

**“Improving the
Architecture & Development
Process for AIRPACT:
Testing Document”**

Mentor: Joe Vaughan

Instructor: Aaron Crandall

Course: Computer Science 423

Team: Kyler Little, Garrett Rudisill, Slater
Weinstock, Jeff Kremer

Date: 10/19/2019

Testing Plan

Testing Granularity

> Unit Tests

Unit	Description	Expected Output / Result	Passed / Approved By
BlueSky Framework Binary (bluesky)	Running the BlueSky Binary	Identical output files (fire_locations.csv, fire_events.csv) + similar summary.txt files as those produced by existing Ubuntu Server	Passed
fire_ptday_SFonly_Zhang.py	Running python script converts BlueSky fire info (fire_locations.csv) into ORL ptday SMOKE files for CMAQ inline fire plume rise	Produces ORL ptday file.	Approved by Kyler Little
fire_ptinv_SFonly_Zhang.py	Running python script converts BlueSky fire info (fire_locations.csv) into ORL ptday SMOKE files for CMAQ inline fire plume rise	Produces ORL ptinv file.	Approved by Kyler Little
write_kml.py	Running python script converts BlueSky fire info into KML for AIRPACT5 webpage	Produces fire_locations_YYYYMMDD.kml file to be displayed on AIRPACT5 webpage	Approved by Kyler Little

> Integration Tests

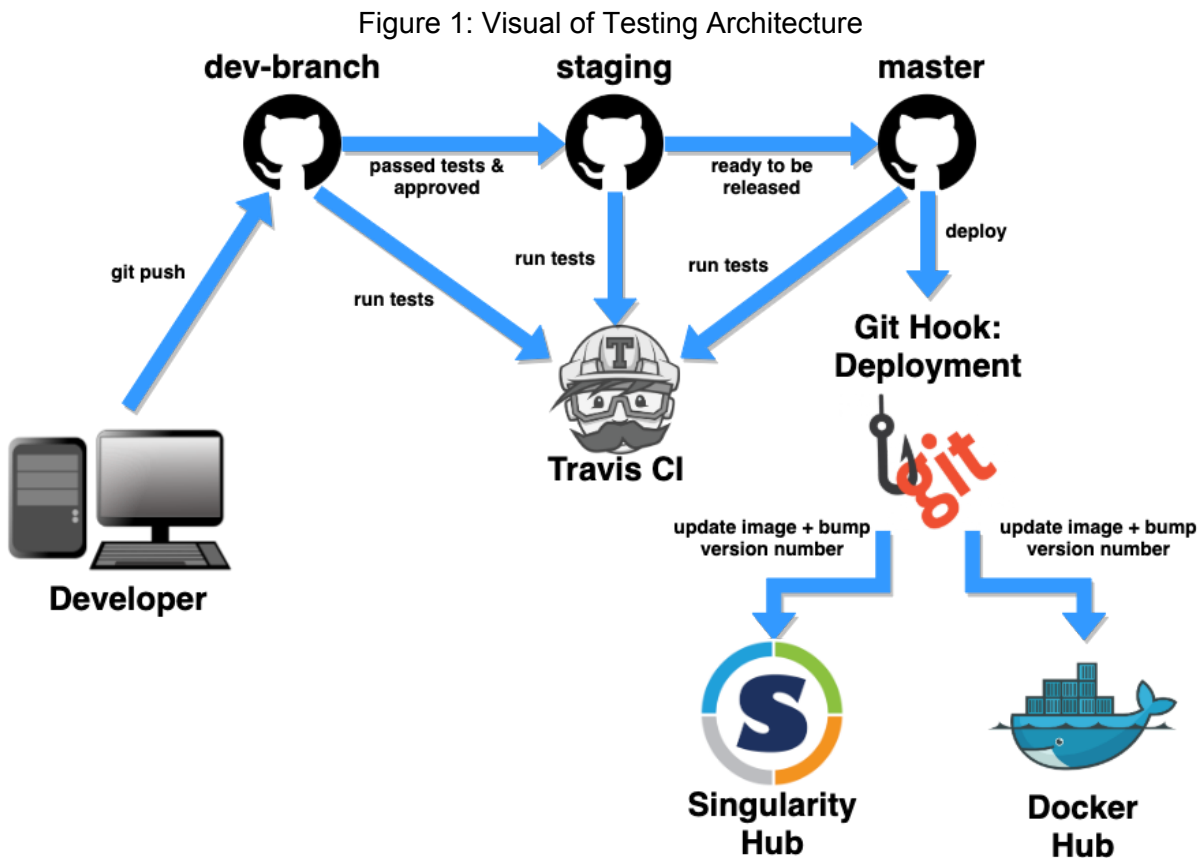
BSF_EFO_AP5_SFonly.csh	Run BlueSky SMARTFIRE for EmissionFilesOnly for AIRPACT-5 Date	Runs the above 4 scripts successfully, transfers output files to "TARGET_DIR = ~/AIRHOME/run_ap5_day1/emis/fire_orl/transfer", deletes the BlueSky Framework output files	Approved by Kyler Little
------------------------	--	---	--------------------------

