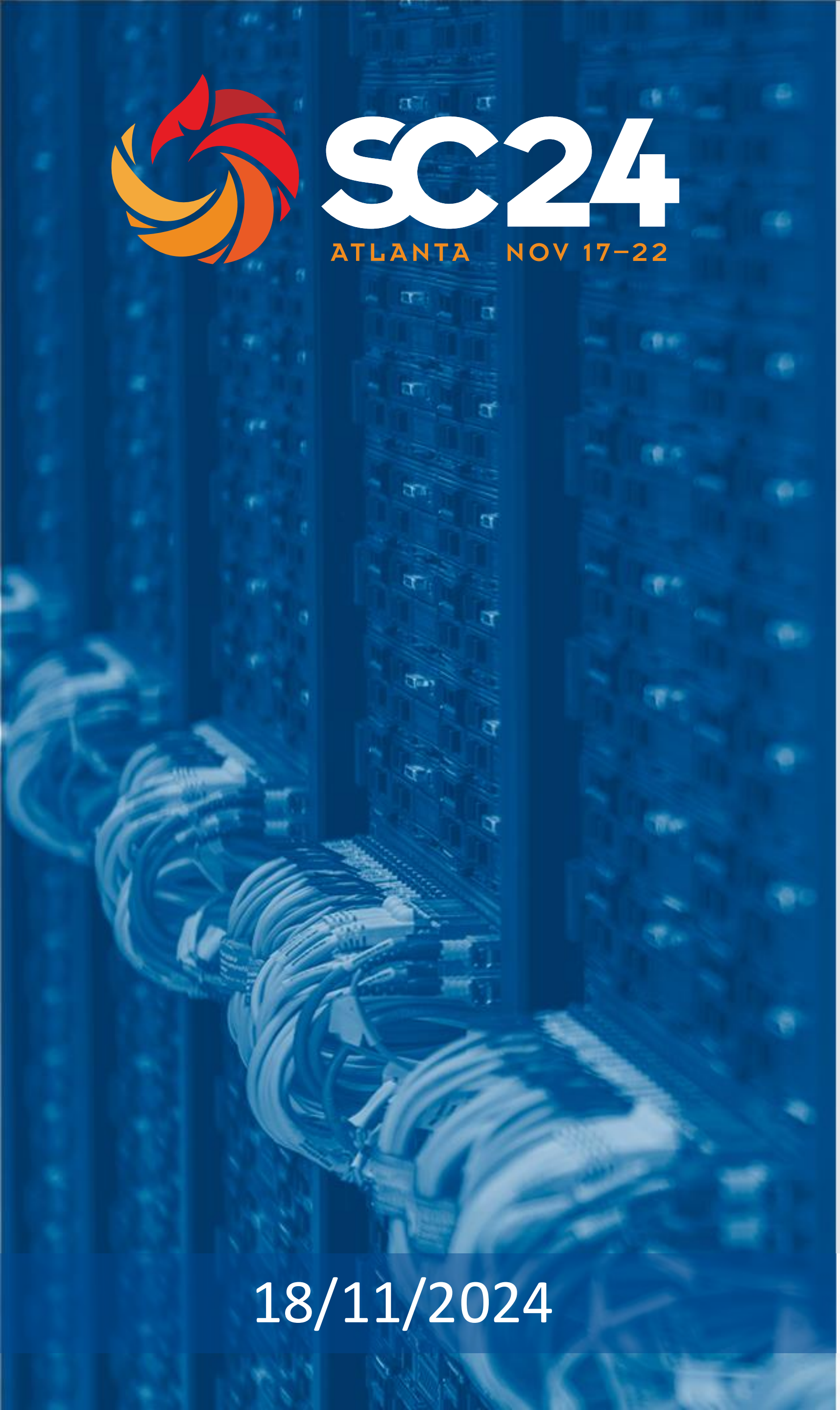


SC24 ATLANTA NOV 17–22

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Optimising workflow lifecycle management: development, HPC-ready containers deployment and reproducibility

Raül Sirvent, Rosa M Badia

18/11/2024

SC24 tutorial, Atlanta, 18 Novembre 2024

Tutorial website

https://github.com/bsc-wdc/Tutorial_SC24

Agenda

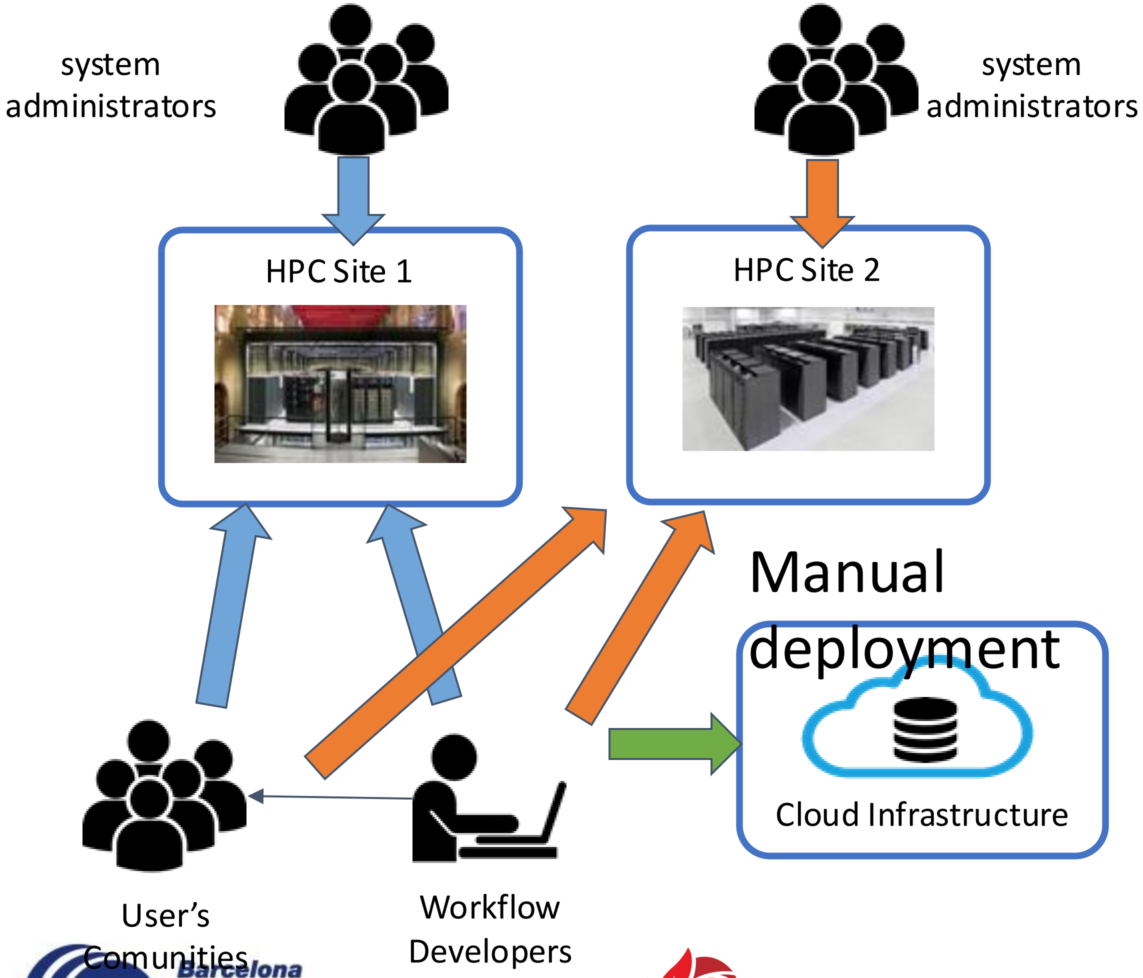
8:30 – 8:45	Overview of tutorial agenda	Rosa M Badia
8:45 – 9:10	Part 1.1: Hybrid HPC+AI+DA workflow development with PyCOMPSs - Context of the workflows at BSC - Overview of workflow development with PyCOMPSs - Extensions for the integration of HPC with AI and DA	Rosa M Badia
9:10 – 9:40	Part 1.2: Workflows' reproducibility through provenance - Motivation for workflow provenance - Design of the recording mechanism - Sharing experiments for reproducibility	Raül Sirvent
9:40 - 10:00	Part 1.3: HPC ready container images - Motivation for architecture specific containers - Overview of the Container Image Creation service - Example of HPC ready container generation	Rosa M Badia
10:00 - 10:30	Coffee break	

