section `Using JuliaPro in a non-GUI environment`

## 2.1 System Library Prerequisites

### 2.1.1 Prerequisites for Installation on CentOS 7

Following libraries are required only if you're installing JuliaPro on an environment which has GUI , if your Linux machine does not have a GUI environment or if you're not planning on using the JuliaPro IDE, then you don't need to install any of these prerequisites.

```
xclip  
libXScrnSaver
```

These libraries need to be installed by a user with appropriate Administrator or sudo privileges using the built-in yum package manager.

```
sudo yum -y install xclip  
sudo yum -y install libXScrnSaver
```

### 2.1.2 Prerequisites for Installation on Ubuntu

Following libraries are required only if you're installing JuliaPro on an environment which has GUI , if your Linux machine does not have a GUI environment or if you're not planning on using the JuliaPro IDE, then you don't need to install any of these prerequisites.

```
xclip  
libgconf-2-4
```

This library need to be installed by a user with appropriate Administrator or sudo privileges using the built-in apt package manager.

```
sudo apt-get -y install xclip  
sudo apt-get -y install libgconf-2-4
```

## 3. Installing JuliaPro

Once the system requirements are met, you can start the JuliaPro installation using the installation script.

To execute the `JuliaPro-1.5.4-1.sh` script, you might first need to change the execution permissions on the script. Use following command to change the execution permission

```
chmod 777 JuliaPro-1.5.4-1.sh
```

The `JuliaPro-1.5.4-1.sh` script takes one argument; The absolute path to the directory into which you wish to install JuliaPro.

Example:-

```
./JuliaPro-1.5.4-1.sh /home/julia/
```

Immediately after the execution of JuliaPro script, you will be prompted with following question

```
Do you want to configure your JuliaPro to work with your private JuliaTeam package server?
```

This question is applicable only for JuliaTeam customers, if you're an individual or your company or organization is not using JuliaTeam, then you can answer "No" and continue with the installation. If you press "No", the default package server `pkg.juliahub.com` will be used for all package and registry operations. You can download (Or update) packages and registries from this server even if you're not a JuliaTeam customer.

If your organization is using JuliaTeam, then you can configure JuliaPro to work with your private package server (JuliaTeam enables you to create private package server) by answering "Yes". Subsequently, you will be prompted to enter your private package server URL with following question.

```
Please enter your JuliaTeam package server URL through which JuliaPro should download packages and registries
```

Once you enter your private JuliaTeam package server URL, the installer will configure JuliaPro to download all the packages and registries from your private JuliaTeam server.

Upon completion of the installer, the contents of the JuliaPro installation directory can be viewed.

## 4. Using JuliaPro in a GUI environment

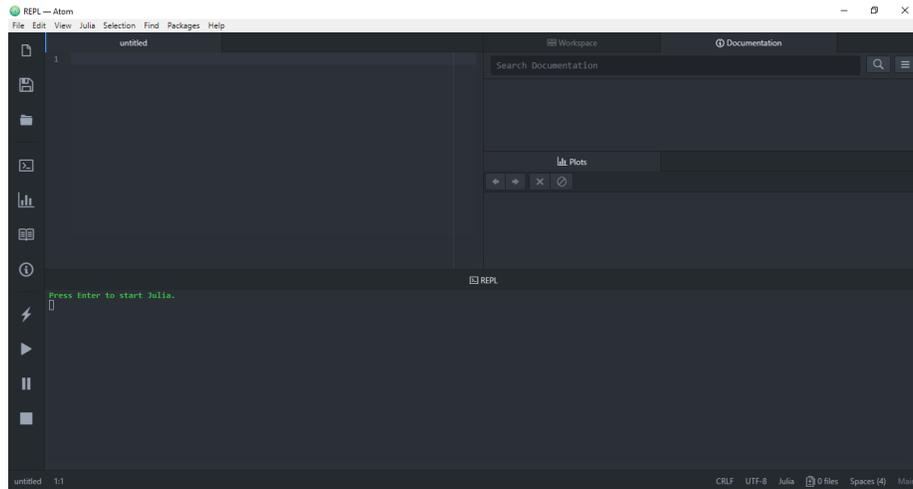
### 4.1 Launching JuliaPro

JuliaPro can be launched from the JuliaPro installation directory as shown below:

```
$ cd <juliapro>/JuliaPro-1.5.4-1
$ ./Launch_JuliaPro
```

