

Workflow Configuration Import and Validation for AliECS

Progress Report II
26th August, 2020

By : Ayaan Zaidi
Mentored by : Teo Mrnjavac

Overview

- Convert a DPL Dump generated by O2/DPL into required number of task templates and one workflow template
- Develop a **package to validate** said templates against schemas



walnut



Validation

Recap

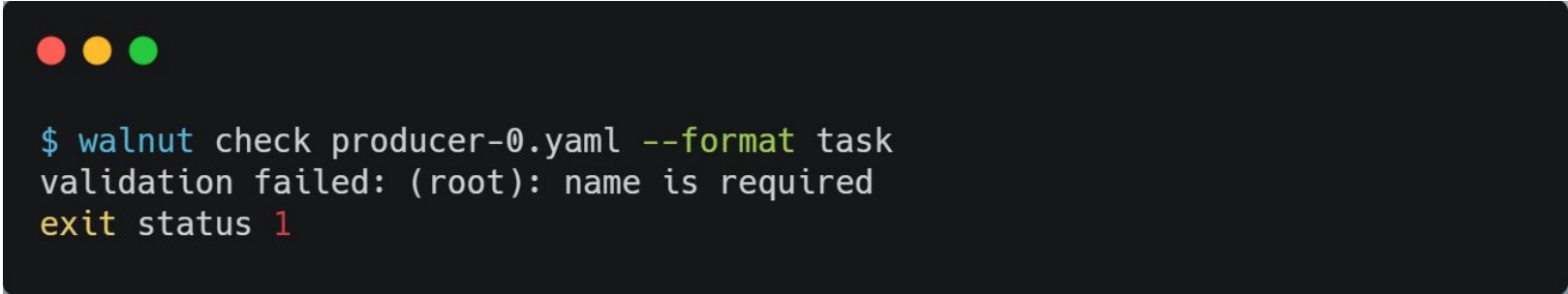
- Two schemas (one for workflow templates and one for task templates) were defined.
- Package `schemata` was built that allows the user to verify if a workflow or task template adheres to the aforementioned schema.

Updates

- Refinements to existing schemas, addition of flags like `enabled`, `trigger`, `timeout` and `critical`
- `schemata` package now outputs why the validation failed
- Available on the [walnut branch of AliceO2Group/Control](#).

Validation – Example

Upon successful validation, the process exits cleanly. If validation fails, walnut exits with exit code 1 and shows the reason for failure:

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. The text inside the terminal is as follows:

```
$ walnut check producer-0.yaml --format task
validation failed: (root): name is required
exit status 1
```

Conversion

Recap

- Convert an input DPL dump to workflow and task template formats that AliECS can work with.
- Ensure that any DPL dump can be converted with **minimal or no additional input** from the user.

```
{
  "workflow": [
    {
      "name": "producer-0",
      "inputs": [],
      "outputs": [
        {
          "binding": "out",
          "origin": "TST",
          "description": "RAWDATA",
          "subspec": 0,
          "lifetime": 0
        }
      ]
    },
    "options": [],
    "rank": 0,
    "nSlots": 1,
    "inputTimeSliceId": 0,
    "maxInputTimeslices": 1
  ]
}
```

Conversion + Grafting

Goals

- Given an existing Workflow template, read a fresh DPL dump, convert it on the fly and append its contents to the said workflow template.
- Grafting should occur at the specified level in the existing workflow template
- Desired format:

```
--graft "/path/to/readout-stfb.yaml:readout-stfb.host-{{ it }}"
```

```
name: readout-stfb
defaults:
  roc_ctp_emulator_enabled: "true"
  stfb_enabled: "true"
roles:
- name: host-{{ it }}
  for:
    range: "{{ hosts }}"
    var: it
  constraints:
    - attribute: machine_id
      value: "{{ it }}"
  roles:
    - name: "readout"
      task:
        load: readout
    - name: "stfb"
      enabled: "{{ stfb_enabled }}"
      connect:
        - name: readout
```

Conversion + Grafting: Implementation

Implementation

- Grafting begins by converting the provided DPL dump into a workflow template (not task!)
- Search for the target role inside provided DPL dump by the help of `yaml.Node` (alternate YAML implementation)
- `yaml.Node` allows us to traverse the syntax of an existing YAML document and insert a structure while preserving ordering and comments.

```
name: dump
roles:
  - bind:
      - name: from_producer-0_to_Dispatcher
        type: push
        transport: shmем
        addressing: ipc
    connect:
      - name: from_internal-dpl-clock_to_producer-0
        type: pull
        transport: shmем
        target: '{{ Parent().Path }}.internal-dpl-clock:from_internal-dpl-clock_to_producer-0'
    defaults:
      dpl_config: ""
      dump_config_uri: ""
      dump_monitoring_url: no-op://
      user: flp
```

Conversion + Grafting: Example

The user can specify one or more DPL dumps to convert and graft to an existing workflow template:

```
$ walnut convert dump.json --graft "/path/to/readout-stfb.yaml:readout-stfb.host-{{ it }}" --workflow-name grafted
```

The above can be read as:

- walnut should read `dump.json` (a DPL dump)
- graft its contents inside an existing WFT called `readout-stfb`
- as a child of `host-{{ it }}` which is a child of `readout-stfb`

All the code can be found at [AliceO2Group/Control](#).

Grafting Demo

```
obviiyus@chessboard: /mnt/c/U x + v
[obviiyus@chessboard walnut]$ vim readout-stfb.yaml
[obviiyus@chessboard walnut]$ ./walnut convert dump.json --graft "readout-stfb.yaml:readout-stfb.host-{{ it }}" --workflow-name grafted
OPENED: dump.jsonWriting to: /mnt/c/Users/zaidi/Desktop/Control/cmd/walnut/grafted.yaml
On branch OCTRL-311
Changes not staged for commit:
  (use "git add <file>.." to update what will be committed)
  (use "git restore <file>.." to discard changes in working directory)
    modified:   ../../go.mod
    modified:   ../../go.sum

Untracked files:
  (use "git add <file>.." to include in what will be committed)
    dump.json
    grafted.yaml
    readout-stfb.yaml
    tasks/
    walnut
    walnut.exe
    workflows/
    ../../walnut/converter/dump/
    ../../walnut/converter/test/

no changes added to commit (use "git add" and/or "git commit -a")
? Would you like to view the git diff? No
[obviiyus@chessboard walnut]$ vim |
```

Future Developments

- Next and final part of the project, adding **support for dangling inputs & outputs** (i.e. channels for which there is no corresponding target within the DPL workflow)
- Minor clean up and improvements to schemata logic

Closing thoughts

- Final week working with CERN
- Had little experience with Go before Google Summer of Code
- One of the best learning opportunities I've been exposed to
- Would've love to work further, specifically on:
 - commit hooks to run validation on all templates uploaded to [ControlWorkflows](#)
 - Preserve fixed ordering of fields during YAML marshaling

The Numbers!

- **90** day long endeavour
- **16** pull requests, **129** commits merged into [AliceO2Group/Control](#)
- **6,800** lines of code additions and **3,400** deletions

Thank you.