Testing Architecture

The testing architecture for this project can be viewed in Figure 1 below. It utilizes a Continuous Integration (CI) system known as Travis CI. It also utilizes a Continuous Deployment (CD) system, which is built on top the CI system. Once the CI system deems that a branch is ready to be merged and all developers approve it, Git Hooks will deploy these successfully tested artifacts to DockerHub and SingularityHub.

The testing process itself runs our containerized Bluesky Framework System and compares the output data from the container to the output data from the Bluesky Framework System that is currently running as an Ubuntu Server on the Aeolus High Performance Computing Cluster. These tests are run automatically any time that commits are pushed to a branch or pull requests are made against another branch. This ensures that all changes do not affect the output of the system when moved into the production environment.

A visual of the entire process is displayed below:



Testing Implementation Details

The automated testing, performed on the “BlueSky Framework Binary” unit, is done programmatically through a Python 3 based script. This script has a dependency on numpy for parsing and populating data. Testing is primarily done through the following process.

1. Parse the saved output data (which we will verify against) into a set.
2. Parse the new output data from the image under test into a set.
3. Assert that these tests are equal, therefore saying the data is equivalent

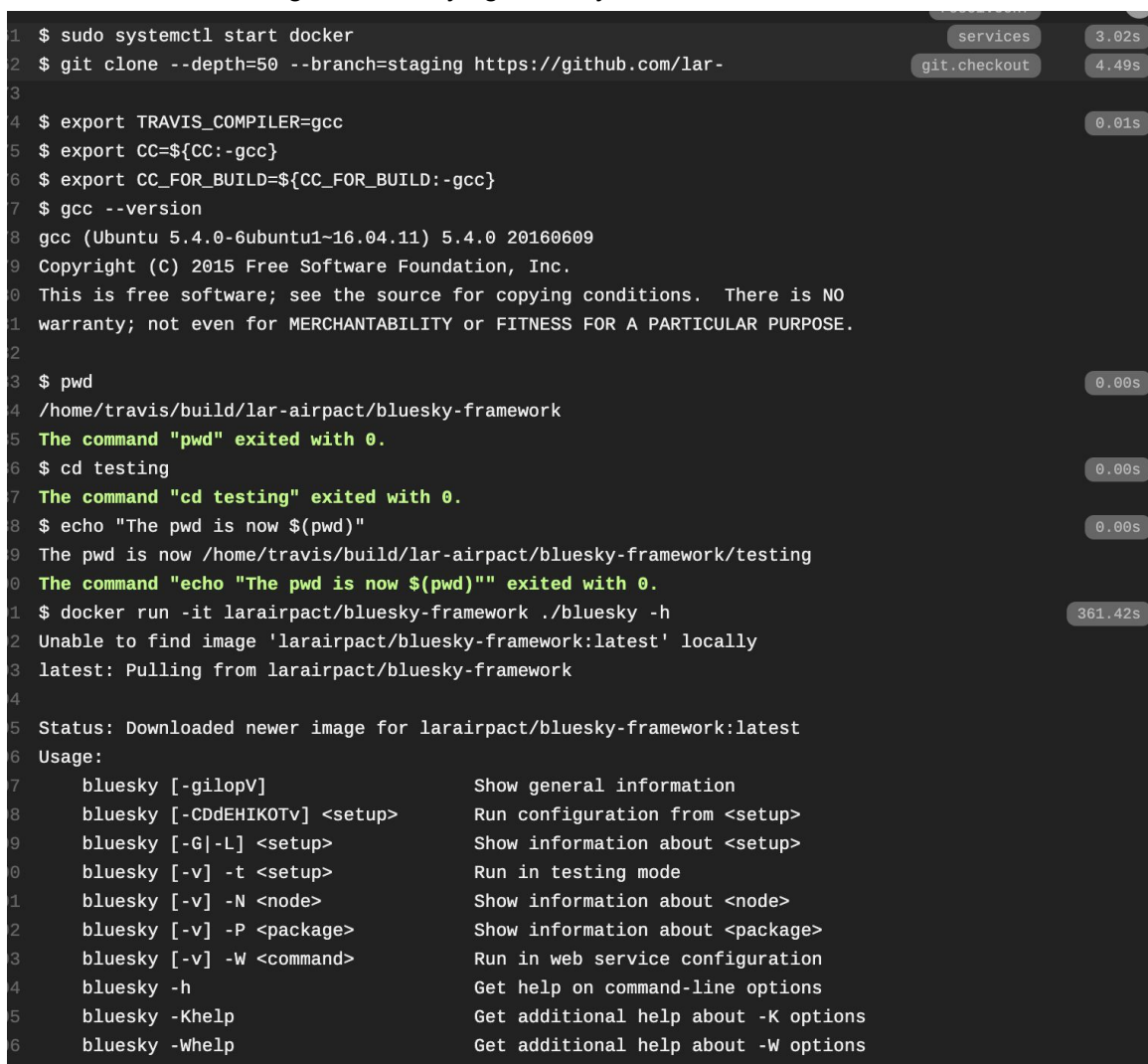
Because the testing checks for perfect equivalency of files, we are able to ensure that our Docker & Singularity Images represent perfect replicas of the existing Bluesky Framework System running on the Ubuntu Server in the Aeolus HPCC.

Testing Results

Through automated testing, we were able to verify that the output of the containerized Bluesky Framework correctly matched the data output of the in-place Bluesky Framework system being hosted on Aeolus. This test now serves as the primary metric against which all future development on the Bluesky Framework system is done. This specific test is crucial because everything else is dependent on this key job. Specifically, once it's been determined that this output is the same, every subsequent unit's work (e.g. the three python scripts) is guaranteed to match what we expect.

In the following figures are various displays of the process of testing and the output during/finishing the test run.

