Tutorial: Juliaunning Julia in Galileo
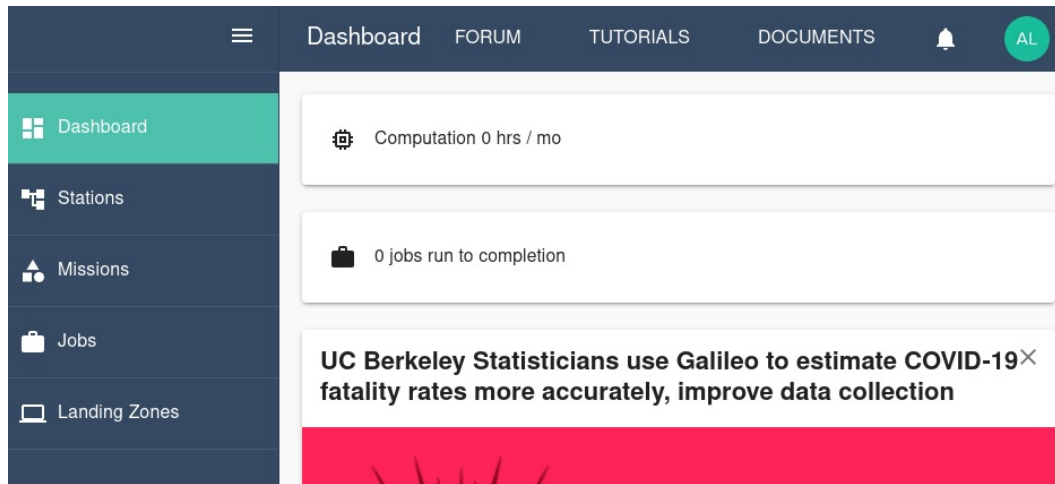
Gettting started with Julia in Galileo

To get started with Galileo, log into your account using Firefox or Chrome.

Understanding the user interface and cloning a Mission

When you log into Galileo, the first thing you'll see is your Dashboard:

View of the Galileo Dashboard

To run the Julia example, start by navigating to the Missions tab using the side menu. Clone the Julia Batch example Mission from the Explore Missions tab. Use the filter to search for the mission by name and click "Apply".
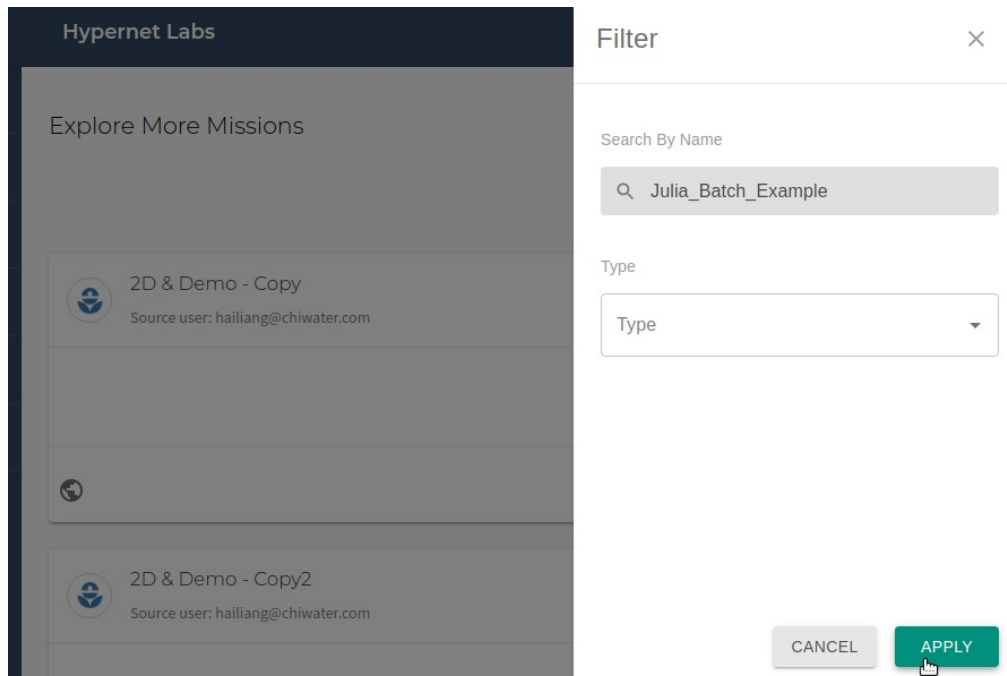
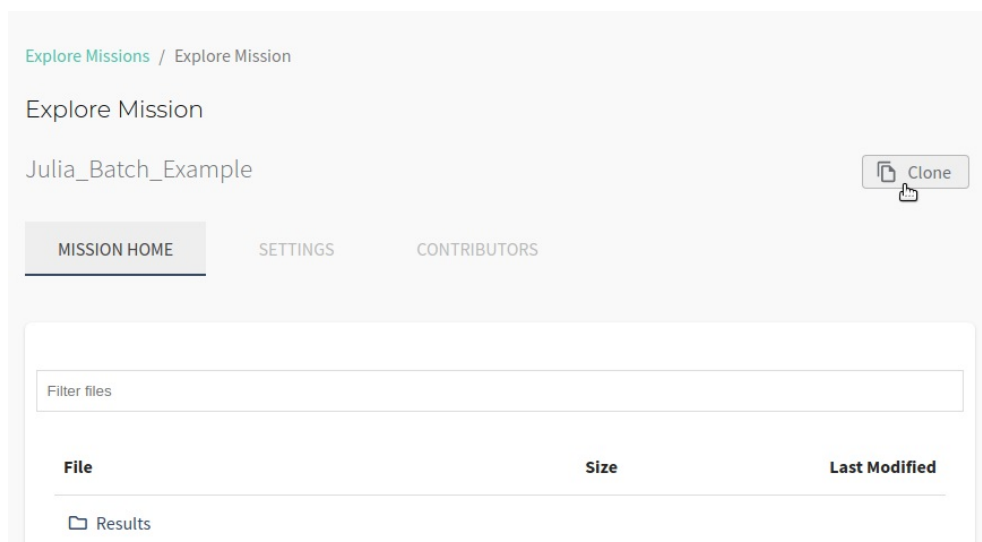Find the public example mission by name

