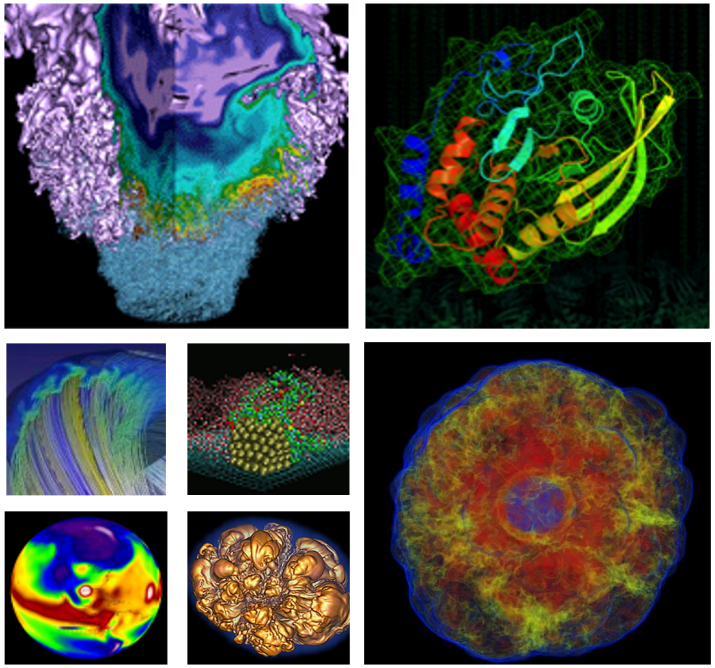
Buildtest: Testing Framework for HPC Systems



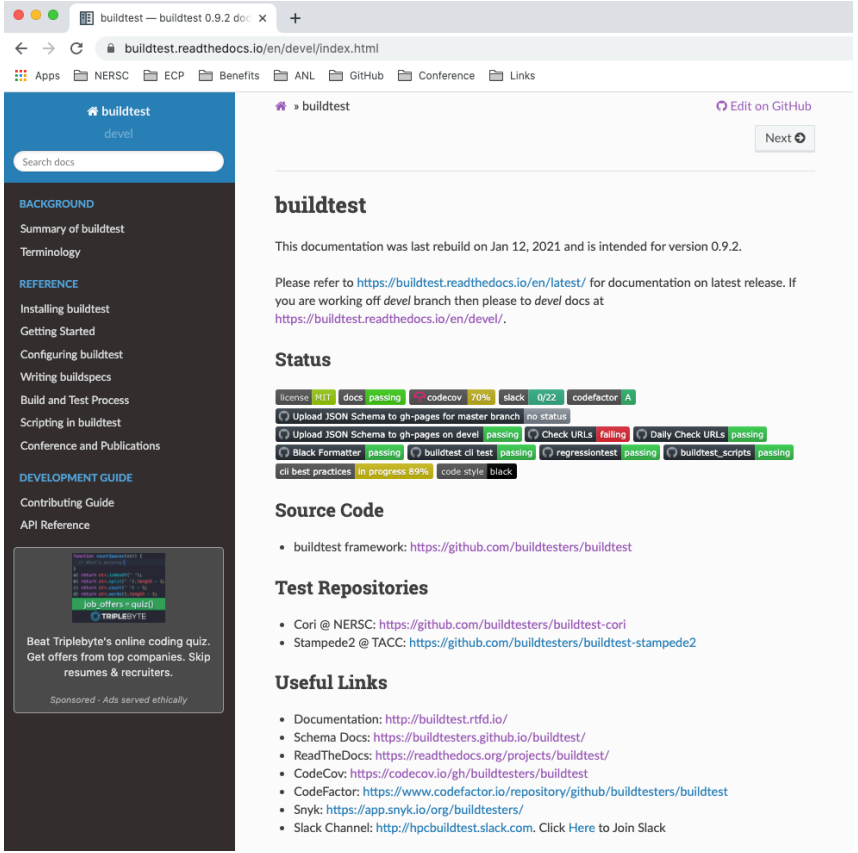
Shahzeb Siddiqui
HPC Consultant/Software Integration Specialist
Lawrence Berkeley National Laboratory

6th Easybuild User Meeting
Jan 29th, 2021

https://buildtest.readthedocs.io/
https://github.com/buildtesters/buildtest
http://hpcbuildtest.slack.com/

What is buildtest

- Buildtest is a HPC Testing Framework for acceptance and regression testing for HPC system.
- Tests are written in YAML that is validated with JSON schema.
- Buildtest automates test creation and execution of test
- Target Audience: HPC Staff
- Buildtest is not
  - Replacement for build tools like make, cmake, autoconf
  - software build framework (easybuild, spack, nix, guix)



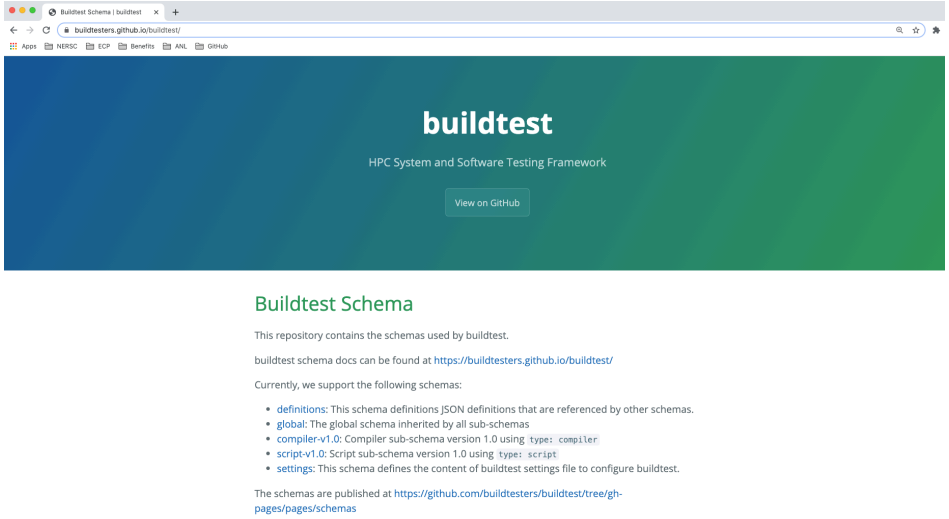
YAML and JSON Schema

- We picked YAML as choice for writing test configuration given its ease of use and widely used in many open source projects such as Travis, Ansible, Gitlab CI, GitHub workflows, Kubernetes, Docker Compose.
- JSON Schema is a vocabulary that annotates and validates JSON documents.
- JSON and YAML are represented as a **dict** in python which makes it easy to convert between each data format. We write JSON Schemas to help validate tests that are written in YAML.
- Buildtest adopts the versioned based schema used in Docker compose to help continue schema development while retaining backward compatibility with previous versions.

```yaml
version: "3.8"
services:
  web:
    build: .
    ports:
      - "5000:5000"
    volumes:
      - .:/code
      - logvolume01:/var/log
    links:
      - redis
  redis:
    image: redis
volumes:
  logvolume01: {}
```

Schemas

- The schema development is implemented independent to buildtest. The schemas and docs are hosted at https://buildtesters.github.io/buildtest/
- We run regression test against example YAML files for each schema to ensure schemas are written in accordance to desired YAML construct.
- We automate JSON Schema documentation using adobe/jsonschema2md into Markdown pages and publish schema and documentation to GitHub pages
- Schemas are versioned to allow development to schemas and its YAML structure.