Figure 2: Verifying Bluesky Framework build runs



```
1 $ sudo systemctl start docker
2 $ git clone --depth=50 --branch=staging https://github.com/lar-
3
4 $ export TRAVIS_COMPILER=gcc
5 $ export CC=${CC:-gcc}
6 $ export CC_FOR_BUILD=${CC_FOR_BUILD:-gcc}
7 $ gcc --version
8 gcc (Ubuntu 5.4.0-6ubuntu1~16.04.11) 5.4.0 20160609
9 Copyright (C) 2015 Free Software Foundation, Inc.
10 This is free software; see the source for copying conditions. There is NO
11 warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
12
13 $ pwd
14 /home/travis/build/lar-airpact/bluesky-framework
15 The command "pwd" exited with 0.
16 $ cd testing
17 The command "cd testing" exited with 0.
18 $ echo "The pwd is now $(pwd)"
19 The pwd is now /home/travis/build/lar-airpact/bluesky-framework/testing
20 The command "echo "The pwd is now $(pwd)"" exited with 0.
21 $ docker run -it larairpact/bluesky-framework ./bluesky -h
22 Unable to find image 'larairpact/bluesky-framework:latest' locally
23 latest: Pulling from larairpact/bluesky-framework
24
25 Status: Downloaded newer image for larairpact/bluesky-framework:latest
26 Usage:
27     bluesky [-gilopV]                Show general information
28     bluesky [-CDdEHKOTv] <setup>    Run configuration from <setup>
29     bluesky [-G|-L] <setup>         Show information about <setup>
30     bluesky [-v] -t <setup>         Run in testing mode
31     bluesky [-v] -N <node>          Show information about <node>
32     bluesky [-v] -P <package>       Show information about <package>
33     bluesky [-v] -W <command>       Run in web service configuration
34     bluesky -h                      Get help on command-line options
35     bluesky -Khhelp                 Get additional help about -K options
36     bluesky -Whelp                  Get additional help about -W options
```

Figure 3: Verifying the data output of Bluesky Framework Docker Image

```

0 WildfireGrowth: Persistence from package BlueSky v3.5.1
1 WildfireGrowth: Persistence model created 599 new fire records
2 PrescribedFireGrowth: Persistence from package BlueSky v3.5.1
3 PrescribedFireGrowth: Persistence model created 3600 new fire records
4 OtherFireGrowth: NoGrowth from package BlueSky v3.5.1
5 ConsolidateGrowth: ConsolidateFires from package BlueSky v3.5.1
6 FCCS: FCCS Fuel Loading Module v2
7 FCCS: Successfully added fuel loadings for 5037 of 6018 input fires
8 ConsumptionSplit: FireTypeSplitter from package BlueSky v3.5.1
9 WildfireConsumption: NoConsumption from package BlueSky v3.5.1
0 PrescribedFireConsumption: NoConsumption from package BlueSky v3.5.1
1 OtherFireConsumption: NoConsumption from package BlueSky v3.5.1
2 ConsolidateConsumption: ConsolidateFires from package BlueSky v3.5.1
3 NoTimeProfile: NoTimeProfile from package BlueSky v3.5.1
4 NoEmissions: NoEmissions from package BlueSky v3.5.1
5 NoPlumeRise: NoPlumeRise from package BlueSky v3.5.1
6 StandardFiles: Writing fire locations to standard format file
7 StandardFiles: Successfully wrote 6018 fire locations
8 BlueSky: Completed in 2 minutes 40.00 seconds
9 The command "docker run -it --mount type=bind,source="$(pwd)",target=/bluesky/dist/bluesky/output/
  larairpact/bluesky-framework ./bluesky -d 2019090100Z -K no-archive defaultLAR_SFonly" exited with 0.
$ mv 2019090100.1 newinput 0.01s
1 The command "mv 2019090100.1 newinput" exited with 0.
2 $ echo "testing data output" 0.00s
3 testing data output
4 The command "echo "testing data output"" exited with 0.
5 $ python3 test.py 1.14s
6 ...
7 -----
8 Ran 4 tests in 0.249s
9
10 OK
1 The command "python3 test.py" exited with 0.
2
3
4 Done. Your build exited with 0.

```

Finally, since the end product will be running on the Aeolus HPCC, which only uses Singularity as its containerization technology, we needed to test that running the code from within a Singularity Image also worked. The main unit which runs all of the code for this: “BSF_EFO_AP5_SFonly.csh” script. Its output and behavior was manually verified to meet expectations. Within the script itself, the LAR Group calls “ls -al \$TARGET_DIR” as a final command to verify that the data was correctly placed. The “ls” command lists the files in the specified directory, while the “la” options ensure all files are listed in a verbose manner. Lastly, TARGET_DIR is where the LAR Group desired that files were moved to.