Agenda

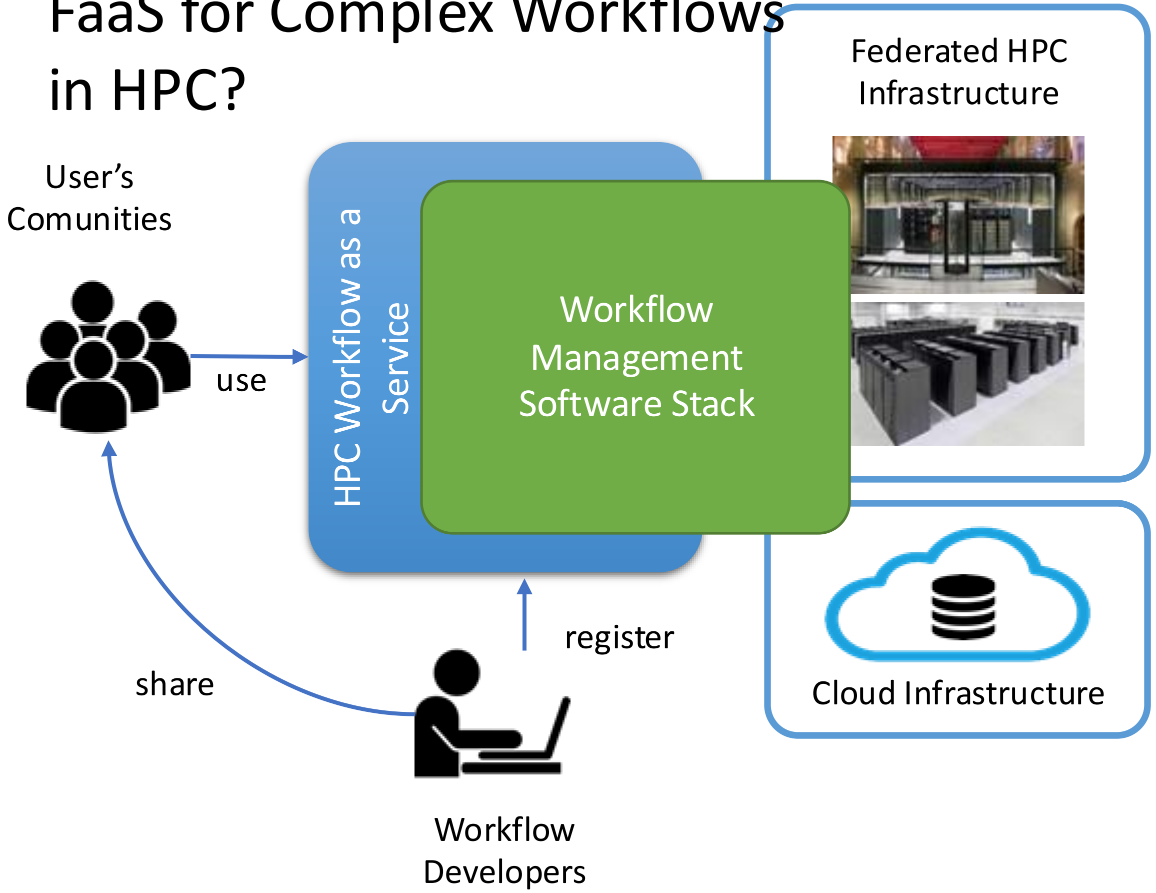
10:30 – 10:45	Hands-on preparation (credentials distribution, how to access, etc)	All presenters
10:45 – 11:15	Part 2.1: Hands-on session: Sample workflows with PyCOMPSs, execution with containers, task-graph generation, tracefile generation (optional)	Rosa M Badia
11:15 – 11:55	Part 2.2: Hands-on session: How to automatically record workflow provenance and use it to share experiments in WorkflowHub	Raül Sirvent
11:55 - 12:00	Tutorial conclusions	All presenters

Deployment in HPC Environments

Current approach

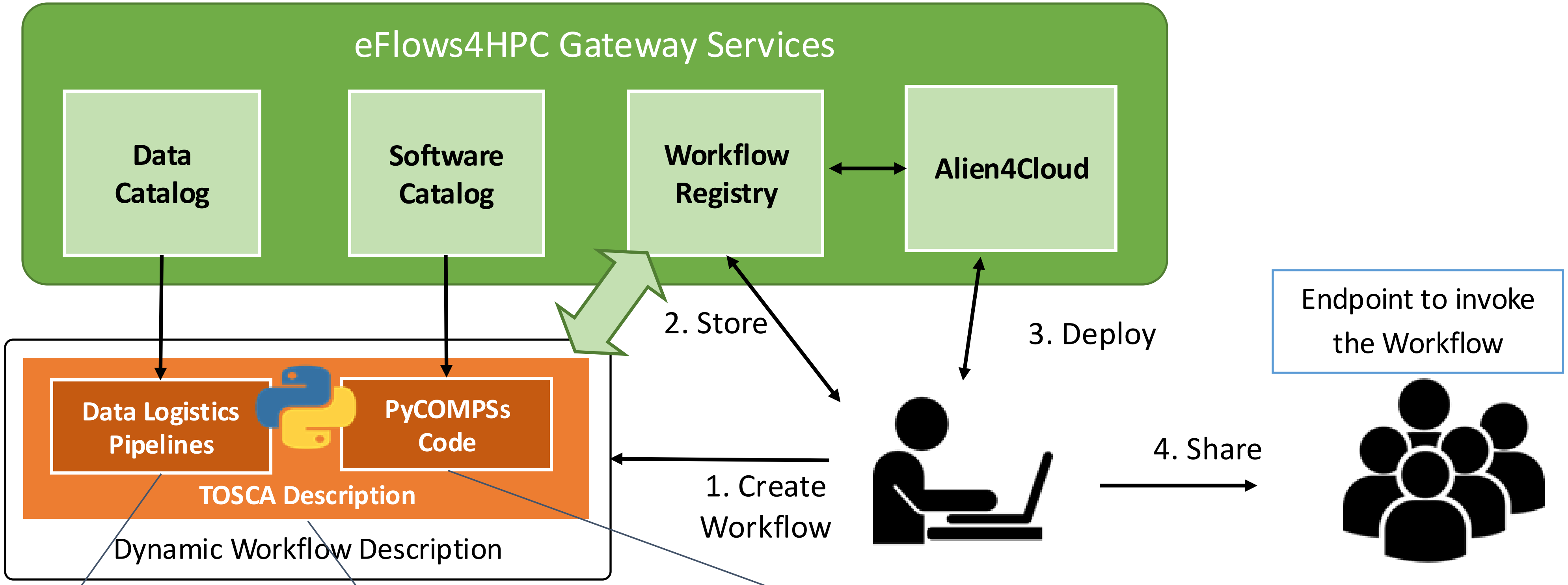


Can we apply something like FaaS for Complex Workflows in HPC?


Federated HPC Infrastructure
User's Comunities
use
HPC Workflow as a Service
Workflow Management Software Stack
register
share
Cloud Infrastructure
Workflow Developers

HPCWaaS: Workflow lifecycle overview

eFlows4HPC

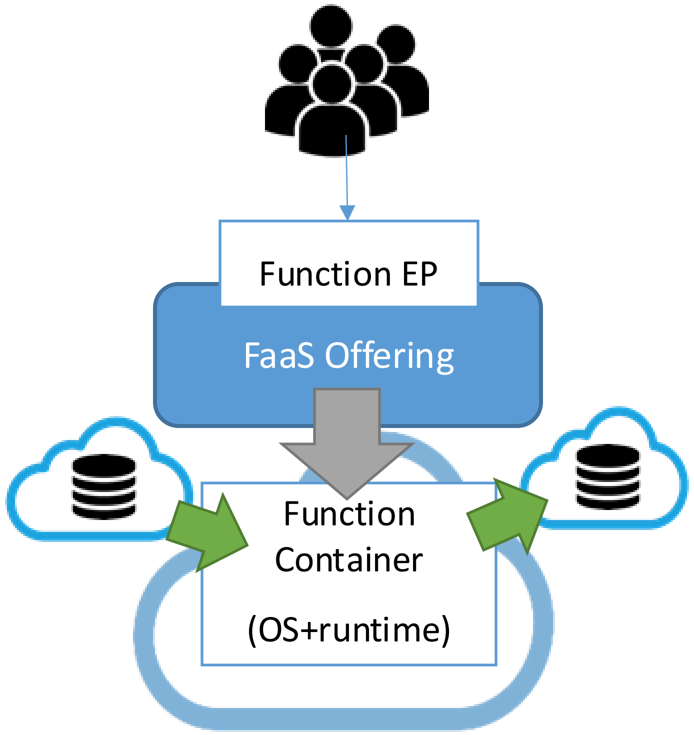


Description of data movements as Python functions.
Input/output datasets described at Data Catalog

Computational Workflow as a simple Python script.
Invocation of software described in the Software Catalog

Topology of the components involved in the workflow lifecycle and their relationship.

