Porting Your Algorithm to Algorun Platform

Center of Quantitative Medicine – UConn Health

195 Farmington Avenue, Farmington - CT

Contents

Version Control

Date	Author	Contact	Changes
June 23, 2015	Ibrahim	abdelrahman.hosny@hotmail.com	- Initiated the document. - Added steps for algorun 0.9.1
June 30, 2015	Ibrahim	abdelrahman.hosny@hotmail.com	- Added introduction about Docker. - Added information about algorithm. - Enhanced the example.
July 8, 2015	Ibrahim	abdelrahman.hosny@hotmail.com	- Reflect changes in algorun_info
July 22, 2015	Ibrahim	abdelrahman.hosny@hotmail.com	- Added offline command - Added keywords for search directory.
August 5, 2015	Ibrahim	abdelrahman.hosny@hotmail.com	- Updated Dockerfile and manifest.json file.

Introduction

Algorun is the base component of a data analytics platform that is built to support running and integrating algorithms written for algebraic modeling. It is built mainly to make ADAM[1] modules work together. In this document, we are illustrating how to make your algorithm run on Algorun in three easy steps. If you have more questions, go to http://algorun.org and ask for help.

Scope

The document covers the steps needed to run your algorithm on Algorun. By the end of this document, you should be able to run your algorithm as a web service with the least configuration possible. However, it does not describe how the platform is built nor the technical details of the internal workings. You may navigate to our website for more technical details.

Intended Audience

Information presented here are mainly intended for algorithms developers. Specially, developers who have an algorithm that fits in the domain of Algorun and wants to make it work on our platform.

What will the platform provide for you?

Using the platform, you are wrapping your algorithm dependencies along with all your algorithm code in a single container. Then, you can run your algorithm as a callable web service.

[1] ADAM: Analysis of Dynamic Algebraic Models. More info: https://github.com/PlantSimLab/ADAM

Before you start

Before you go with the rest of the document, we assume that you have a minimal knowledge with Docker[2] technology. If this is your first time to work with Docker, we are presenting a brief introduction about the technology and how to start with it.

What is Docker?

Docker is a platform that enables developers and system admins to ship their applications in the easiest way. In other way, Docker is a technology we use to wrap our applications in a container that we can move and run anywhere without having to make difficult configurations or dependencies installation. As long as you have Docker installed on a machine, you can run your application in a container; whatever your application does or dependencies it has. The following figure illustrates the main scheme.

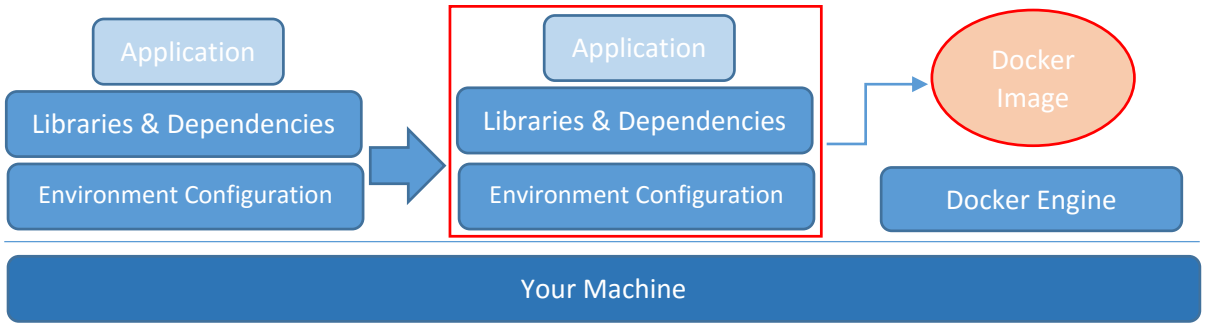

Figure 1 - Docker main scheme

Why do we use Docker?

So, in our work domain, we are using Docker for many reasons:

1- It enables us to wrap all application dependencies, libraries, source code and environment's default configuration in one image that can be run on any machine that has Docker engine installed. This is extremely useful in our problem domain as it comes over the problem of reproducibility of science. In other word, the problem of reproducing somebody else's results in computational sciences is still a barrier in improving research. Using Algorun, you can easily re-produce results very quickly.