compiler schema version 1.0 Schema

`compiler-v1.0.schema.json`

The compiler schema is of `type: compiler` in sub-schema which is used for compiling and running programs

Abstract	Extensible	Status	Identifiable	Custom Properties	Additional Properties	Acc Restri
Can be instantiated	Yes	Unknown status	No	Forbidden	Forbidden	none

compiler schema version 1.0 Type

`object` (compiler schema version 1.0)

compiler schema version 1.0 Properties

Property	Type	Required	Nullable	Defined by
type	`string`	Required	cannot be null	compiler schema version 1.0
description	`string`	Optional	cannot be null	compiler schema version 1.0
compilers	`object`	Required	cannot be null	compiler schema version 1.0
source	`string`	Required	cannot be null	compiler schema version 1.0
executor	`string`	Required	cannot be null	compiler schema version 1.0
run_only	`object`	Optional	cannot be null	compiler schema version 1.0
skip	`boolean`	Optional	cannot be null	compiler schema version 1.0
tags	Merged	Optional	cannot be null	compiler schema version 1.0

type

Select schema type to use when validating buildspec. This must be of set to `compiler`.

`type`

- is required
- Type: `string`
- cannot be null
- defined in: compiler schema version 1.0

type Type

`string`

type Constraints

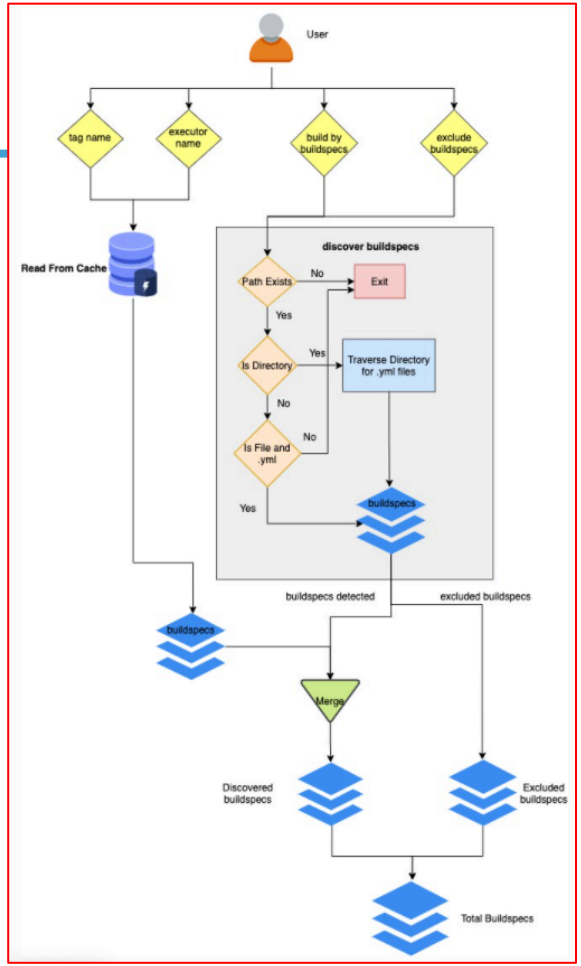
pattern: the string must match the following regular expression:

```
^compiler$
```

try pattern

Command Line Usage

Command	Description
buildtest build –b <FILE>	Build from a single file
buildtest build –b <DIR>	Build all buildspecs in directory
buildtest build –b <FILE> -b <DIR>	Build from a file and directory
buildtest build –-tags <TAGNAME>	Build all buildspecs with tag <TAGNAME> from buildspec cache
buildtest build –b <FILE> -b <DIR> --tags <TAGNAME>	Build buildspec by file, directory and tag
buildtest build –b <dir> -x <file> –x <dir>	Build buildspec by directory and exclude a file and directory
buildtest build –executor <EXECUTORNAME>	Build all tests with executor <EXECUTORNAME> from buildspec cache



Build by Buildspec

```
$ buildtest build -b $HOME/Documents/buildtest/tutorials/python-hello.yml

+-------------------------------+
| Stage: Discovering Buildspecs |
+-------------------------------+


Discovered Buildspecs:

/Users/siddiq90/Documents/buildtest/tutorials/python-hello.yml

+---------------------------+
| Stage: Parsing Buildspecs |
+---------------------------+

 schemafile              | validstate   | buildspec
-------------------------+--------------+-------------------------------------------------------------------------
script-v1.0.schema.json | True         | /Users/siddiq90/Documents/buildtest/tutorials/python-hello.yml

+----------------------+
| Stage: Building Test |
+----------------------+

 name         | id       | type   | executor   | tags   | testpath
--------------+----------+--------+------------+--------+-----------------------------------------------------------------------------------------------------------------------------------
python_hello | bd2c48dc | script | local.bash | python | /Users/siddiq90/Documents/buildtest/var/tests/local.bash/python-hello/python_hello/17/stage/generate.sh

+----------------------+
| Stage: Running Test  |
+----------------------+

 name         | id       | executor   | status   |   returncode | testpath
--------------+----------+------------+----------+--------------+-----------------------------------------------------------------------------------------------------------------------------------
 python_hello | bd2c48dc | local.bash | PASS     |            0 | /Users/siddiq90/Documents/buildtest/var/tests/local.bash/python-hello/python_hello/17/stage/generate.sh

+----------------------+
| Stage: Test Summary  |
+----------------------+
Executed 1 tests
Passed Tests: 1/1 Percentage: 100.000%
Failed Tests: 0/1 Percentage: 0.000%
```