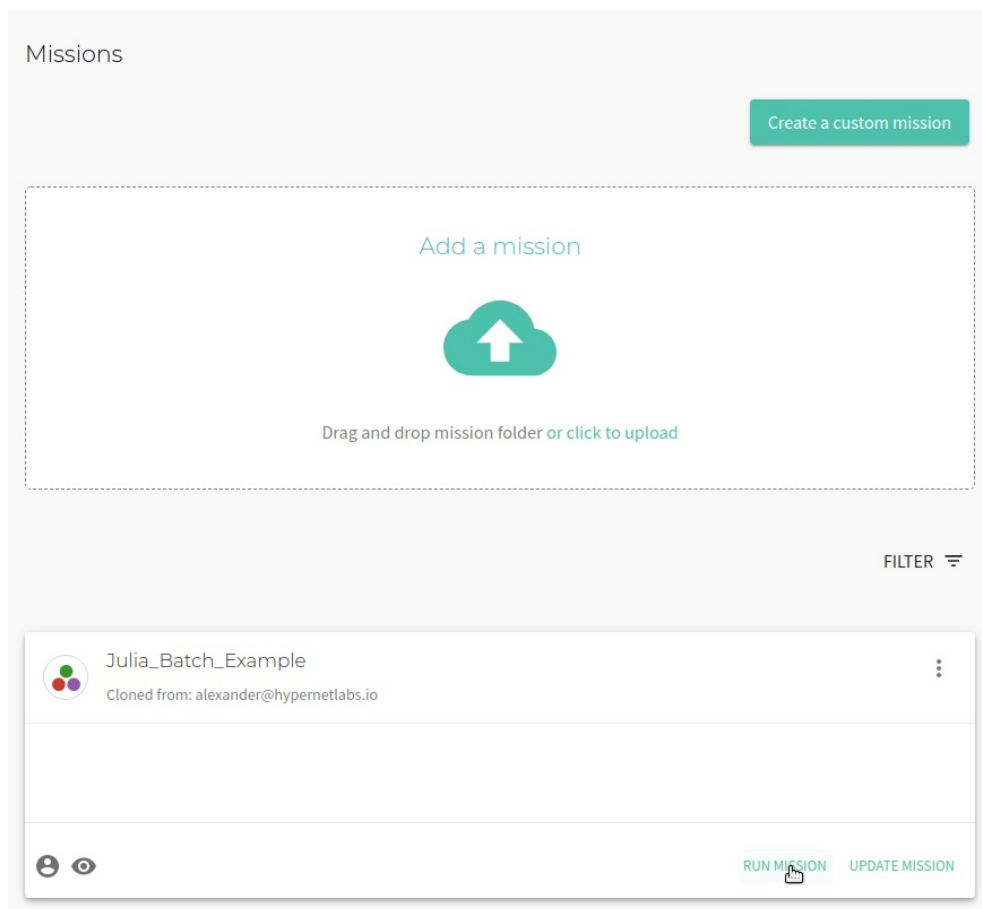Once you have found the correct Mission, click "View Mission".
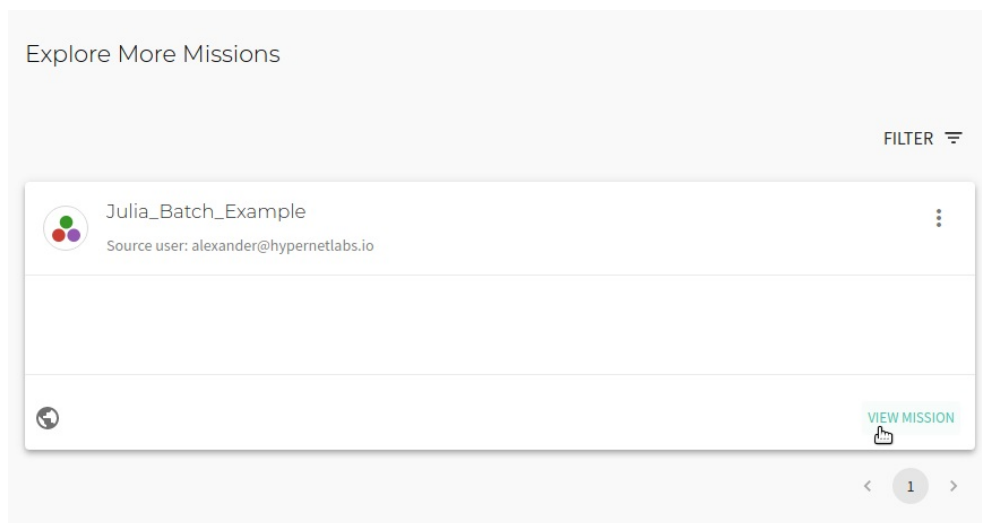
Click View Mission