2- It gives us the ability to transform an already-existing algorithm to work as a web interface without having to re-write algorithm code itself. This gives the opportunity for modelers to try an algorithm with no difficulty.

3- You do not have to write any additional code to port your algorithm to Algorun. Algorun Docker image will do the task for you.

Installing Docker

To install Docker on your machine, please navigate to http://docs.docker.com/mac/started/ and select your machine (Mac OS X, Linux or Windows) and follow the steps. You can make sure that the Docker is installed correctly by running:

```
docker run hello-world
```

[2] Docker is an open platform for distributed applications, developers and sys admins. More info: www.docker.com

Dockerfile

To build a Docker image, you can describe it in a Dockerfile. Find more information about Dockerfile on: https://docs.docker.com/reference/builder/

Algorun comes with a template Dockerfile that you will just edit; no need to write a one from scratch.

How to Port Your Algorithm to Algorun Platform

In this section, we will show you how to port your algorithm to Algorun. If you have not worked with Docker before, please refer to the previous section.

Step 1: Prepare your algorithm

Prepare your algorithm to work in the following scenario:

- Accept the input as a file and produce the output to another file.
- Have a single entry point. That is, it starts execution from one point. If your algorithm start executing on multiple files, you should combine them such that the **algo_exec** (details below) operates on one file only.
- Read the first command line argument as the file name as following:
  `command input_file`

`command`	Mandatory	The command you use through terminal to run your algorithm. No restriction on the command length. You can use flags as well or pass more arguments before `input_file` (then the algorithm should read the input file as the nth argument; not the first.)
`input_file`	Mandatory	The input_file your algorithm read and process.

Step 2: Wrap your algorithm in a container

To make the process easy, we have provided you with a directory that contains all what you need. You will just edit the files on it. Go to our repository: https://github.com/algorun/algorun and clone it to your machine (You will just use 'try it' directory for now). When you open the directory, you should be able to figure out the folders as in the following figure:

Figure 2 - Try Algorun directory

- First, you put all your algorithm source code in src folder.
- Second, edit the Dockerfile. Open the file and follow the instructions in it. Basically, you are going only to install algorithm dependencies.
  Example: `RUN apt-get install -y wget ruby2.0`
- Finally, edit files in the algorun_info folder.

manifest.json	
algo_name	the name of your algorithm
algo_summary	a 1-line summary of your algorithm
algo_description	a few lines description of your algorithm (basic html accepted)
algo_website	url where users can find more information about your algorithm
algo_keywords	a list of keywords describing your algorithm
algo_authors	a list of all the algorithm's authors
algo_exec	the command to start algorithm execution. AlgoRun passes the input file as the first argument
algo_output_filename	path of the file where your algorithm outputs its results
algo_parameters	list all the parameters available for your algorithm. These parameters will be available for use in your algorithm as environment variables.
algo_image	Example: "ahosny/react". leave "" if you did not upload your algorithm to docker hub or if you do not know what docker hub is :)
input_example.txt	Fill it with the format of the input your algorithm accepts. This serves as a guidance users to know how to input data
output_example.txt	Fill it with the format of the output your algorithm produces. This gives the user a glance on what result he will get

What are algorithm parameters (algo_parameters)?

[OPTIONAL] In some cases, your algorithm might use some configuration. This configuration should be presented as parameters in a key-value pair **(parameter name, parameter value)**. To provide default configuration, we make use of environment variables to save the default values of these parameters. Your algorithm then reads default configuration from environment variables. In the provided web interface, the user can easily change these parameters and run the algorithm again with the new configuration.

Step 3: Build, run and test your container

- In you working directory, use the following command to build your container:
  `docker build -t <username>/<algorithm> .`
- Make sure that the build is successful.
- Use the following command to run your container:
  `docker run -d -p 31331:8765 <username>/<algorithm>`
- Open your web browser and go to http://localhost:31331 and see your algorithm in action.

> ***Tip:*** *you can also send an HTTP POST request to http://localhost:31331/do/run with parameter 'input' containing the test input to the algorithm. You should expect the output immediately as the output of the request.*