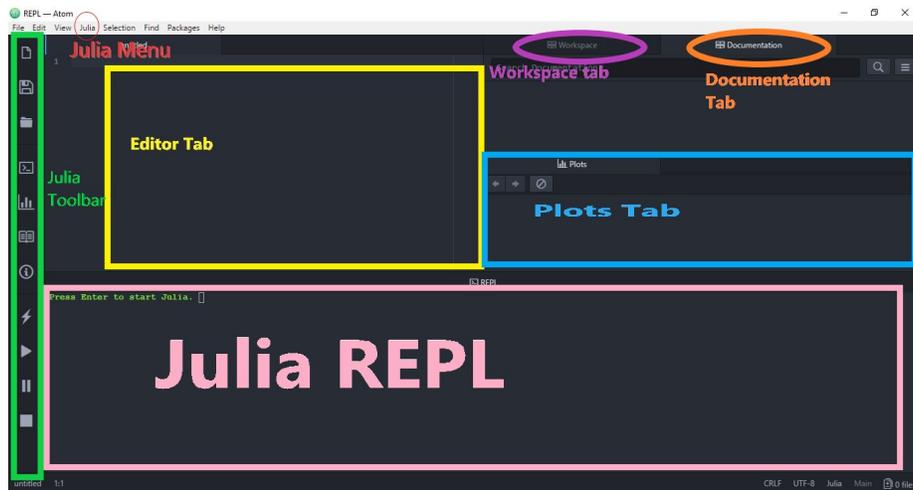
Where `<juliapro>` has to be replaced with the JuliaPro installation path.

Upon initially launching Juno, you will be presented with the following window.

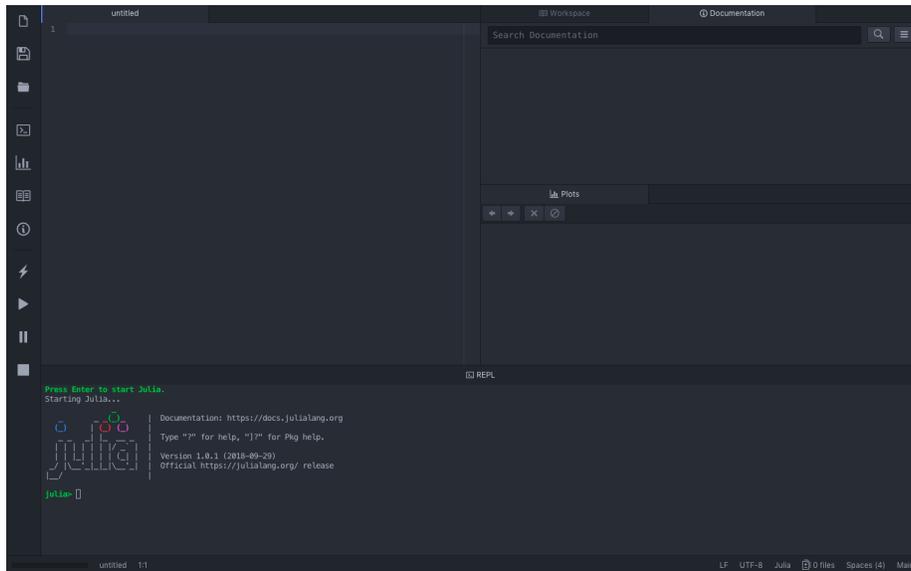


## 4.2 Getting Started with JuliaPro

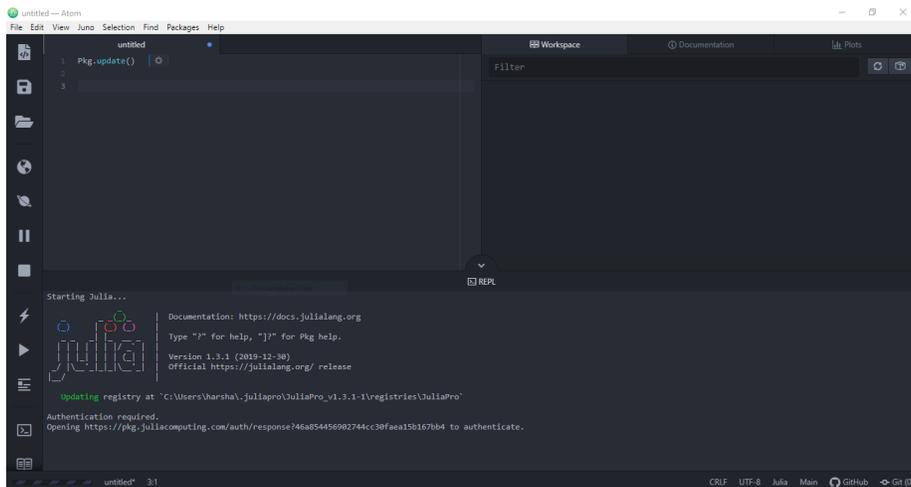
Naming convention for different layout (Panels) in Juno IDE can be seen below