List of Discovered buildspecs

JSON Schema for validating buildspec

Name of Test

Unique Test ID

Generated Test

State of test, can be PASS or FAIL

Building By Tags

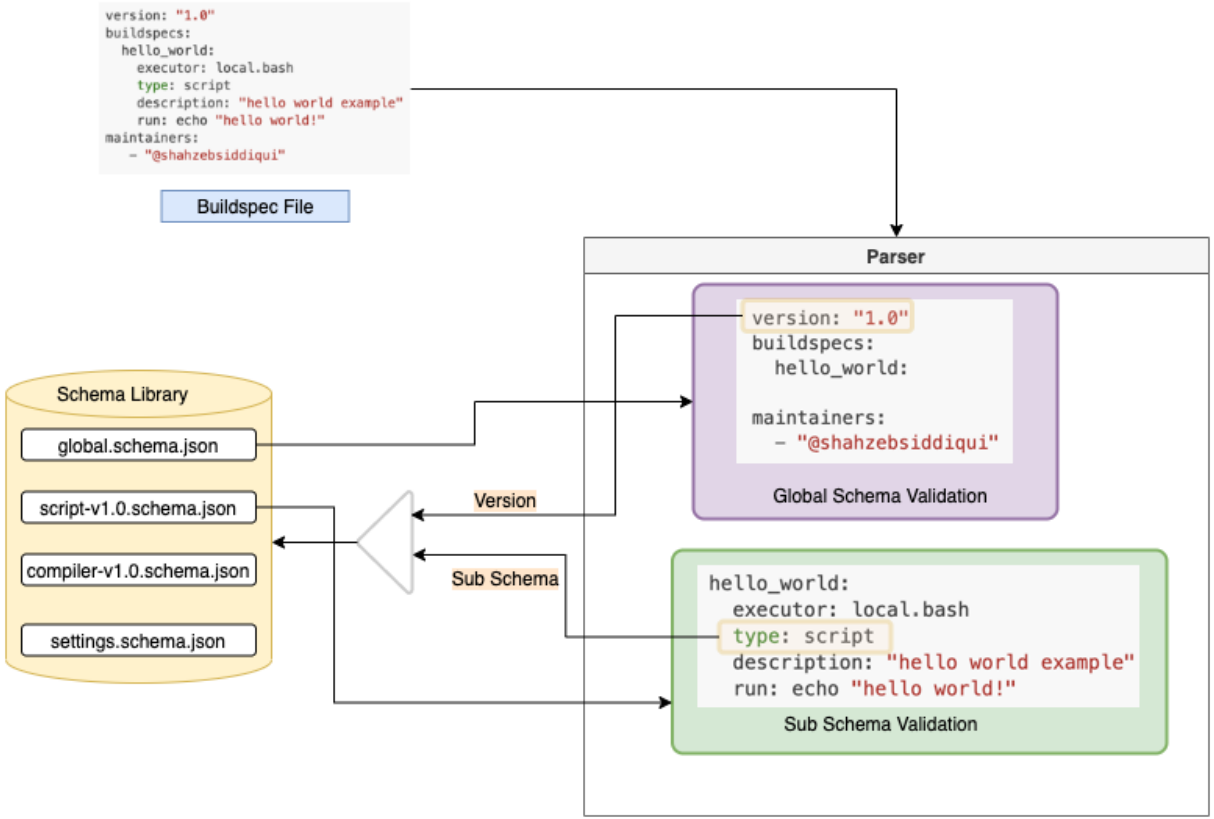
```
$ buildtest build --tags pass

+-------------------------------+
| Stage: Discovering Buildspecs |
+-------------------------------+


Discovered Buildspecs:

/Users/siddiq90/Documents/buildtest/tutorials/pass_returncode.yml

+----------------------------+
| Stage: Parsing Buildspecs |
+----------------------------+

 schemafile              | validstate   | buildspec
-------------------------+--------------+-------------------------------------------------------------------------
 script-v1.0.schema.json | True         | /Users/siddiq90/Documents/buildtest/tutorials/pass_returncode.yml

+----------------------+
| Stage: Building Test |
+----------------------+

 name                     | id       | type   | executor   | tags                  | testpath
--------------------------+----------+--------+------------+-----------------------+-----------------------------------------------------------------------------------------------------------------------------------------------------------
 exit1_fail               | 29710ea6 | script | local.sh   | ['tutorials', 'fail'] | /Users/siddiq90/Documents/buildtest/var/tests/local.sh/pass_returncode/exit1_fail/30/stage/generate.sh
 exit1_pass               | 924f9d61 | script | local.sh   | ['tutorials', 'pass'] | /Users/siddiq90/Documents/buildtest/var/tests/local.sh/pass_returncode/exit1_pass/32/stage/generate.sh
 returncode_list_mismatch | 66b4f136 | script | local.sh   | ['tutorials', 'fail'] | /Users/siddiq90/Documents/buildtest/var/tests/local.sh/pass_returncode/returncode_list_mismatch/30/stage/generate.sh
 returncode_int_match     | 136e5563 | script | local.sh   | ['tutorials', 'pass'] | /Users/siddiq90/Documents/buildtest/var/tests/local.sh/pass_returncode/returncode_int_match/32/stage/generate.sh



+----------------------+
| Stage: Running Test  |
+----------------------+

 name                     | id       | executor   | status   |   returncode | testpath
--------------------------+----------+------------+----------+--------------+-----------------------------------------------------------------------------------------------------------------------------------------------------------
 exit1_fail               | 29710ea6 | local.sh   | FAIL     |            1 | /Users/siddiq90/Documents/buildtest/var/tests/local.sh/pass_returncode/exit1_fail/30/stage/generate.sh
 exit1_pass               | 924f9d61 | local.sh   | PASS     |            1 | /Users/siddiq90/Documents/buildtest/var/tests/local.sh/pass_returncode/exit1_pass/32/stage/generate.sh
 returncode_list_mismatch | 66b4f136 | local.sh   | FAIL     |            2 | /Users/siddiq90/Documents/buildtest/var/tests/local.sh/pass_returncode/returncode_list_mismatch/30/stage/generate.sh
 returncode_int_match     | 136e5563 | local.sh   | PASS     |          128 | /Users/siddiq90/Documents/buildtest/var/tests/local.sh/pass_returncode/returncode_int_match/32/stage/generate.sh

+----------------------+
| Stage: Test Summary  |
+----------------------+

Executed 4 tests
Passed Tests: 2/4 Percentage: 50.000%
Failed Tests: 2/4 Percentage: 50.000%
```

General Pipeline

- For every discovered buildspecs, buildtest will do the following:
  - Parse: Validates buildspec with JSON Schema
  - Build: Generates testscript from YAML
  - Run: Executes tests via local or batch executor
  - Gather Results: Write output/error file and get return code
  - Update Report: Update report file with test results including any metadata

Buildspec Validation Process

- Every buildspec is validated by global schema and a subschema defined b **type** field.
- Buildtest will skip any buildspecs that fails validation.



Buildspec Structure

```
version: "1.0"                                                    Schema Version
buildspecs:                                                       Declaration of tests
  systemd_default_target:                                         Name of Test
    executor: local.bash                                          Name of Executor
    type: script                                                  Schema Type
    description: check if default target is multi-user.target     Description of Test
    tags: [tutorials]                                             Tag Name
    run: |
      if [ "multi-user.target" == `systemctl get-default` ]; then
        echo "multi-user is the default target";
        exit 0                                                    Script
      fi
      echo "multi-user is not the default target";
      exit 1
```

Control build stages

- For development of buildspec, you may want to build test without running it. This can be done via **–stage** option.
- With –stage=parse we can stop after the Parse stage that's useful if you want to validate buildspec.
- We can also stop after build phase using –stage=build which can be used if you want to inspect generated script

```
$ buildtest build -b tutorials/systemd.yml --stage=parse
Paths:
___________
Test Directory: /Users/siddiq90/Documents/buildtest/var/tests

+---------------------------------+
| Stage: Discovered Buildspecs  |
+---------------------------------+

/Users/siddiq90/Documents/buildtest/tutorials/systemd.yml

+-----------------------------+
| Stage: Parsing Buildspecs |
+-----------------------------+

 schemafile              | validstate   | buildspec
-------------------------+--------------+-----------------------------------------------------------------
 script-v1.0.schema.json | True         | /Users/siddiq90/Documents/buildtest/tutorials/systemd.yml
```