FaaS vs HPCWaaS

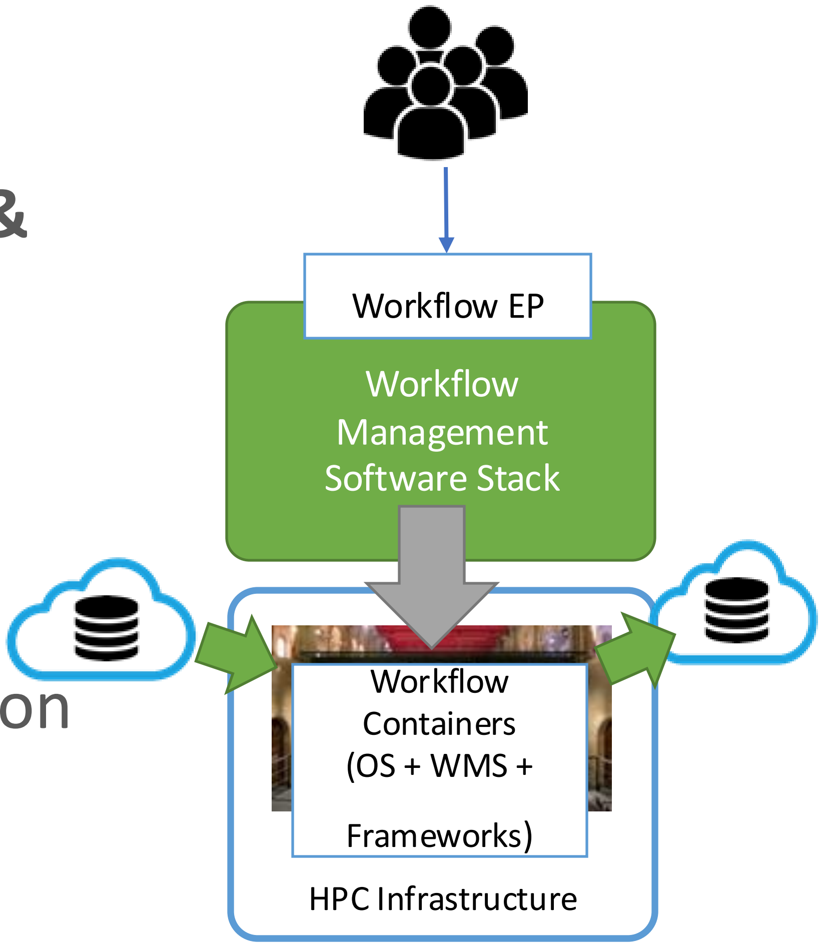


Similarities

- Easy to use for final user
- **Automate deployment & execution**
- Data integration
- **Containers**

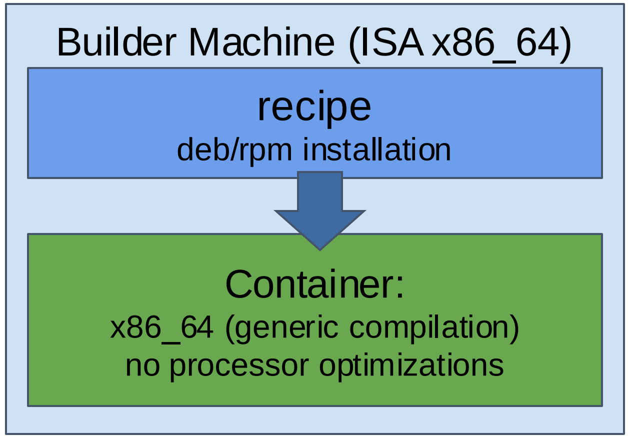
Differences

- Restrictions
- Deployment and Execution Complexity
- **Performance**


Workflow EP
Workflow Management Software Stack
Workflow Containers (OS + WMS + Frameworks)
HPC Infrastructure

Containers and HPC

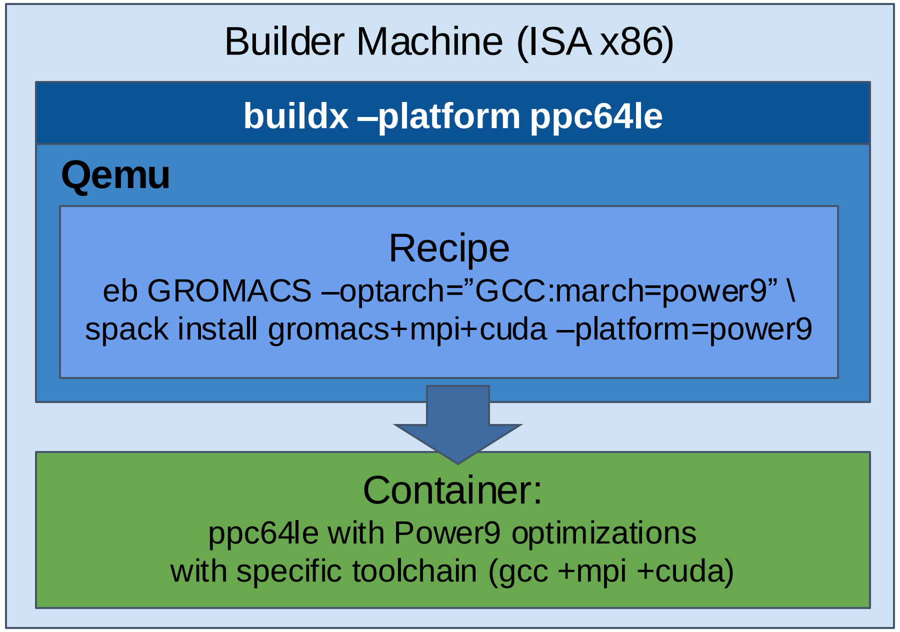
Standard container image creation


Builder Machine (ISA x86_64)
recipe
deb/rpm installation
Container:
x86_64 (generic compilation)
no processor optimizations

- **Simplicity for deployment**
  - Just pull or download the image
- **Trade-Off performance/portability**
  - Architecture Optimizations
- **Accessing Hardware from Containers**
  - MPI Fabric /GPUs
- **Host-Container Version Compatibility**

HPC Ready Containers

eFlows4HPC approach



Builder Machine (ISA x86)

buildx –platform ppc64le

Qemu

Recipe

eb GROMACS –optarch="GCC:march=power9" \
spack install gromacs+mpi+cuda –platform=power9

Container:

ppc64le with Power9 optimizations
with specific toolchain (gcc +mpi +cuda)