You can start Julia by pressing enter in “Julia REPL” pane



By default, JuliaPro will download all the packages from `pkg.juliahub.com`, this website requires authentication, hence, you will go through the authentication process the first time you perform a `Pkg` operation; JuliaPro will open the authentication link in your default browser, once you successfully authenticate in your browser, JuliaPro will download a unique token for you and place it in the appropriate location. This token will be reused in your subsequent `Pkg` operations.

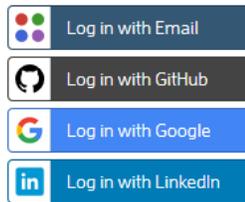


Once the URL link is open in your browser, you will be directed to authenticate by signing into either one of these accounts : LinkedIn or Gmail or GitHub or JuliaComputing account.



## JuliaTeam Authentication

Please proceed to authenticate yourself for package operations.



Once the authentication is done, you will get a message in browser saying “Authenticated. You may close this browser window now.”

**Authenticated. You may close this browser window now.**

Your Pkg operation should continue as is, all your subsequent Pkg operations will use the token that JuliaPro just downloaded.

```
Starting Julia...
Documentation: https://docs.julialang.org
Type "?" for help, "]" for Pkg help.
Version 1.3.1 (2019-12-30)
Official https://julialang.org/ release

Updating registry at `C:\Users\harsha\.juliapro\JuliaPro_v1.3.1-1\registries\JuliaPro`
Authentication required.
Opening https://pkg.juliacomputing.com/auth/response?46a854456902744cc30faea15b167bb4 to authenticate.
Authentication successful.
```

If the automated browser authentication is failing, please use the instructions in following section to manually download the token and place it in the appropriate location: “Using JuliaPro in a non-GUI environment”

## 5. Using JuliaPro in a non-GUI environment

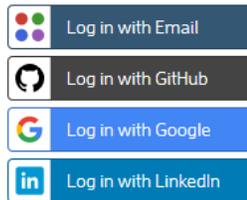
By default, JuliaPro will download all the packages from `pkg.juliahub.com`, this website requires authentication, hence, you have to download `token.toml` file to authenticate any requests from your JuliaPro installation to the server. This file can be downloaded by visiting following website in your browser `https://pkg.juliahub.com/auth`

Once you open the URL link, you will be directed to authenticate by signing into either one of these accounts : LinkedIn or Gmail or GitHub or JuliaComputing account.



### JuliaTeam Authentication

Please proceed to authenticate yourself for package operations.



Once the authentication is done, your `token.toml` file download should begin immediately, if your download doesn't begin automatically, you can always click on "here" URL link to download the token

← → 🔄 <https://pkg.juliacomputing.com/auth/download/token.toml>

#### Authenticated to JuliaTeam

Your token file should begin downloading now.  
Click [here](#) if it does not.

Once you have `token.toml`, rename this file as `auth.toml` and then you can move this file to the machine where you have installed JuliaPro, the default

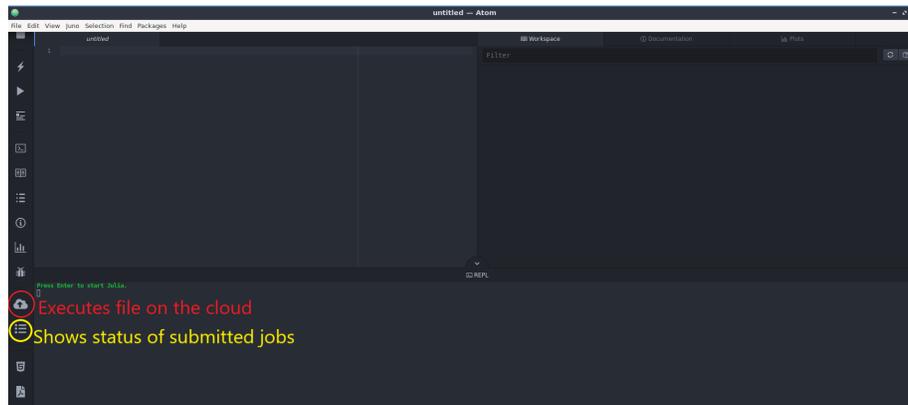
location to place this file is `~/.julia/servers/pkg.juliahub.com/auth.toml` i.e create following path `.julia/servers/pkg.juliahub.com/` in your home directory and place `auth.toml` in the path you just created. Once you place the file in this location, all subsequent `Pkg` operations will use this token to authenticate the download from `pkg.juliahub.com`