Return Code Matching

- The returncode field can be used to customize how test is passed, by default a returncode 0 is a **PASS**
- The returncode can be a a single number or a list of returncodes to match

```
version: "1.0"
buildspecs:

  exit1_fail:
    executor: local.sh
    type: script
    description: exit 1 by default is FAIL
    tags: [tutorials, fail]
    run: exit 1

  exit1_pass:
    executor: local.sh
    type: script
    description: report exit 1 as PASS
    run: exit 1
    tags: [tutorials, pass]
    status:
      returncode: [1]

  returncode_list_mismatch:
    executor: local.sh
    type: script
    description: exit 2 failed since it failed to match returncode 1
    run: exit 2
    tags: [tutorials, fail]
    status:
      returncode: [1, 3]

  returncode_int_match:
    executor: local.sh
    type: script
    description: exit 128 matches returncode 128
    run: exit 128
    tags: [tutorials, pass]
    status:
      returncode: 128
```

```
$ buildtest build -b tutorials/pass_returncode.yml

+--------------------------------+
| Stage: Discovering Buildspecs |
+--------------------------------+


Discovered Buildspecs:

/Users/siddiq90/Documents/buildtest/tutorials/pass_returncode.yml

+----------------------------+
| Stage: Parsing Buildspecs |
+----------------------------+

 schemafile              | validstate   | buildspec
-------------------------+--------------+-----------------------------------------------------------------------
 script-v1.0.schema.json | True         | /Users/siddiq90/Documents/buildtest/tutorials/pass_returncode.yml

+-----------------------+
| Stage: Building Test |
+-----------------------+

 name                      | id       | type   | executor   | tags                    | testpath
---------------------------+----------+--------+------------+-------------------------+-----------------------------------------------------------------------------------------------------------------------------------------------------------
 exit1_fail                | 1b7337d4 | script | local.sh   | ['tutorials', 'fail']   | /Users/siddiq90/Documents/buildtest/var/tests/local.sh/pass_returncode/exit1_fail/5/stage/generate.sh
 exit1_pass                | c4d22774 | script | local.sh   | ['tutorials', 'pass']   | /Users/siddiq90/Documents/buildtest/var/tests/local.sh/pass_returncode/exit1_pass/3/stage/generate.sh
 returncode_list_mismatch  | dfc4ad0f | script | local.sh   | ['tutorials', 'fail']   | /Users/siddiq90/Documents/buildtest/var/tests/local.sh/pass_returncode/returncode_list_mismatch/5/stage/generate.sh
 returncode_int_match      | 0a6555c3 | script | local.sh   | ['tutorials', 'pass']   | /Users/siddiq90/Documents/buildtest/var/tests/local.sh/pass_returncode/returncode_int_match/3/stage/generate.sh

+-----------------------+
| Stage: Running Test  |
+-----------------------+

 name                      | id       | executor   | status   |   returncode | testpath
---------------------------+----------+------------+----------+--------------+-----------------------------------------------------------------------------------------------------------------------------------------------------------
 exit1_fail                | 1b7337d4 | local.sh   | FAIL     |            1 | /Users/siddiq90/Documents/buildtest/var/tests/local.sh/pass_returncode/exit1_fail/5/stage/generate.sh
 exit1_pass                | c4d22774 | local.sh   | PASS     |            1 | /Users/siddiq90/Documents/buildtest/var/tests/local.sh/pass_returncode/exit1_pass/3/stage/generate.sh
 returncode_list_mismatch  | dfc4ad0f | local.sh   | FAIL     |            2 | /Users/siddiq90/Documents/buildtest/var/tests/local.sh/pass_returncode/returncode_list_mismatch/5/stage/generate.sh
 returncode_int_match      | 0a6555c3 | local.sh   | PASS     |          128 | /Users/siddiq90/Documents/buildtest/var/tests/local.sh/pass_returncode/returncode_int_match/3/stage/generate.sh

+-----------------------+
| Stage: Test Summary  |
+-----------------------+

Executed 4 tests
Passed Tests: 2/4 Percentage: 50.000%
Failed Tests: 2/4 Percentage: 50.000%
```

Customize Shell

- The **shell** property can be used to customize shell and shell options that are passed to test.
- The default shell is /bin/bash

shell

Specify a shell launcher to use when running jobs. This sets the shebang line in your test script. The `shell` key can be used with `run` section to describe content of script and how its executed

`shell`

- is optional
- Type: `string`
- cannot be null
- defined in: script schema version 1.0

shell Type

`string`

shell Constraints

pattern: the string must match the following regular expression:

```
^(/bin/bash|/bin/sh|/bin/csh|/bin/tcsh|/bin/zsh|bash|sh|csh|tcsh|zsh|python).*
```

```
version: "1.0"
buildspecs:
  _bin_sh_shell:
    executor: local.sh
    type: script
    description: "/bin/sh shell example"
    shell: /bin/sh
    tags: [tutorials]
    run: "bzip2 --help"

  _bin_bash_shell:
    executor: local.bash
    type: script
    description: "/bin/bash shell example"
    shell: /bin/bash
    tags: [tutorials]
    run: "bzip2 -h"

  bash_shell:
    executor: local.bash
    type: script
    description: "bash shell example"
    shell: bash
    tags: [tutorials]
    run: "echo $SHELL"

  sh_shell:
    executor: local.sh
    type: script
    description: "sh shell example"
    shell: sh
    tags: [tutorials]
    run: "echo $SHELL"

  shell_options:
    executor: local.sh
    type: script
    description: "shell options"
    shell: "sh -x"
    tags: [tutorials]
    run: |
      echo $SHELL
      hostname
```

Python Shell

- The **run** property can be used for writing shell commands or it can be used for writing python scripts.
- To enable python scripts use shell: python and one must use the python executor
- For more complex python scripts, it's recommended one develops a python script and invoke the python script using bash/sh shell.

```
version: "1.0"
buildspecs:
  circle_area:
    executor: local.python
    type: script
    shell: python
    description: "Calculate circle of area given a radius"
    tags: [tutorials, python]
    run: |
      import math
      radius = 2
      area = math.pi * radius * radius
      print("Circle Radius ", radius)
      print("Area of circle ", area)
```

Python Code

Scheduler Agnostic Configuration

- Buildtest provides a scheduler agnostic configuration through batch field.
- The batch field implements a subset of options supported by bsub, sbatch, and qsub options that are shared between LSF, Slurm and Cobalt.

Batch Translation Table

Field	Slurm	LSF	Cobalt
account	-account	-P	-project
begin	-begin	-b	N/A
cpucount	-ntasks	-n	-proccount
email-address	-mail-user	-u	-notify
exclusive	-exclusive=user	-x	N/A
memory	-mem	-M	N/A
network	-network	-network	N/A
nodecount	-nodes	-nnodes	-nodecount
qos	-qos	N/A	N/A
queue	-partition	-q	-queue
tasks-per-core	-ntasks-per-core	N/A	N/A
tasks-per-node	-ntasks-per-node	N/A	N/A
tasks-per-socket	-ntasks-per-socket	N/A	N/A
timelimit	-time	-W	-time

```
version: "1.0"
buildspecs:
  sleep:
    type: script
    executor: slurm.normal
    description: sleep 2 seconds
    tags: [tutorials]
    batch:
      nodecount: "1"
      cpucount: "1"
      timelimit: "5"
      memory: "5MB"
      exclusive: true

    vars:
      SLEEP_TIME: 2
    run: sleep $SLEEP_TIME
```

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --time=5
#SBATCH --mem=5MB
#SBATCH --exclusive=user
source /home1/06908/sms1990/buildtest/var/executors/slurm.normal/before_script.sh
SLEEP_TIME=2
sleep $SLEEP_TIME
```

Cray Burst Buffer and Data Warp Support

- Cray systems, we can access burst buffers using **BB** and **DW** property.
- In this example we create a persistent burst buffer named **databuffer** of size 10GB with striped access.