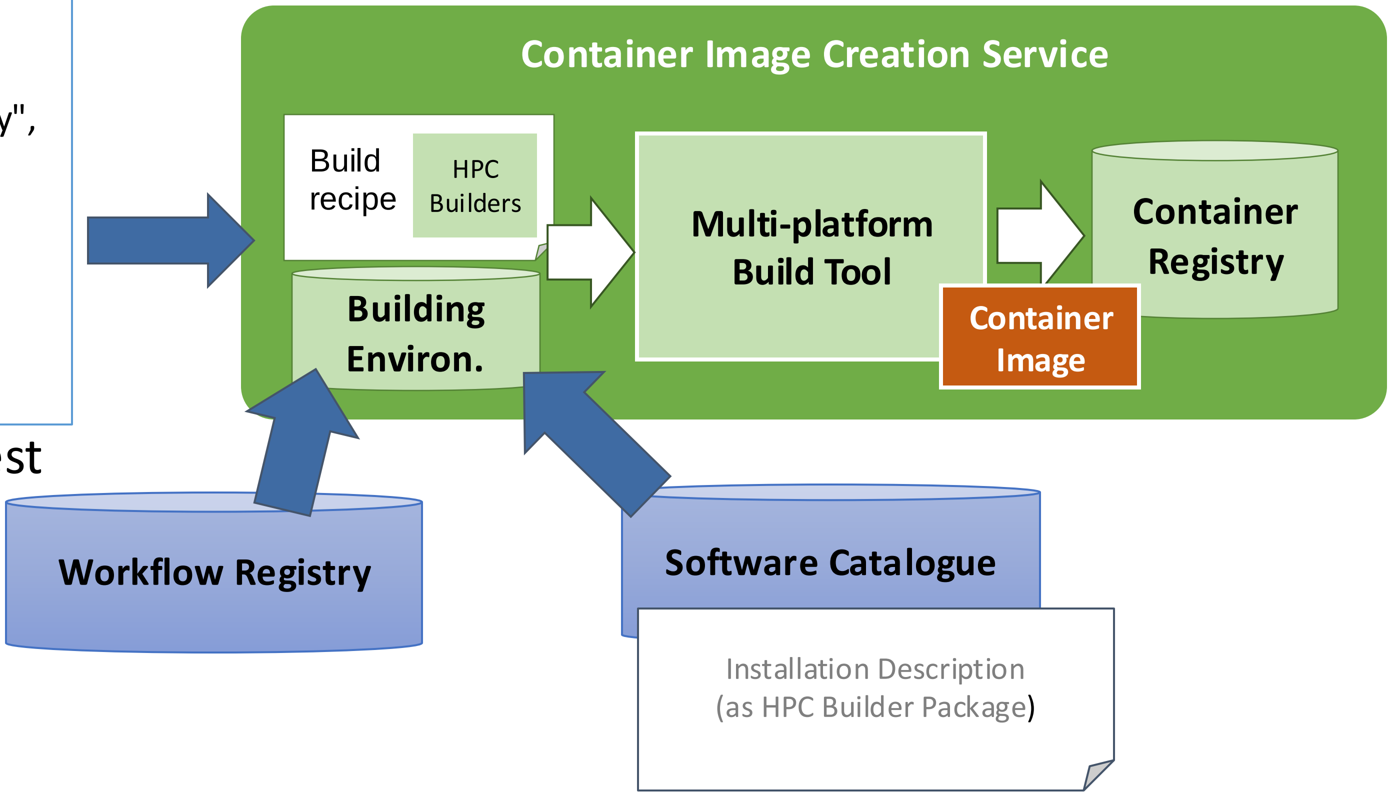
- **Methodology to allow the creation containers for specific HPC system**
  - Leverage HPC and Multi-platform container builders
- **It is tight to do by hand but let's automate!**

HPC Ready Containers

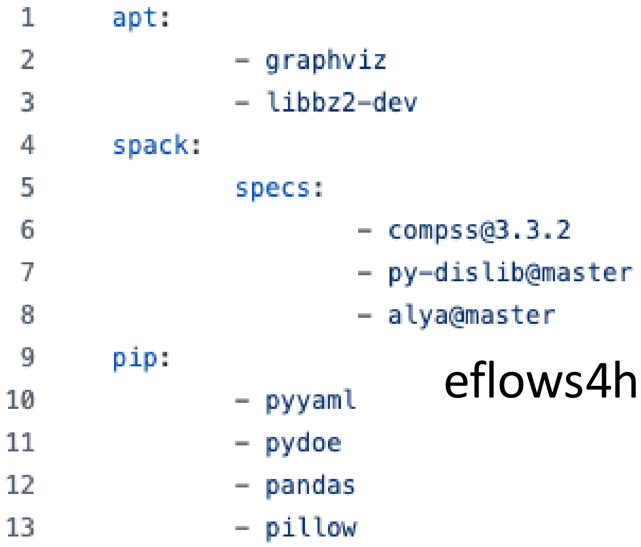
Workflow step + target system

```
{
"machine": {
                "platform": "linux/amd64",
                "architecture": "skylake",
                "container_engine": "singularity",
                "mpi":"openmpi@4.1.1"
},
"workflow":”tutorial",
"step_id" : " HPC_AI_training",
"workflow_yaml" : "eflows4hpc.yaml",
}
```

Service request



package.py



```
apt:
        - graphviz
        - libbz2-dev
spack:
        specs:
                - compss@3.3.2
                - py-dislib@master
                - alya@master
pip:
        - pyyaml
        - pydoe
        - pandas
        - pillow
```

eflows4hpc.yaml

CAELESTIS workflow

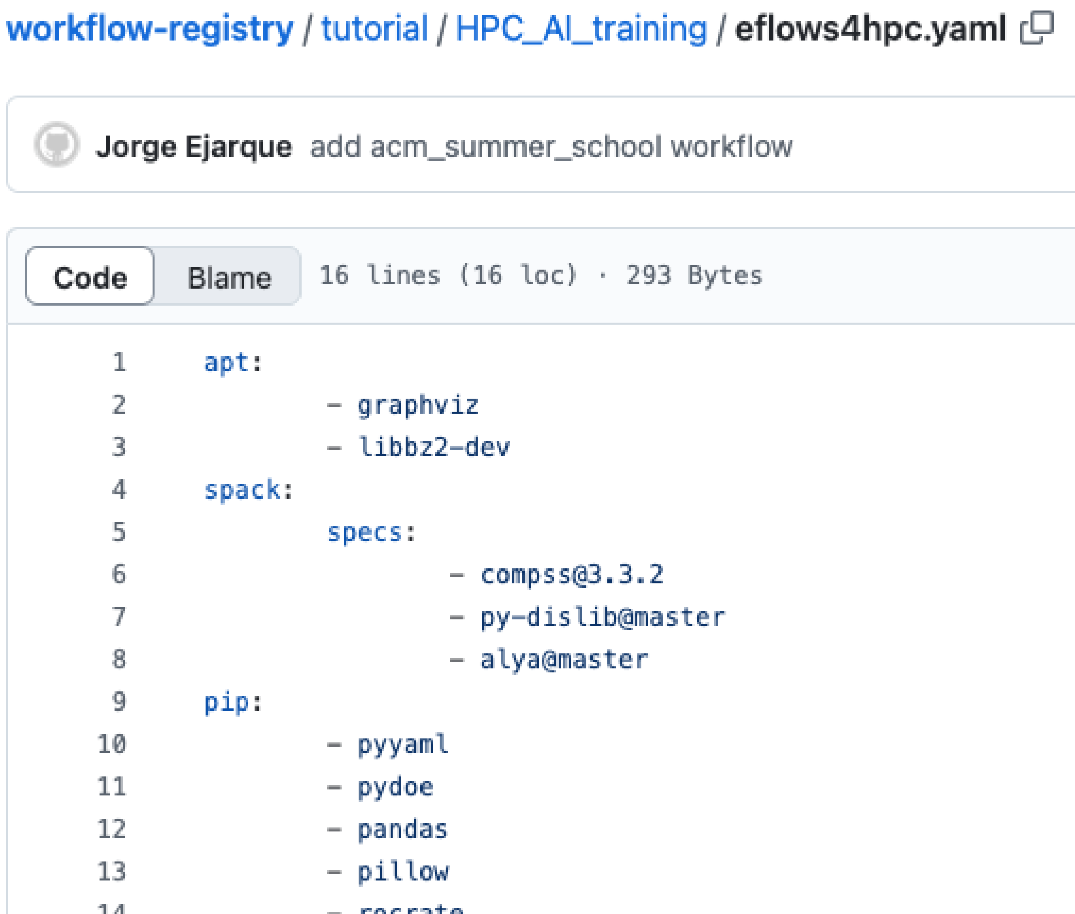
- Example of HPC ready container generation
- Sample request (json):

Target: MareNostrum4 architecture

```
{
"machine": {
            "platform": "linux/amd64",
            "architecture": "skylake",
            "container_engine": "singularity",
            "mpi":"openmpi@4.1.1"
},
"workflow":”tutorial",
"step_id" : "HPC_AI_training",
"workflow_yaml" : "eflows4hpc.yaml",
"force": "True",
"push" : "False"
}
```