## 6. Cloud computing with JuliaPro

JuliaPro has built-in support to submit jobs to JuliaHub or your private JuliaRun instance, you can bring up cloud compute UI; either by executing following commands in Atom Command Palette (Press `Ctrl+Shift+P` while focused in an editor pane to see command palette pop up) or by using the cloud compute buttons in the Julia toolbar.

- Julia Run: Run File
- Julia Run: Show Jobs

The cloud compute buttons are located at the bottom of the Julia toolbar, so you need to place the mouse pointer on the toolbar and scroll down to the bottom to locate the cloud compute buttons, relevant cloud compute buttons are shown in below screenshot (Circled in Red and Yellow along with the description).



Following prerequisites need to be satisfied before using the cloud compute feature in JuliaPro

1. You need a valid token for the server to which you're submitting the job, if a valid token is not found; then cloud compute module will trigger the authentication mechanism to download a valid token from the server. In case of JuliaHub, a browser Window will open up requesting you to login using one of the following authentication mechanisms.



---

## JuliaTeam Authentication

---

Please proceed to authenticate yourself for package operations.



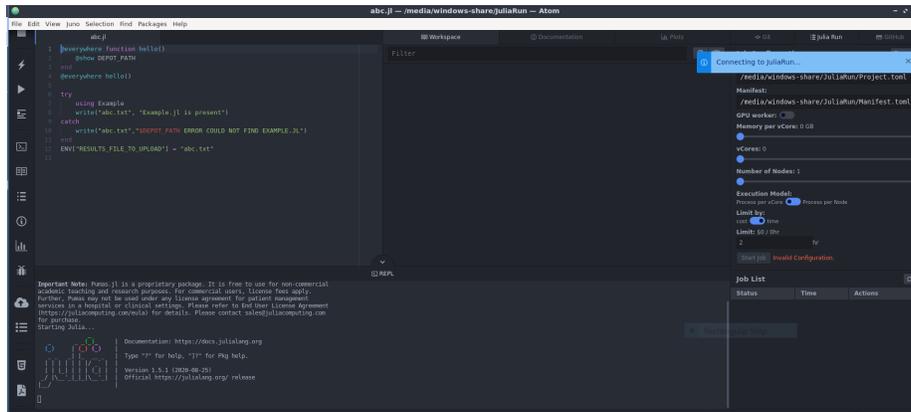
Once you successfully login, you should see following message in your browser, this message indicates the token download was successful, you can return back to the IDE and wait until the cloud compute module connects to the server.

**Authenticated. You may close this browser window now.**

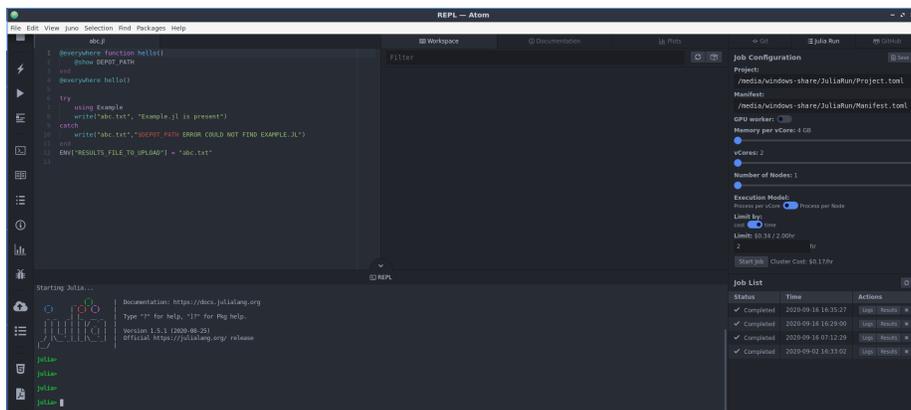
2. If you're submitting jobs to JuliaHub; then make sure you have enough credits to run the job, if you don't have any JuliaHub cloud compute credits; then you can always login to JuliaHub website <https://juliahub.com> and provide payment details to which the cloud compute charges will be billed. If you're submitting jobs to your private JuliaRun instance; then make sure the account associated with `~/.julia/servers/<Your-domain-name>/auth.toml` is authorized to run the jobs on your JuliaRun instance.
3. You should be using JuliaPro v1.5.4-1 or higher