```
Output File
_____________________________
/var/opt/cray/dws/mounts/batch/databuffer_35693664_striped_scratch
total 5.0G
-rw-rw---- 1 siddiq90 siddiq90 5.0G Oct 29 13:06 random.txt



Error File
_____________________________
5+0 records in
5+0 records out
5368709120 bytes (5.4 GB, 5.0 GiB) copied, 90.6671 s, 59.2 MB/s
```

```
$ scontrol show burst | grep databuffer
    Name=databuffer CreateTime=2020-10-29T13:06:21 Pool=wlm_pool Size=20624MiB State=allocated UserID=siddiq90(92503)
```

```
#!/bin/bash
#SBATCH -C knl
#SBATCH --nodes=1
#SBATCH --time=5
#SBATCH --ntasks=1
#SBATCH --job-name=create_burst_buffer
#SBATCH --output=create_burst_buffer.out
#SBATCH --error=create_burst_buffer.err
#BB create_persistent name=databuffer capacity=10GB access_mode=striped type=scratch
#DW persistentdw name=databuffer
source /global/u1/s/siddiq90/buildtest/var/executors/slurm.debug/before_script.sh
cd $DW_PERSISTENT_STRIPED_databuffer
pwd
dd if=/dev/urandom of=random.txt bs=1G count=5 iflag=fullblock
ls -lh $DW_PERSISTENT_STRIPED_databuffer/

source /global/u1/s/siddiq90/buildtest/var/executors/slurm.debug/after_script.sh
```

```
buildspecs:
  create_burst_buffer:
    type: script
    executor: slurm.debug
    batch:
      nodecount: "1"
      timelimit: "5"
      cpucount: "1"
    sbatch: ["-C knl"]
    description: Create a burst buffer
    tags: [jobs]
    BB:
      - create_persistent name=databuffer capacity=10GB access_mode=striped type=scratch
    DW:
      - persistentdw name=databuffer
    run: |
      cd $DW_PERSISTENT_STRIPED_databuffer
      pwd
      dd if=/dev/urandom of=random.txt bs=1G count=5 iflags=fullblock
      ls -lh $DW_PERSISTENT_STRIPED_databuffer/
```

Compiler Selection and Compiler Defaults

- This test will be built using gcc@10.2.0 and gcc@9.3.0
- Compilers are defined in buildtest configuration, one can retrieve compilers using **buildtest config compilers**

```
$ buildtest config compilers -l
builtin_gcc
gcc@10.2.0
gcc@9.3.0
```

```
version: "1.0"
buildspecs:
  vecadd_gnu:
    type: compiler                                                    Compiler Schema
    description: Vector Addition example with GNU compiler
    tags: [tutorials, compile]
    executor: local.bash
    source: src/vecAdd.c                                              Source File
    compilers:                                                        Start of Compiler Block
      name: ["^(gcc)"]                                                Select Compilers based on Regular Expression
      default:                                                        Default Section for compilers organized by compiler groups
        gcc:                                                          Default Section for gcc compilers
          cflags: -fopenacc                                           Set cflags
          ldflags: -lm                                                Set ldflags
```

Override Compiler Default

- Compiler defaults can be overridden via **config** section. This is organized by named compilers defined in buildtest setting.
- Buildtest will ignore compiler in **config** if it's not picked up in regular expression.
- In this example **builtin_gcc** will use default cflags: -O1 while **gcc@9.3.0** will use –O2 and **gcc@10.2.0** will use –O3

```
version: "1.0"
buildspecs:
  hello_c:
    type: compiler
    description: "Hello World C Compilation"
    executor: local.bash
    tags: [tutorials, compile]
    source: "src/hello.c"
    compilers:
      name: ["^(builtin_gcc|gcc)"]
      default:
        gcc:
          cflags: -O1
      config:
        gcc@9.3.0:
          cflags: -O2
        gcc@10.2.0:
          cflags: -O3
```

Multi Compiler Test

- This OpenMP reduction example is built with all gcc, intel and cray modules.
- OpenMP support for gcc, intel and cray differ slightly this is defined in compiler group.
- The default **all** defines configuration inherited by all compiler groups, in this case all tests sets environment OMP_NUM_THREADS to 4.
- Properties in **all** can be overridden at compiler group or named compiler.

```
+----------------------+
| Stage: Building Test |
+----------------------+
```

```yaml
version: "1.0"
buildspecs:
  reduction:
    type: compiler
    executor: local.bash
    source: src/reduction.c
    description: OpenMP reduction example using gcc, intel and cray compiler
    tags: [openmp]
    compilers:
      name: ["^(gcc|intel|PrgEnv-cray)"]
      default:
        all:
          env:
            OMP_NUM_THREADS: 4
        gcc:
          cflags: -fopenmp
        intel:
          cflags: -qopenmp
        cray:
          cflags: -h omp
```

name	id	type	executor	tags	compiler	testpath
reduction	4eb31800	compiler	local.bash	['openmp']	gcc/6.1.0	/global/u1/s/siddiq90/buildtest/var/tests/local.bash/reduction/reduction/72/stage/generate.sh
reduction	514a32a1	compiler	local.bash	['openmp']	gcc/7.3.0	/global/u1/s/siddiq90/buildtest/var/tests/local.bash/reduction/reduction/73/stage/generate.sh
reduction	9bb7a57c	compiler	local.bash	['openmp']	gcc/8.1.0	/global/u1/s/siddiq90/buildtest/var/tests/local.bash/reduction/reduction/74/stage/generate.sh
reduction	91e61ba6	compiler	local.bash	['openmp']	gcc/8.2.0	/global/u1/s/siddiq90/buildtest/var/tests/local.bash/reduction/reduction/75/stage/generate.sh
reduction	f6a8d54e	compiler	local.bash	['openmp']	gcc/8.3.0	/global/u1/s/siddiq90/buildtest/var/tests/local.bash/reduction/reduction/76/stage/generate.sh
reduction	29490f3a	compiler	local.bash	['openmp']	gcc/9.3.0	/global/u1/s/siddiq90/buildtest/var/tests/local.bash/reduction/reduction/77/stage/generate.sh
reduction	5e58e1cf	compiler	local.bash	['openmp']	gcc/10.1.0	/global/u1/s/siddiq90/buildtest/var/tests/local.bash/reduction/reduction/78/stage/generate.sh
reduction	a4e696d3	compiler	local.bash	['openmp']	gcc/6.3.0	/global/u1/s/siddiq90/buildtest/var/tests/local.bash/reduction/reduction/79/stage/generate.sh
reduction	c571b53e	compiler	local.bash	['openmp']	gcc/8.1.1-openacc-gcc-8-branch-20190215	/global/u1/s/siddiq90/buildtest/var/tests/local.bash/reduction/reduction/80/stage/generate.sh
reduction	b7cba893	compiler	local.bash	['openmp']	PrgEnv-cray/6.0.5	/global/u1/s/siddiq90/buildtest/var/tests/local.bash/reduction/reduction/81/stage/generate.sh
reduction	67f9d327	compiler	local.bash	['openmp']	PrgEnv-cray/6.0.7	/global/u1/s/siddiq90/buildtest/var/tests/local.bash/reduction/reduction/82/stage/generate.sh
reduction	16713092	compiler	local.bash	['openmp']	PrgEnv-cray/6.0.9	/global/u1/s/siddiq90/buildtest/var/tests/local.bash/reduction/reduction/83/stage/generate.sh
reduction	f5982111	compiler	local.bash	['openmp']	intel/19.0.3.199	/global/u1/s/siddiq90/buildtest/var/tests/local.bash/reduction/reduction/84/stage/generate.sh
reduction	c2b22eff	compiler	local.bash	['openmp']	intel/19.1.2.254	/global/u1/s/siddiq90/buildtest/var/tests/local.bash/reduction/reduction/85/stage/generate.sh
reduction	e3f6faa4	compiler	local.bash	['openmp']	intel/16.0.3.210	/global/u1/s/siddiq90/buildtest/var/tests/local.bash/reduction/reduction/86/stage/generate.sh
reduction	d95a3883	compiler	local.bash	['openmp']	intel/17.0.1.132	/global/u1/s/siddiq90/buildtest/var/tests/local.bash/reduction/reduction/87/stage/generate.sh
reduction	0aee1fee	compiler	local.bash	['openmp']	intel/17.0.2.174	/global/u1/s/siddiq90/buildtest/var/tests/local.bash/reduction/reduction/88/stage/generate.sh
reduction	853d3ff4	compiler	local.bash	['openmp']	intel/18.0.1.163	/global/u1/s/siddiq90/buildtest/var/tests/local.bash/reduction/reduction/89/stage/generate.sh
reduction	0e66bc4a	compiler	local.bash	['openmp']	intel/18.0.3.222	/global/u1/s/siddiq90/buildtest/var/tests/local.bash/reduction/reduction/90/stage/generate.sh
reduction	69826793	compiler	local.bash	['openmp']	intel/19.0.0.117	/global/u1/s/siddiq90/buildtest/var/tests/local.bash/reduction/reduction/91/stage/generate.sh
reduction	f67d8953	compiler	local.bash	['openmp']	intel/19.0.8.324	/global/u1/s/siddiq90/buildtest/var/tests/local.bash/reduction/reduction/92/stage/generate.sh
reduction	e12ac611	compiler	local.bash	['openmp']	intel/19.1.0.166	/global/u1/s/siddiq90/buildtest/var/tests/local.bash/reduction/reduction/93/stage/generate.sh
reduction	fc8386f4	compiler	local.bash	['openmp']	intel/19.1.1.217	/global/u1/s/siddiq90/buildtest/var/tests/local.bash/reduction/reduction/94/stage/generate.sh
reduction	80e39fa5	compiler	local.bash	['openmp']	intel/19.1.2.275	/global/u1/s/siddiq90/buildtest/var/tests/local.bash/reduction/reduction/95/stage/generate.sh
reduction	b9181f22	compiler	local.bash	['openmp']	intel/19.1.3.304	/global/u1/s/siddiq90/buildtest/var/tests/local.bash/reduction/reduction/96/stage/generate.sh

Filter and Format buildspec cache

- We can filter and format buildspec cache using –filter and –format option.
- The filter option expects a list of key=value pair separated by comma.
- To see list of all filter and format fields we can use –helpfilter and –helpformat option

