Quickstart Guide

Joe Mills

10/5/2017

Contents

Overview

Welcome to the NCAR/CUAHSI WRF-Hydro training workshop! This guide will give you an overview of the training materials and software.

Software

There are 4 primary pieces of software required for this training.

- Windows 10 w/ Windows Powershell
- ArcGIS 10.3.1
- Docker for Windows 17.09
- Xming 6.9

In reality, there are many other required software package and libraries. However, Docker is used to wrap all these other dependencies into environments called "containers". These environments use Ubuntu 16 Linux as the underlying OS and build upon this base with all the required libraries for each part of this training.

Docker

Docker is a light-weight container solution that is built on top of the Linux kernel. Briefly, Docker allows one to run isolated environments that share the same Linux kernel, thus requiring less overhead to start and run than virtual machines. However, on Windows, Docker must run on-top of a virtual machine to provide the base Linux kernel.

Docker containers are used in this training to package all of the required dependencies for WRF-Hydro and associated tools. One additional benefit is that all of these dependencies are documented in the form of Docker files. Docker files are scripts used to build the docker containers, and thus contain all of the information needed to replicate the environment elsewhere. These docker files can be used as a road map for replicating the training environment on another Linux machine, even if docker is not installed.

The Docker files used for this training are in the **docker/dockerFiles** directory.

Docker images

There are three Docker images used in this training, which can also be built from scratch using the Docker files provided.

1. wrf_hydro_docker - Container for the WRF-Hydro model, associated dependencies, and command line tools such as NC-View and NCL.
2. wps_docker - Container for the WRF Preprocessing System (WPS) and all dependencies.
3. rwrfhydro_docker - Container for the R interpreter, the rwrfhydro R package and dependencies, and R-Studio server.

These images are provided in the **docker/dockerImages** directory as tar balls.

File sharing

The training is setup such that one folder, **ClassMaterials**, is shared among all the containers as well as with the native Windows file system. Thus, any data created in a container is stored on the Windows file system, accessible through windows explorer, and persistent after Docker is closed.

Important notes on files

1. All data that is created within a Docker container is **lost** when the container is closed unless it is stored in the persistent **ClassMaterials** folder.
2. Some data is stored inside the container at start up, including the source code for WRF-Hydro and WPS. WRF-Hydro and WPS may not compile correctly if the data is stored outside the container on the native Windows file system.

Training environment

For convenience, a number of ***.bat** files have been included to automate the start-up procedure. In addition to setting up Docker, these files also setup links between Docker and the **ClassMaterials** folder on the local file system, establish X11 forwarding to Xming, and open ports for R-Studio.

There are 6 ***.bat** files contained in the **/bat** directory.

First time startup

IMPORTANT Using these startup files will start a clean training environment and any data created in the training that is *NOT* in teh **ClassMaterials** folder will be **LOST**

1. **training_config.bat** - This file initializes the Docker environment by first **removing all Docker images currently loaded**. This is done for purposes of this training to remove any potential Artifacts from previous sessions. After the containers are removed, the images are reloaded and the Powershell window will close.
2. **wps_training.bat** - This file opens Windows Powershell and starts the WPS Docker container.
3. **wrf_hydro_training.bat** - This file opens Windows Powershell and starts 3 containers: WRF-Hydro on Ubuntu, R + rwrfhydro on Ubuntu, and R-Studio server.

Restarting a closed container

Use the ***.bat** files with the **_RESTART** filename to restart a closed container. This will preserve data created in the container, such as compiled executables.

Pre-setup.

This is only required if replicating this training on your own. File sharing must be enabled to link the C drive with the docker containers in the Docker Virtual Machine.

1. Right-click the Docker icon in the background applications tab in the lower right hand corner of Windows 10.
2. Click **settings** -> **Shared Drives** and check **C:**. Click **apply** at the bottom right.
3. Docker will ask for your login password for permission. Use the login password provided to you at the training.
4. Wait for Docker to finish applying your changes.
5. Click **Advanvced** on the left-hand navigation bar to access performance settings.
6. Change the number of CPUs to 7 and the memory to 12GB.

Launching the training environment

1. Navigate to **ClassMaterials/bat**
2. Double-click the **training_config.bat** file to execute and initialize the training environment
3. Double-click **wps_training.bat** or **wrf_hydro_training.bat** files to start-up the WPS or WRF-Hydro + R containers, respectively

Connecting to R-Studio server

R-Studio server is accessed via web browser. Open your web browser and put the following into the address bar: **localhost:8787**. You should be directed to a prompt for login and password for R-Studio. The login is **rstudio** and password is **rstudio**