Once all the prerequisites are satisfied, you can follow these steps to start submitting jobs to your server

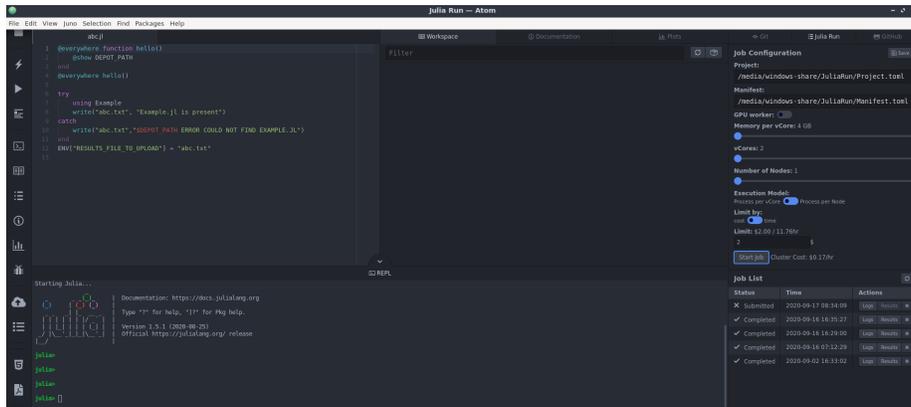
1. Open the project (In JuliaPro IDE) that requires to be ran on the cloud instance
2. Click on the "Show JuliaRun jobs" button from the Julia toolbar or execute following command in Command Palette: **Julia Run: Show Jobs**, this should open up the cloud compute UI, you should also see a message saying "Connecting to JuliaRun"



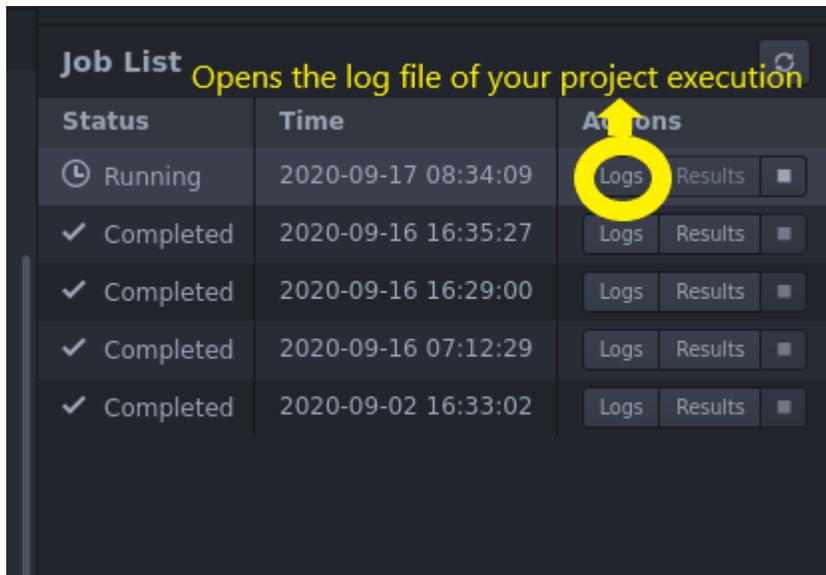
3. Wait until the cloud compute module connects to your server, if all the prerequisites are satisfied, you should see the “Start job” button enabled once the connection is established, you should also be able to see all the past jobs that you submitted (if you have any).