Location in workflow-registry

Yaml file describing modules involved in the workflow



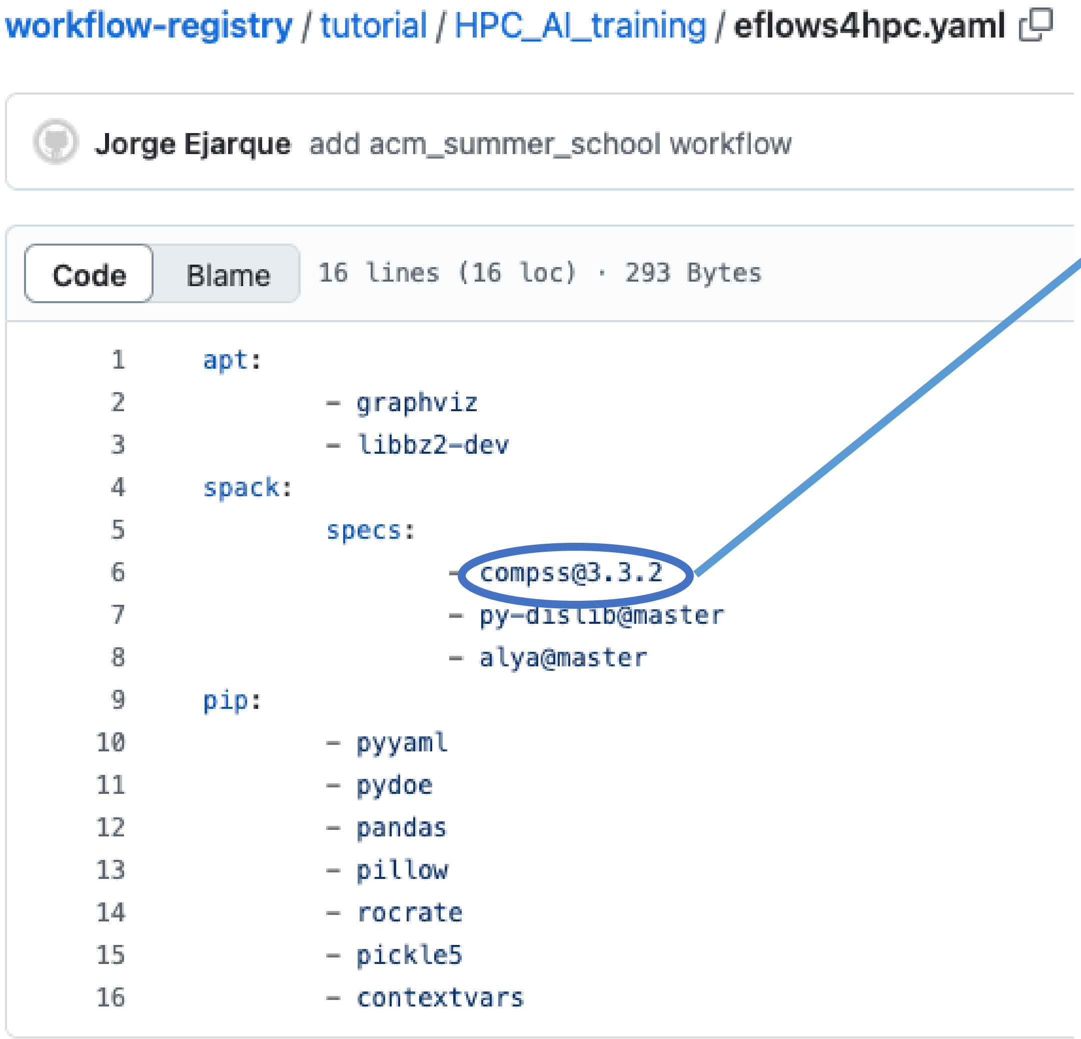
workflow-registry / tutorial / HPC_AI_training / eflows4hpc.yaml

Jorge Ejarque add acm_summer_school workflow

16 lines (16 loc) · 293 Bytes

```
apt:
        - graphviz
        - libbz2-dev
spack:
        specs:
                - compss@3.3.2
                - py-dislib@master
                - alya@master
pip:
        - pyyaml
        - pydoe
        - pandas
        - pillow
        - rocrate
```

CAELESTIS workflow



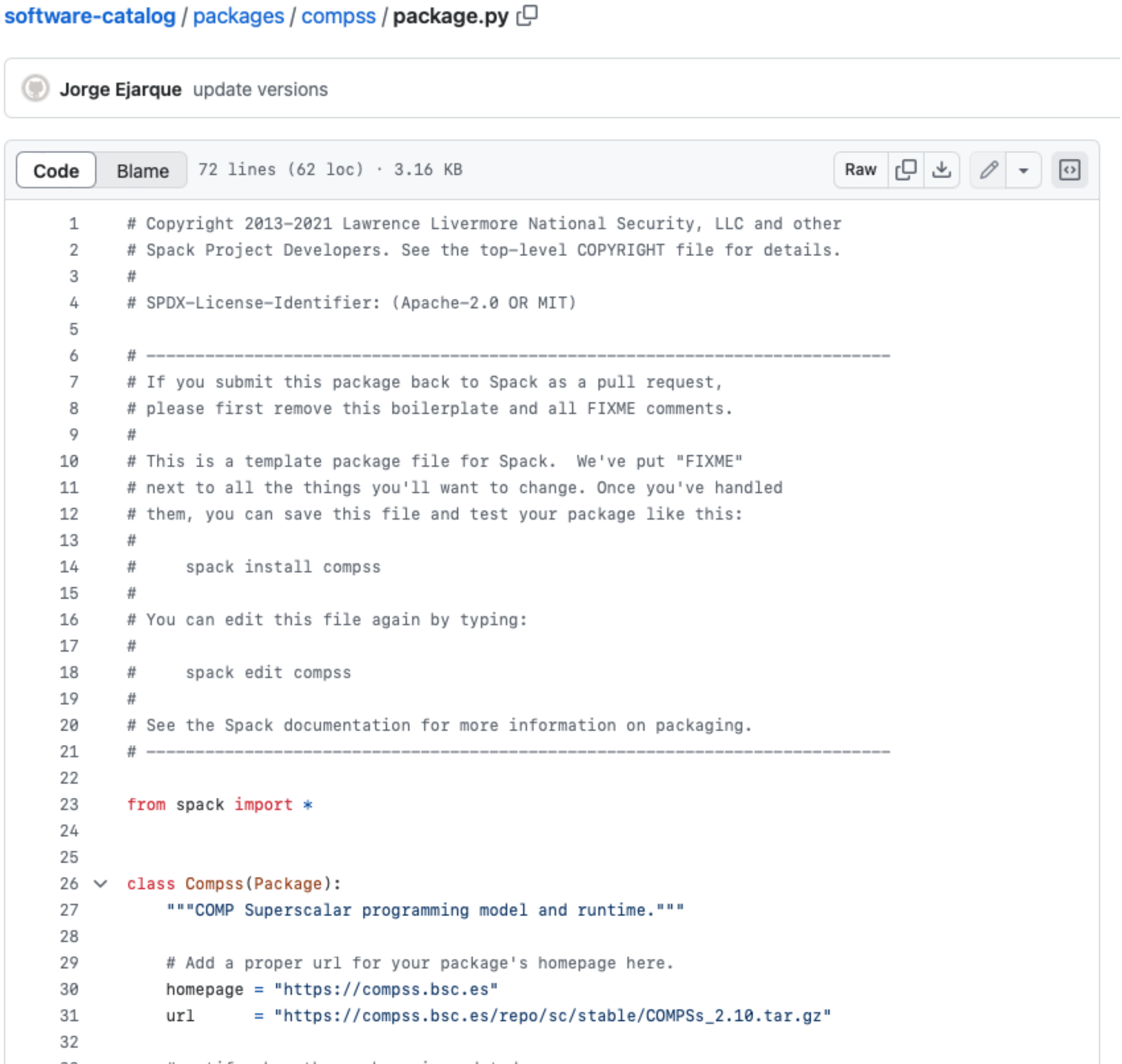
workflow-registry / tutorial / HPC_AI_training / eflows4hpc.yaml