```
$ buildtest buildspec find --filter tags=fail
+----------------------+---------+------------+----------------------+-----------------------------------------------------------+
| Name                 | Type    | Executor   | Tags                 | Description                                               |
+======================+=========+============+======================+===========================================================+
| exit1_fail           | script  | local.sh   | ['tutorials', 'fail'] | exit 1 by default is FAIL                                 |
+----------------------+---------+------------+----------------------+-----------------------------------------------------------+
| returncode_mismatch  | script  | local.sh   | ['tutorials', 'fail'] | exit 2 failed since it failed to match returncode 1       |
+----------------------+---------+------------+----------------------+-----------------------------------------------------------+
```

```
(buildtest) bash-3.2$ buildtest buildspec find --filter tags=fail --format name,tags
+--------------------------+-----------------------+
| name                     | tags                  |
+==========================+=======================+
| exit1_fail               | ['tutorials', 'fail'] |
+--------------------------+-----------------------+
| returncode_list_mismatch | ['tutorials', 'fail'] |
+--------------------------+-----------------------+
```

Multi key filter is evaluated as logical AND.

```
$ buildtest buildspec find --filter tags=tutorials,executor=local.sh,type=script
+----------------------+---------+------------+----------------------+-----------------------------------------------------------+
| Name                 | Type    | Executor   | Tags                 | Description                                               |
+======================+=========+============+======================+===========================================================+
| _bin_sh_shell        | script  | local.sh   | ['tutorials']        | /bin/sh shell example                                     |
+----------------------+---------+------------+----------------------+-----------------------------------------------------------+
| sh_shell             | script  | local.sh   | ['tutorials']        | sh shell example                                          |
+----------------------+---------+------------+----------------------+-----------------------------------------------------------+
| shell_options        | script  | local.sh   | ['tutorials']        | shell options                                             |
+----------------------+---------+------------+----------------------+-----------------------------------------------------------+
| exit1_fail           | script  | local.sh   | ['tutorials', 'fail'] | exit 1 by default is FAIL                                 |
+----------------------+---------+------------+----------------------+-----------------------------------------------------------+
| exit1_pass           | script  | local.sh   | ['tutorials', 'pass'] | report exit 1 as PASS                                     |
+----------------------+---------+------------+----------------------+-----------------------------------------------------------+
| returncode_mismatch  | script  | local.sh   | ['tutorials', 'fail'] | exit 2 failed since it failed to match returncode 1       |
+----------------------+---------+------------+----------------------+-----------------------------------------------------------+
```

```
$ buildtest buildspec find --helpfilter
Field     Description              Type
--------  -----------------------  ------
executor  Filter by executor name  STRING
tags      Filter by tag name       STRING
type      Filter by schema type    STRING
```

```
$ buildtest buildspec find --helpformat
Field        Description
-----------  ----------------------
name         Format by test name
tags         Format by tag name
type         Format by schema type
executor     Format by executor type
description  Format by description
file         Format by file
```

Query Test Reports with Filter and Format Examples

- We provide access to test reports through CLI. The reports are stored in JSON file for post-processing.
- The buildtest report will display all test results which can be queried with filter and format options.
- The **–filter** option are passed as key=value pair
- Multiple filter arguments can be delimited by comma separator and buildtest will treat multiple filter argument as a logical AND operation
- The **–format** option alter the columns in the report tables.

```
$ buildtest report --filter state=FAIL,executor=local.sh --format=name,id,state,executor
+--------------------------+-----------+---------+------------+
| name                     | id        | state   | executor   |
+==========================+===========+=========+============+
| exit1_fail               | bff47f17  | FAIL    | local.sh   |
+--------------------------+-----------+---------+------------+
| exit1_fail               | 527880f9  | FAIL    | local.sh   |
+--------------------------+-----------+---------+------------+
| exit1_fail               | dce0c689  | FAIL    | local.sh   |
+--------------------------+-----------+---------+------------+
| exit1_fail               | 4acbbb41  | FAIL    | local.sh   |
+--------------------------+-----------+---------+------------+
| exit1_fail               | d551071e  | FAIL    | local.sh   |
+--------------------------+-----------+---------+------------+
| exit1_fail               | 1b7337d4  | FAIL    | local.sh   |
+--------------------------+-----------+---------+------------+
| returncode_list_mismatch | 30a6c390  | FAIL    | local.sh   |
+--------------------------+-----------+---------+------------+
| returncode_list_mismatch | 3c781f83  | FAIL    | local.sh   |
+--------------------------+-----------+---------+------------+
| returncode_list_mismatch | 701c31ab  | FAIL    | local.sh   |
+--------------------------+-----------+---------+------------+
| returncode_list_mismatch | 6222b488  | FAIL    | local.sh   |
+--------------------------+-----------+---------+------------+
| returncode_list_mismatch | c5d00af1  | FAIL    | local.sh   |
+--------------------------+-----------+---------+------------+
| returncode_list_mismatch | dfc4ad0f  | FAIL    | local.sh   |
+--------------------------+-----------+---------+------------+
```