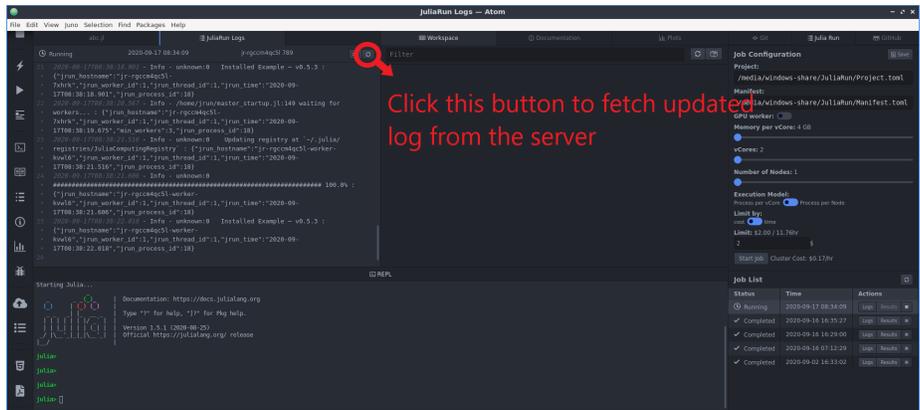
4. Select the appropriate configuration for the workers over which your project will be ran, if you’re using JuliaHub; then make sure to limit your expenditure for the current job, either based on time or by Dollars using “Limit by” toggle button, the estimate cost per hour is displayed in the same configuration window, this cost is relative to the worker configuration you selected. For more details regarding CPU and GPU configurations, please refer to following documentation: <https://docs.juliahub.com/>
5. Click on “Start job” button to execute your project on the cloud instance, you should see a popup in your IDE saying “Job submitted”, your job should also appear in the “Job list”



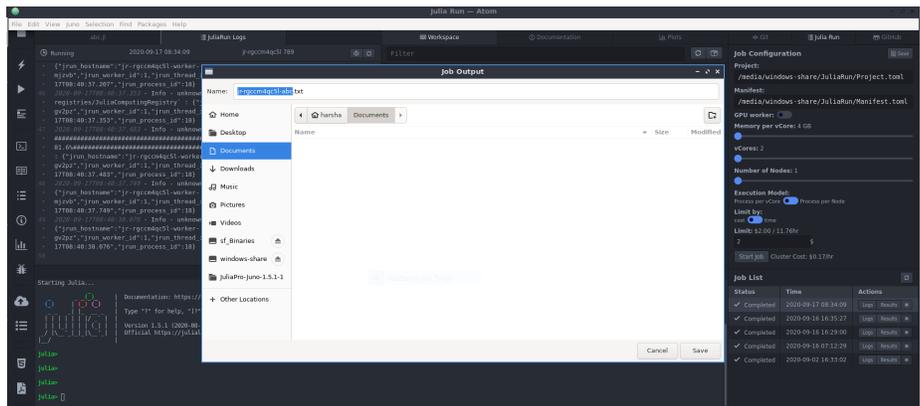
- Once the status of your job changes from “Submitted” to “Running”, you can click on “Logs” button next to your job to see the progress of your project execution.



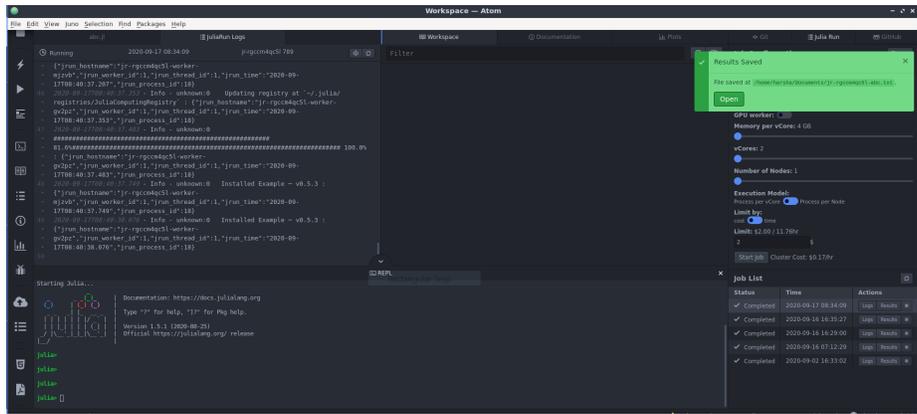
- A log file should open up in log viewer, these logs are not “live”, you need to hit the refresh button in order to fetch the latest logs from the server.