Jorge Ejarque add acm_summer_school workflow

16 lines (16 loc) · 293 Bytes

```yaml
apt:
        - graphviz
        - libbz2-dev
spack:
        specs:
                - compss@3.3.2
                - py-dislib@master
                - alya@master
pip:
        - pyyaml
        - pydoe
        - pandas
        - pillow
        - rocrate
        - pickle5
        - contextvars
```

Specific spack packages



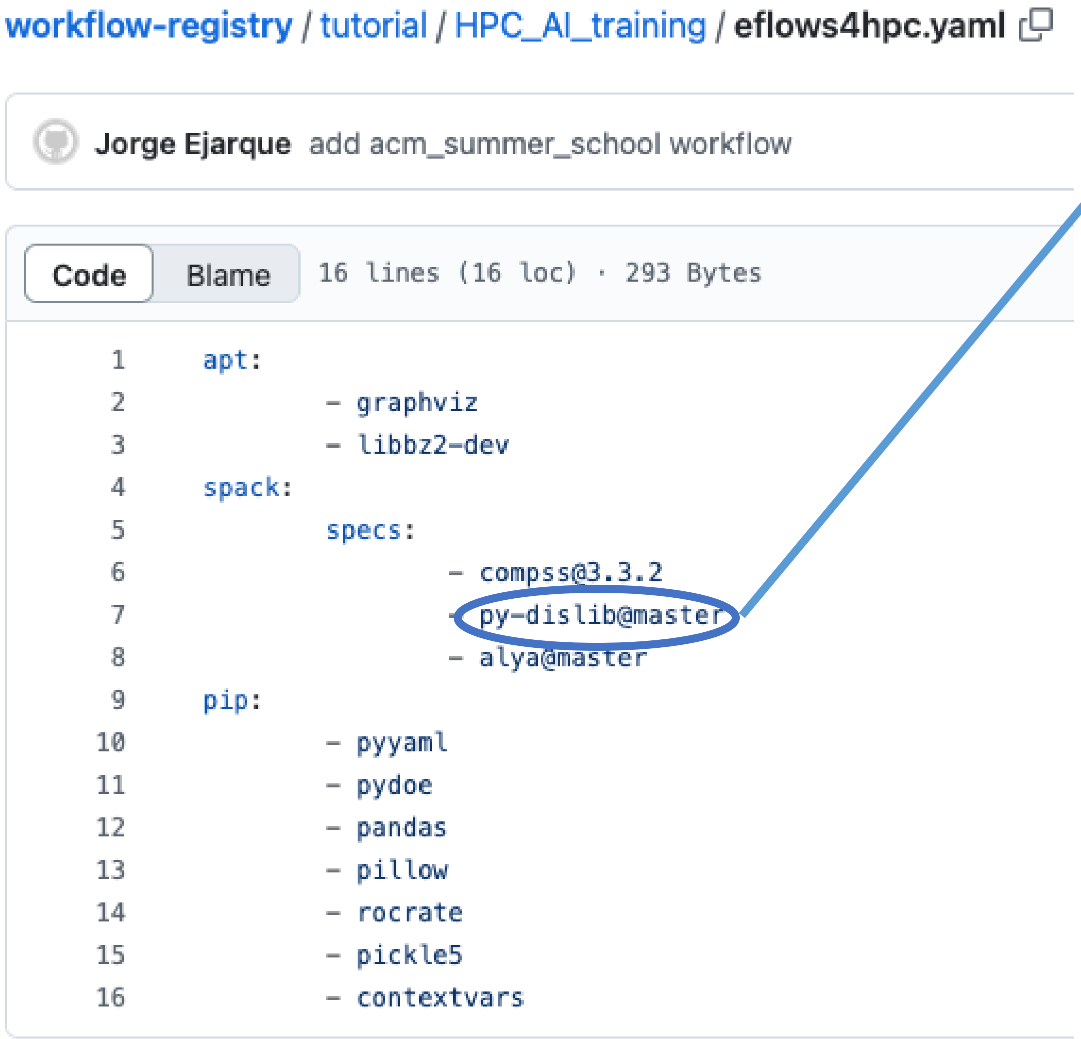
software-catalog / packages / compss / package.py

Jorge Ejarque update versions

72 lines (62 loc) · 3.16 KB

```python
# Copyright 2013-2021 Lawrence Livermore National Security, LLC and other
# Spack Project Developers. See the top-level COPYRIGHT file for details.
#
# SPDX-License-Identifier: (Apache-2.0 OR MIT)

# ----------------------------------------------------------------------------
# If you submit this package back to Spack as a pull request,
# please first remove this boilerplate and all FIXME comments.
#
# This is a template package file for Spack.  We've put "FIXME"
# next to all the things you'll want to change. Once you've handled
# them, you can save this file and test your package like this:
#
#     spack install compss
#
# You can edit this file again by typing:
#
#     spack edit compss
#
# See the Spack documentation for more information on packaging.
# ----------------------------------------------------------------------------

from spack import *


class Compss(Package):
    """COMP Superscalar programming model and runtime."""

    # Add a proper url for your package's homepage here.
    homepage = "https://compss.bsc.es"
    url      = "https://compss.bsc.es/repo/sc/stable/COMPSs_2.10.tar.gz"

```

CAELESTIS workflow



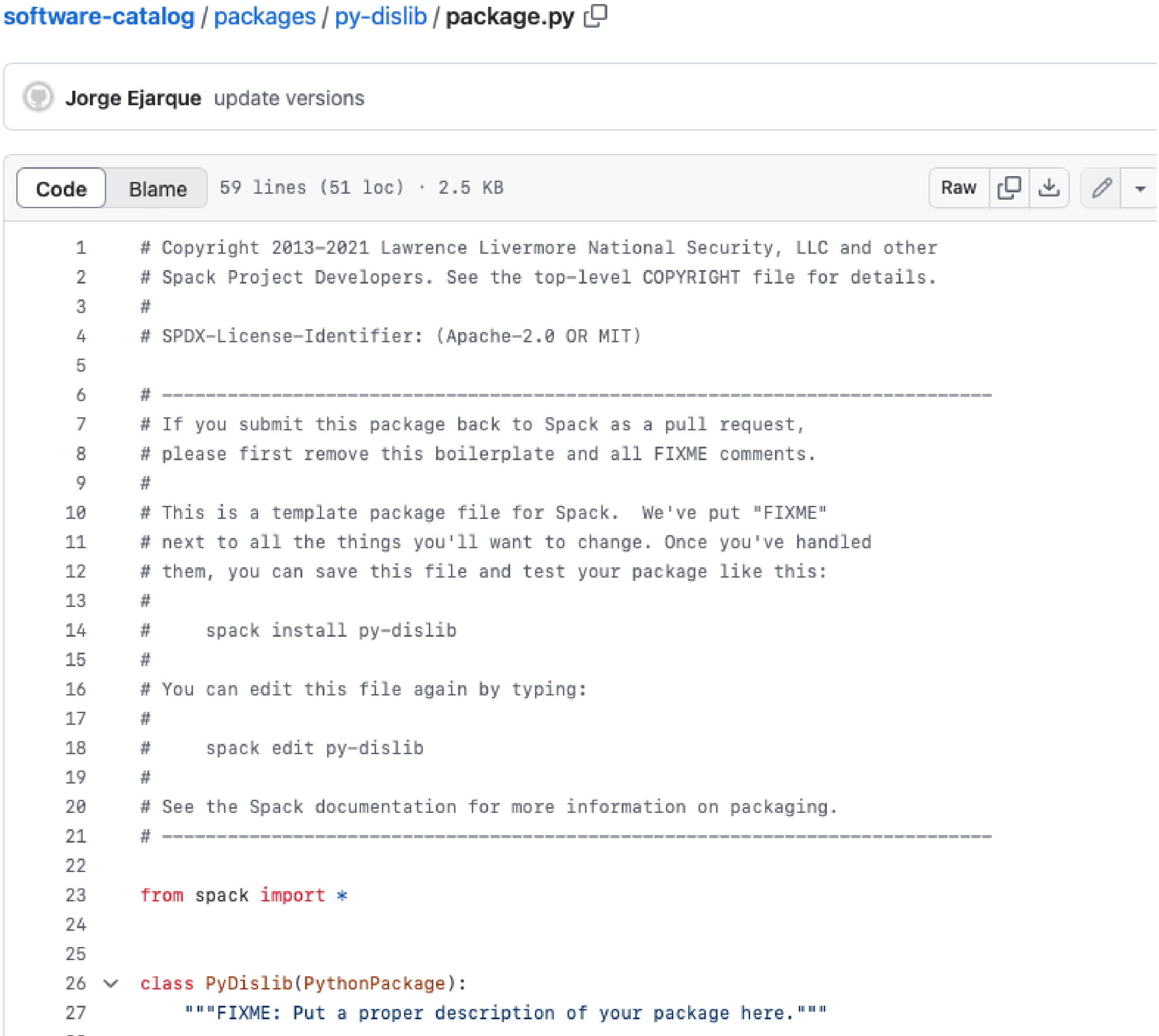
workflow-registry / tutorial / HPC_AI_training / eflows4hpc.yaml