```
$ buildtest report --filter name=exit1_pass --format=name,id,returncode,state
+------------+----------+--------------+---------+
| name       | id       |   returncode | state   |
+============+==========+==============+=========+
| exit1_pass | b25ede10 |            1 | PASS    |
+------------+----------+--------------+---------+
| exit1_pass | 13c3cde8 |            1 | PASS    |
+------------+----------+--------------+---------+
| exit1_pass | 725f66de |            1 | PASS    |
+------------+----------+--------------+---------+
| exit1_pass | c4d22774 |            1 | PASS    |
+------------+----------+--------------+---------+
```

```
$ buildtest report --filter returncode=2 --format=name,id,returncode
+--------------------------+----------+--------------+
| name                     | id       |   returncode |
+==========================+==========+==============+
| returncode_list_mismatch | 30a6c390 |            2 |
+--------------------------+----------+--------------+
| returncode_list_mismatch | 3c781f83 |            2 |
+--------------------------+----------+--------------+
| returncode_list_mismatch | 701c31ab |            2 |
+--------------------------+----------+--------------+
| returncode_list_mismatch | 6222b488 |            2 |
+--------------------------+----------+--------------+
| returncode_list_mismatch | c5d00af1 |            2 |
+--------------------------+----------+--------------+
| returncode_list_mismatch | dfc4ad0f |            2 |
+--------------------------+----------+--------------+
```

Cori Test Suite

Category	Description
System	Filesystem, mountpoint check, timezone, ping gpfs nodes, /etc/profile.d/ scripts, os release, ulimits, time test
Filesystem	gpfs, lustre, cvmfs, filesystem benchmarks
Network	Ping nodes (login, dtn, gerty), ssh test on login nodes, nslookup, ssh host authentication, nameservers
Tools	iris, sqs, jobstats, myquota
Slurm	sinfo, scontrol, sacctmgr, squeue, ping slurm controller, partitions, esslurm
Jobs	Hostname to all QOS, submit to esslurm, timeout, exit1, OOM, create burstbuffer, stage-in to burst buffer, fail jobs on time-limit/max nodes by queues
Apps	OpenACC, OpenMP, MPI, bupc, upc, Spack, darshan, gpuquery, MKL, STREAM, Serial Hello, shifter pull image, shifter job, E4S Testsuite, Lmodule

Cori Test Suite: https://github.com/buildtesters/buildtest-cori

SSH, Ping and Uptime Test

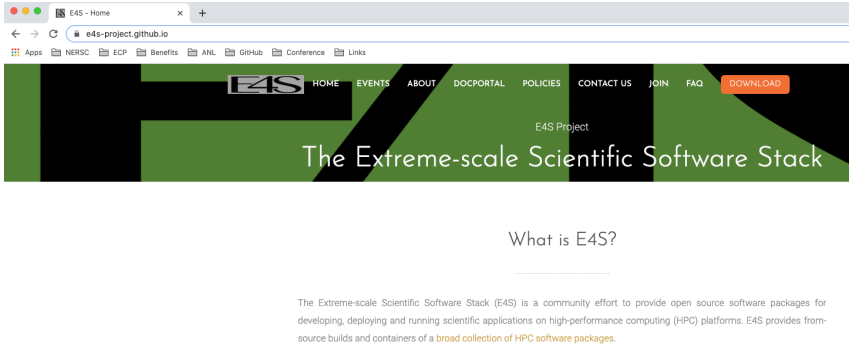
```
version: "1.0"
buildspecs:
  login_nodes:
    executor: local.bash
    description: Confirm all Cori Login Nodes are accessible via ping
    tags: ["system", "network"]
    type: script
    run: |
      ping -c 1 -W 20 cori01
      ping -c 1 -W 20 cori02
      ping -c 1 -W 20 cori03
      ping -c 1 -W 20 cori04
      ping -c 1 -W 20 cori05
      ping -c 1 -W 20 cori06
      ping -c 1 -W 20 cori07
      ping -c 1 -W 20 cori08
      ping -c 1 -W 20 cori09
      ping -c 1 -W 20 cori10
      ping -c 1 -W 20 cori11
      ping -c 1 -W 20 cori12

  data_transfer_nodes:
    executor: local.bash
    description: Confirm all Cori Data Transfer Nodes are accessible via ping
    type: script
    tags: ["system", "network"]
    run: |
      ping -c 1 -W 20 dtn01
      ping -c 1 -W 20 dtn02
      ping -c 1 -W 20 dtn03
      ping -c 1 -W 20 dtn04
      ping -c 1 -W 20 dtn05
      ping -c 1 -W 20 dtn06

  uptime_login_nodes:
    executor: local.bash
    description: Run uptime across all login nodes
    tags: ["system"]
    type: script
    run: |
      pdsh -w cori[01-12] uptime


  uptime_data_transfer_nodes:
    executor: local.bash
    description: Run uptime across all data transfer nodes
    tags: ["system"]
    type: script
    run: |
      pdsh -w dtn[01-06] uptime
```

```
version: "1.0"
buildspecs:
  ssh_login_nodes:
    type: script
    executor: local.bash
    tags: [system, network]
    description: "test ssh connection to login nodes"
    run: |
      ssh -q cori01 hostname
      ssh -q cori02 hostname
      ssh -q cori03 hostname
      ssh -q cori04 hostname
      ssh -q cori05 hostname
      ssh -q cori06 hostname
      ssh -q cori07 hostname
      ssh -q cori08 hostname
      ssh -q cori09 hostname
      ssh -q cori10 hostname
      ssh -q cori11 hostname
      ssh -q cori12 hostname

  ssh_data_transfer_nodes:
    type: script
    executor: local.bash
    tags: [system, network]
    description: "test ssh connection to data transfer nodes"
    run: |
      ssh -q dtn01 hostname
      ssh -q dtn02 hostname
      ssh -q dtn03 hostname
      ssh -q dtn04 hostname
      ssh -q dtn05 hostname
      ssh -q dtn06 hostname

  ssh_gerty:
    type: script
    executor: local.bash
    description: "test ssh connection to gerty"
    tags: [system, network]
    run: ssh -q gerty01 hostname
```


NERSC
E4S TestSuite on Cori

E4S Test Suite

- Extreme-scale Scientific Software Stack (E4S) is a collection of spack packages built and tested on several platforms. E4S is deployed as spack manifest (spack.yaml), containers, and buildcache.
- The E4S Test Suite is a collection of tests to validate E4S stack and increase test coverage for deployed stack.
- The main script test-all.sh can be run as standalone program which will test everything or you can specify an argument to a directory of tests to run. In example below we run the qthreads test on Cori.




```
git clone https://github.com/E4S-Project/testsuite.git
cd testsuite
source /global/common/software/spackecp/luke-wyatt-testing/spack/share/spack/setup-env.sh
source setup.sh
./test-all.sh ./validation_tests/qthreads
```

```
siddiq90@cori02:~/sw/testsuite> ./test-all.sh ./validation_tests/qthreads
===
./validation_tests/qthreads
Cleaning /global/homes/s/siddiq90/sw/testsuite/validation_tests/qthreads
Compiling /global/homes/s/siddiq90/sw/testsuite/validation_tests/qthreads
Running /global/homes/s/siddiq90/sw/testsuite/validation_tests/qthreads
Success
```

Cori E4S Testing Strategy

- We have deployed E4S 20.10 release for Cori with up to 135 installed specs see https://docs.nersc.gov/applications/e4s/ for more details
- We can leverage upstream E4S testsuite (https://github.com/E4S-Project/testsuite) to validate the e4s stack since it provides majority of the tests.
- E4S tests are integrated into Cori testsuite at https://github.com/buildtesters/buildtest-cori/tree/master/e4s and run using buildtest via gitlab pipeline.
- Next release of E4S on Cori will utilize **spack test** to run tests. However, will want to develop site specific tests that utilize batch queue system.
- We can run all e4s tests using the **e4s** tags

```
buildtest build --tags e4s
```

Custom Executor for e4s

- We have a custom executor **slurm.e4s** to run e4s stack through slurm scheduler.
- The **before_script** that is sourced for all tests that utilize executor **slurm.e4s** executor.