In the following figures, we can see the output from running the “BSF_EFO_AP5_SFonly.csh” in our Singularity Image on both a local development machine as well as the Aeolus HPCC. The expected output is a log file, fire_locations CSV file, fire_locations KML file (for the webpage), a ptday ORL file (for further computation done by AIRPACT), a ptinv ORL file (for further computation done by AIRPACT), and lastly, a simple summary text file. As seen below, the actual outputs meet our expectations.

Figure 4: Output from Bluesky Framework Singularity Image on Local Dev Machine

```
kylerlittle@Kylers-MacBook-Air:~/Desktop/Projects/bluesky-framework$ singularity exec --bind /Users/kylerlittle/Desktop/Projects/bluesky-framework/bsf_output_temp/output:/bluesky/dist/bluesky/output --bind /Users/kylerlittle/Desktop/Projects/bluesky-framework/bsf_output_temp/working:/bluesky/dist/bluesky/working --bind /Users/kylerlittle/Desktop/Projects/bluesky-framework/bsf_output_temp/log:/bluesky/dist/bluesky/log --bind /Users/kylerlittle/Desktop/Projects/bluesky-framework/bsf_output_temp/conversion/output:/bluesky/dist/bluesky/conversion/output bluesky-framework_latest.sif /bluesky/dist/bluesky/BSF_EFO_AP5_SFonly.csh 20190901
total 3540
drwxr-xr-x 1 kylerlittle dialout 256 Oct 11 15:01 .
drwxr-xr-x 1 kylerlittle dialout 128 Oct 7 03:56 ..
-rw-r--r-- 1 kylerlittle dialout 29 Oct 11 15:01 bluesky_job.log
-rw-r--r-- 1 kylerlittle dialout 2436215 Oct 11 15:01 fire_locations.csv
-rw-r--r-- 1 kylerlittle dialout 227896 Oct 11 15:01 fire_locations_20190901.kml
-rw-r--r-- 1 kylerlittle dialout 828627 Oct 11 15:01 ptday-2019090100.orl
-rw-r--r-- 1 kylerlittle dialout 127805 Oct 11 15:01 ptinv-2019090100.orl
-rw-r--r-- 1 kylerlittle dialout 847 Oct 11 15:01 summary.txt
END OF SCRIPT /bluesky/dist/bluesky/BSF_EFO_AP5_SFonly.csh
[ 166.431518] reboot: Power down
```

Figure 5: Output from Bluesky Framework Singularity Image on Aeolus HPCC

```
[klittle@aeolus demo]$ singularity exec --bind /home/klittle/demo/bsf_output_temp/output:/bluesky/dist/bluesky/output --bind /home/klittle/demo/bsf_output_temp/working:/bluesky/dist/bluesky/working --bind /home/klittle/demo/bsf_output_temp/log:/bluesky/dist/bluesky/log --bind /home/klittle/demo/bsf_output_temp/conversion/output:/bluesky/dist/bluesky/conversion/output bluesky-framework_latest.sif /bluesky/dist/bluesky/BSF_EFO_AP5_SFonly.csh 20190901
total 3580
drwxr-xr-x 1 klittle lar 222 Oct 11 09:18 .
drwxr-xr-x 1 klittle lar 16 Oct 7 17:15 ..
-rw-r--r-- 1 klittle lar 25188 Oct 11 09:18 bluesky_job.log
-rw-r--r-- 1 klittle lar 2436215 Oct 11 09:18 fire_locations.csv
-rw-r--r-- 1 klittle lar 227896 Oct 11 09:18 fire_locations_20190901.kml
-rw-r--r-- 1 klittle lar 828627 Oct 11 09:18 ptday-2019090100.orl
-rw-r--r-- 1 klittle lar 127805 Oct 11 09:18 ptinv-2019090100.orl
-rw-r--r-- 1 klittle lar 847 Oct 11 09:18 summary.txt
END OF SCRIPT /bluesky/dist/bluesky/BSF_EFO_AP5_SFonly.csh
[klittle@aeolus demo]$
```