Jorge Ejarque add acm_summer_school workflow

Code Blame 16 lines (16 loc) · 293 Bytes

```
apt:
        - graphviz
        - libbz2-dev
spack:
        specs:
                - compss@3.3.2
                - py-dislib@master
                - alya@master
pip:
        - pyyaml
        - pydoe
        - pandas
        - pillow
        - rocrate
        - pickle5
        - contextvars
```

Specific spack packages



software-catalog / packages / py-dislib / package.py

Jorge Ejarque update versions

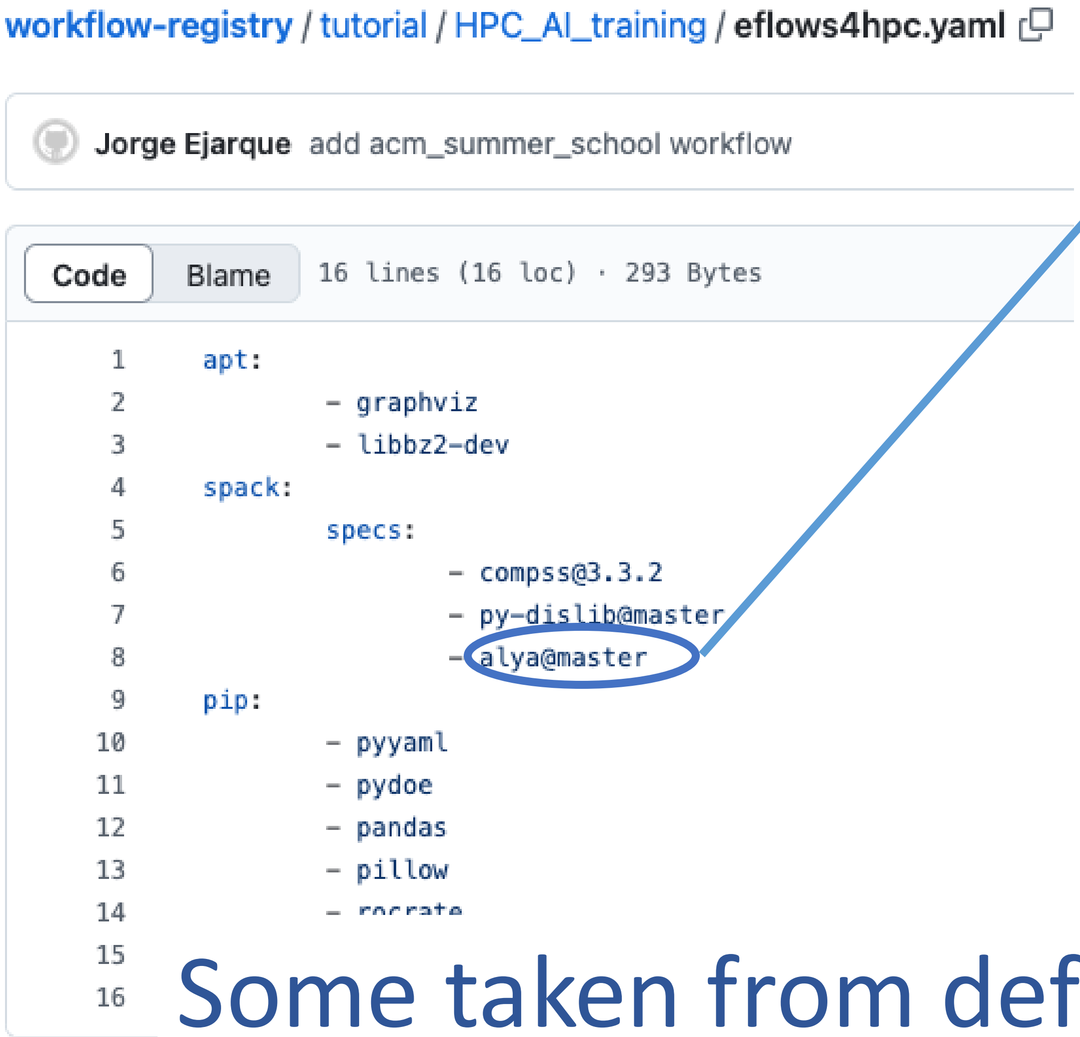
Code Blame 59 lines (51 loc) · 2.5 KB Raw

```
# Copyright 2013-2021 Lawrence Livermore National Security, LLC and other
# Spack Project Developers. See the top-level COPYRIGHT file for details.
#
# SPDX-License-Identifier: (Apache-2.0 OR MIT)

# ----------------------------------------------------------------------------
# If you submit this package back to Spack as a pull request,
# please first remove this boilerplate and all FIXME comments.
#
# This is a template package file for Spack.  We've put "FIXME"
# next to all the things you'll want to change. Once you've handled
# them, you can save this file and test your package like this:
#
#     spack install py-dislib
#
# You can edit this file again by typing:
#
#     spack edit py-dislib
#
# See the Spack documentation for more information on packaging.
# ----------------------------------------------------------------------------

from spack import *


class PyDislib(PythonPackage):
    """FIXME: Put a proper description of your package here."""
```

Supercomputing Center
Centro Nacional de Supercomputación

ATLANTA NOV 17–22

CAELESTIS workflow



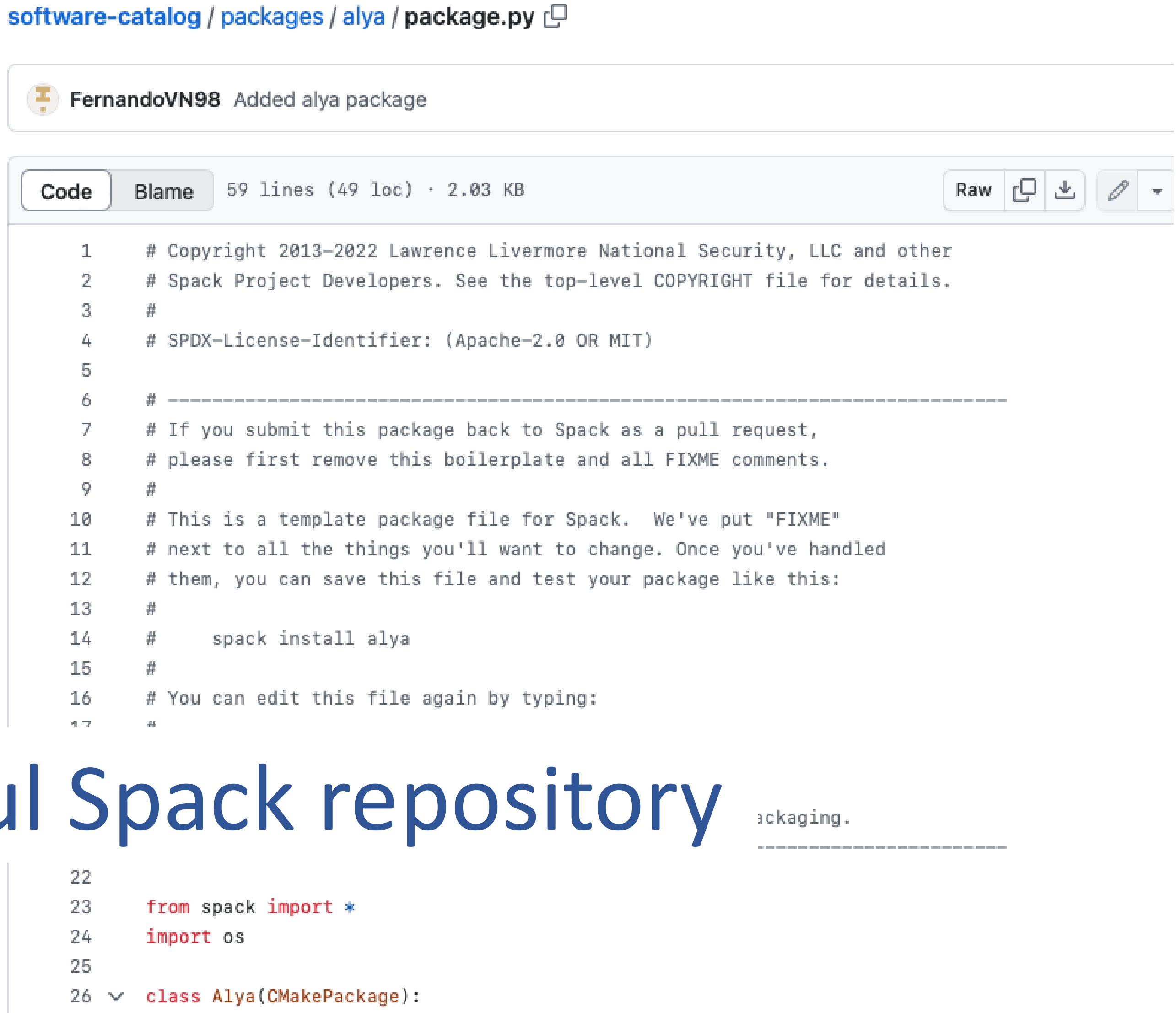
workflow-registry / tutorial / HPC_AI_training / eflows4hpc.yaml