```
e4s:
  description: E4S runner
  cluster: cori
  max_pend_time: 20000
  options:
  - -q regular
  - -C knl
  - -t 10
  - -n 4
  before_script: |
    source /global/common/software/spackecp/e4s-20.10/spack/share/spack/setup-env.sh
    cd $SCRATCH/testsuite
    source setup.sh
```

```
version: "1.0"
buildspecs:
  e4s_adios2:
    type: script
    executor: slurm.e4s
    description: Run adios2 test from E4S Testsuite
    tags: [e4s]
    run: $SCRATCH/testsuite/test-all.sh $SCRATCH/testsuite/validation_tests/adios2
    status:
      regex:
        stream: stderr
        exp: "Success"
```

Cori E4S 20.10 Pipeline

```yaml
variables:
  SCHEDULER_PARAMETERS: "-C haswell -M escori -q xfer -N1 -t 12:00:00"
  SPACK_REPOSITORY: https://github.com/spack/spack.git
  COMMIT_SHA: "e1e0bbb4cbe11a3f0d7e50466ffa86071ee653b7"

stages:
  - generate
  - build
  - deploy
  - post-test
```

Remove buildcache

```yaml
remove_buildcache:
  stage: generate
  tags: [cori]
  only:
    variables:
      - $REMOVE_BUILDCACHE == "True"
  script:
    rm -rf /global/common/software/spackecp/mirrors/e4s-2020-10/
```

Generate Pipeline via **spack ci**

```yaml
generate-spack-pipeline:
  stage: generate
  tags: [cori]
  only:
    variables:
      - $SPACK_BUILD_E4S == "True"
    changes:
      - spack.yaml
  before_script:
  - pwd
  - git clone ${SPACK_REPOSITORY}
  - cd spack
  - git reset ${COMMIT_SHA} --hard
  - git log  --oneline -1
  - . share/spack/setup-env.sh
  - echo $SPACK_ROOT
  - cd $CI_PROJECT_DIR
  script:
  - hostname
  - echo $SPACK_ROOT $SPACK_REPOSITORY
  - cd $CI_PROJECT_DIR
  - rm -rf $HOME/.spack
  - spack compiler find
  - spack compiler list
  - spack env activate .
  - spack env st
  - export SPACK_GNUPGHOME=$HOME/.gnupg
  - spack gpg list
  - spack concretize -f 2>&1
  - spack ci generate --dependencies --optimize  --output-file ${CI_PROJECT_DIR}/jobs_scratch_dir/pipeline.yml
  after_script:
  - echo $SPACK_ROOT
  - rm -rf $SPACK_ROOT
  artifacts:
    paths:
    - jobs_scratch_dir/pipeline.yml
```

Build E4S stack

```yaml
build-e4s:
  stage: build
  only:
    variables:
      - $SPACK_BUILD_E4S == "True"
    changes:
      - "spack.yaml"
  trigger:
    include:
    - artifact: "jobs_scratch_dir/pipeline.yml"
      job: generate-spack-pipeline
    strategy: depend
```

Deploy Stack

```yaml
deploy-e4s:
  stage: deploy
  tags: [cori]
  only:
    variables:
      - $DEPLOY_STACK == "True"
    changes:
      - "prod/spack.yaml"
  script:
    - cd /global/common/software/spackecp/e4s-20.10
    - rm -rf spack
    - git clone ${SPACK_REPOSITORY}
    - cd spack
    - git reset ${COMMIT_SHA} --hard
    - export SPACK_GNUPGHOME=$HOME/.gnupg
    - . share/spack/setup-env.sh
    - cd $CI_PROJECT_DIR/prod
    - spack env create e4s-2010 spack.yaml
    - spack env activate e4s-2010
    - spack env st
    - spack buildcache update-index -d /global/common/software/spackecp/mirrors/e4s-2020-10
    - spack buildcache list -L
    - spack install --cache-only
    #- spack buildcache list -L | awk '{print $1}' | tail -n +2 | while read line; do spack install --cache-only /$line; done
    - spack module tcl refresh --delete-tree -y
    - spack find
```

Run Post Tests

```yaml
post-test-e4s:
  stage: post-test
  tags: [cori]
  only:
    variables:
      - $POST_TEST == "True"
  script:
    - git clone --depth 1 --branch v0.9.1 https://github.com/buildtesters/buildtest
    - cd buildtest && git log --pretty=oneline -1 && cd $CI_PROJECT_DIR
    - git clone --depth 1 https://github.com/buildtesters/buildtest-cori
    - module load python
    - source buildtest/setup.sh
    - buildtest --version
    - cp buildtest-cori/.buildtest/config.yml $HOME/.buildtest/config.yml
    - buildtest buildspec find --root $CI_PROJECT_DIR/buildtest-cori
    - buildtest buildspec find --filter tags=e4s
    - buildtest build --tags e4s
    - buildtest report --filter tags=e4s --format name,id,tags,executor,starttime,endtime,runtime,state,returncode,command,buildspec
```

Build all test with e4s tags

2020 Summary of Updates

- Introduce JSON Schema for validating buildspecs and buildtest configuration
- We publish JSON Schema, schema examples, and schema docs on GitHub pages at https://buildtesters.github.io/buildtest/
- Remove buildtest module features into Lmodule project - https://github.com/buildtesters/lmodule
- Move regression tests from Travis to Github workflow
- Add batch queue support for Slurm, LSF and Cobalt
- Move github organization from https://github.com/HPC-buildtest/buildtest/ to https://github.com/buildtesters/buildtest/
- Add gitlab CI checks to run regression test on Cori

Conclusion

- Buildtest is a testing framework to enable HPC sites to write acceptance tests. Users need basic understanding of YAML in order to write buildspecs
- Buildtest is **not** a testsuite, facilities will need to develop tests applicable for their system. There are up to 100+ tests in Cori testsuite: https://github.com/buildtesters/buildtest-cori to help other facilities to get started.
- For additional help please join Slack (self-invite): https://hpcbuildtest.herokuapp.com/ or post issue at https://github.com/buildtesters/buildtest/issues
- References:
  - Buildtest Docs: https://buildtest.readthedocs.io/en/latest/index.html
  - Schema Docs: https://buildtesters.github.io/buildtest/
  - GitHub: https://github.com/buildtesters/buildtest
  - Installing buildtest: https://buildtest.readthedocs.io/en/latest/installing_buildtest.html
  - Getting Started: https://buildtest.readthedocs.io/en/latest/getting_started.html
  - Writing Buildspecs: https://buildtest.readthedocs.io/en/devel/writing_buildspecs.html
  - References: https://buildtest.readthedocs.io/en/latest/references.html
  - Slack Channel: http://hpcbuildtest.slack.com/
  - API: https://buildtest.readthedocs.io/en/latest/api/index.html

Thank You