To clone the public Mission to your account, click the "Clone" button in the upper right corner of the interface. Choose between creating a public or private clone and also choose which Cargo Bay to use.

Clone the mission

You will now see a cloned copy of the Mission in your Missions.

The cloned copy

Let's take a look at our files

The julia_example.jl script conducts a simple linear regression using the supplied mtcars.csv dataset and makes a simple plot. It also demonstrates how to use a dataset loaded from a library.

Next, our julia_example.jl file conducts a Monte Carlo experiment that simulates 50,000 throws of two six-sided dice to calculate the probability that the sum of one throw of two dice is greater than or equal to seven. It then repeats the same experiment 10 million times. Finally it compares the means of the two samples and the amount of time it took to calculate them.

Running a job and collecting results

Now we are ready to run a job using the Mission. Click the **Run** button in the upper right corner of the Mission tab. You will see a "Mission run successfully!" message. At the bottom of the Mission tab, you can track the progress of the job.

Track job progress

Once the computation is completed, the job will shut down and collect the results. Once the job progress reads "Completed", you can download the results by opening the three-dot menu and clicking **Download**.

Download results

Let's take a look at the the output.log file first, which returns the results of the regression and simulation:

```
REGRESSION EXAMPLE
Let's run a simple regression with the mtcars dataset and make a plot, which will be saved in the
filesys folder.
StatsModels.TableRegressionModel{LinearModel{GLM.LmResp{Array{Float64,1}},GLM.DensePredChol{Float64,
LinearAlgebra.Cholesky{Float64,Array{Float64,2}}}},Array{Float64,2}}

mpg ~ 1 + wt

Coefficients:
─────────────────────────────────────────────────────────────────────
              Coef.   Std. Error      t   Pr(>|t|)  Lower 95%  Upper 95%
─────────────────────────────────────────────────────────────────────
(Intercept)  37.2851     1.87763   19.86    <1e-18    33.4505    41.1198
wt           -5.34447    0.559101  -9.56    <1e-9     -6.48631   -4.20263
─────────────────────────────────────────────────────────────────────
MONTE CARLO EXAMPLE
What is the probability that the sum of rolling two fair dice is at least 7?
We can work out the answer (0.583), but let's prove it with simulation.
We'll write a function that simulates trials of dice throws and returns TRUE if at least 7.
Let's try 50,000 trials, record the system time, & calculate the mean of the outcomes.
  0.006619 seconds (50.10 k allocations: 4.055 MiB)
The mean is: 0.58188
Pretty close but we're still a little off -- let's try 10 million throws, record the system time, &
calculate the mean of the outcomes.
  1.386919 seconds (10.02 M allocations: 739.741 MiB, 26.05% gc time)
The mean is: 0.5832895
As you can see, the result is much more accurate, and the run didn't take very long.
```

Output.log results

Contact us

We hope this tutorial was helpful. Please let us know if you have any questions or any problems using Galileo. Your feedback is extremely important to us. Contact us anytime at matthew@hypernetlabs.io or alexander@hypernetlabs.io.