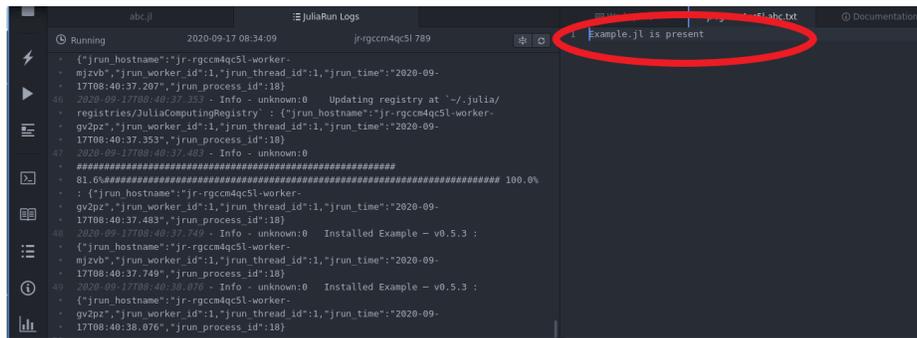
- Once your project execution completes, the results button will get enabled (If your script stored results in a file) and the job status should now say “Completed”, you can click on the “Results” button to download your results from the server. An explorer Window should open up as soon as you click on the “Results” button, you can navigate and store the results file in an appropriate location.



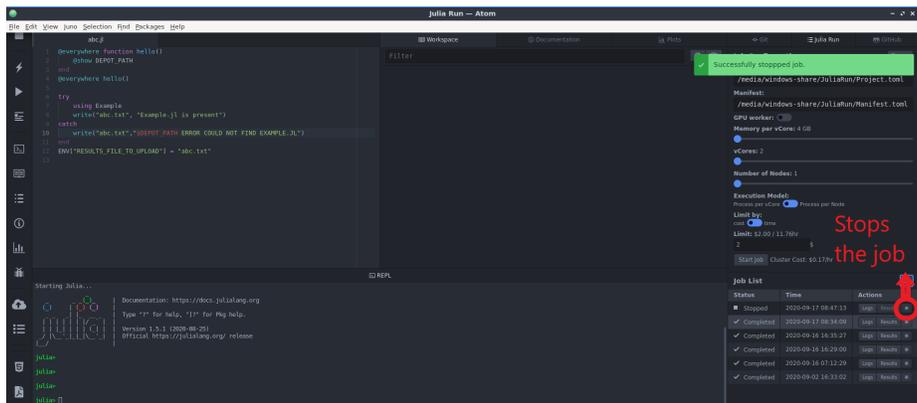
- Once the result file download completes, you should see a popup confirming the same



10. You can choose to open the results file in Atom IDE or any other software to validate it.



11. You can interrupt a “Running” job by clicking on the “Stop” button, this button is enabled only if the job status is “Running”, you should also see a popup that confirms your job was interrupted successfully and job status on interrupted jobs should be “Stopped”



Below source code was used in above example

```

@everywhere function hello()
    @show DEPOT_PATH
end
@everywhere hello()

try
    using Example
    write("abc.txt", "Example.jl is present")
catch
    write("abc.txt", "$DEPOT_PATH ERROR COULD NOT FIND EXAMPLE.JL")
end
ENV["RESULTS_FILE_TO_UPLOAD"] = "abc.txt"

Contents of Project.toml and Manifest.toml in the example project

[deps]
Example = "7876af07-990d-54b4-ab0e-23690620f79a"

# This file is machine-generated - editing it directly is not advised

[[Example]]
git-tree-sha1 = "46e44e869b4d90b96bd8ed1fdc32244fddfb6cc"
uuid = "7876af07-990d-54b4-ab0e-23690620f79a"
version = "0.5.3"

```

## 7. Uninstalling JuliaPro

Most of the JuliaPro contents are present in the installation directory, so removing JuliaPro installation directory essentially uninstalls JuliaPro from your Linux system.

JuliaPro stores end-user profiles outside the installation directory to ease the upgrade process, if you're not planning on reusing the contents of these folders, then you can remove following directories manually:

- Julia packages that are located in following path: `~/.julia`
- Julia kernel located in following path: `~/.local/share/jupyter/kernels/juliapro_v1.5.4-1-1.5`
- Atom cache folder in following location: `~/.config/Atom/`
- Atom package directories: `~/.julia_atom_1.5.4-1` and `~/.atom`

NOTE:- Other applications might be using above directories, please be cautious while removing above folders.