Jorge Ejarque add acm_summer_school workflow

16 lines (16 loc) · 293 Bytes

```
apt:
        - graphviz
        - libbz2-dev
spack:
        specs:
                - compss@3.3.2
                - py-dislib@master
                - alya@master
pip:
        - pyyaml
        - pydoe
        - pandas
        - pillow
        - rocrate
```

Specific spack packages



software-catalog / packages / alya / package.py

FernandoVN98 Added alya package

59 lines (49 loc) · 2.03 KB

```
# Copyright 2013-2022 Lawrence Livermore National Security, LLC and other
# Spack Project Developers. See the top-level COPYRIGHT file for details.
#
# SPDX-License-Identifier: (Apache-2.0 OR MIT)

# ----------------------------------------------------------------------------
# If you submit this package back to Spack as a pull request,
# please first remove this boilerplate and all FIXME comments.
#
# This is a template package file for Spack.  We've put "FIXME"
# next to all the things you'll want to change. Once you've handled
# them, you can save this file and test your package like this:
#
#     spack install alya
#
# You can edit this file again by typing:
#
ackaging.
                        ----------------------

from spack import *
import os

class Alya(CMakePackage):
```

Some taken from defaul Spack repository

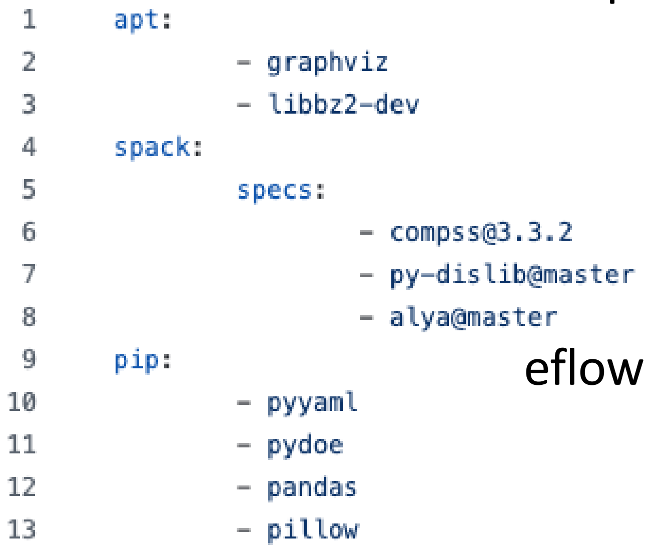
HPC Ready Containers

eFlows4HPC

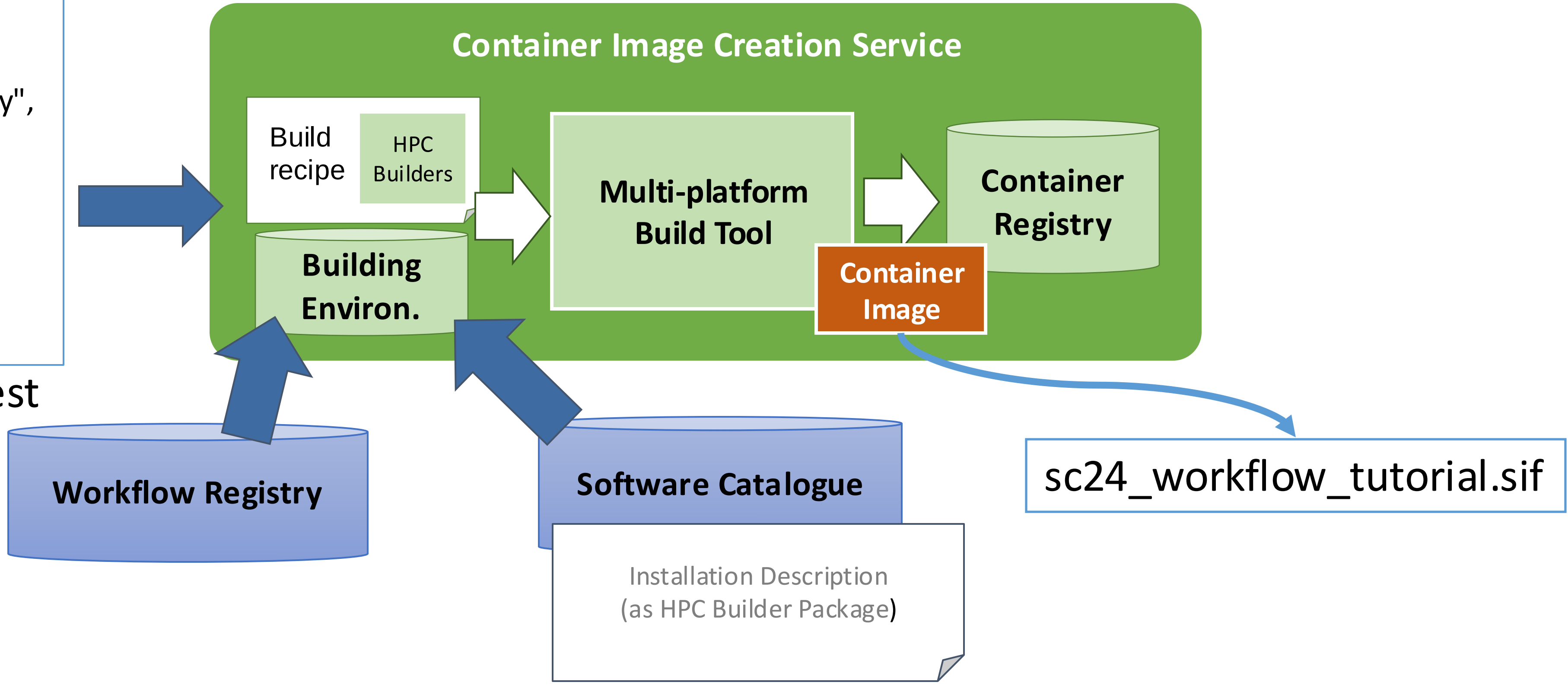
Workflow step + target system

```
{
"machine": {
                "platform": "linux/amd64",
                "architecture": "skylake",
                "container_engine": "singularity",
                "mpi":"openmpi@4.1.1"
},
"workflow":”tutorial",
"step_id" : ”HPC_AI_training",
"workflow_yaml" : "eflows4hpc.yaml",
}
```

Service request



```
apt:
          - graphviz
          - libbz2-dev
spack:
          specs:
                    - compss@3.3.2
                    - py-dislib@master
                    - alya@master
pip:
          - pyyaml
          - pydoe
          - pandas
          - pillow
```



Container Image Creation Service
Build recipe
HPC Builders
Building Environ.
Multi-platform Build Tool
Container Registry
Container Image
Workflow Registry
Software Catalogue
Installation Description (as HPC Builder Package)
sc24_workflow_tutorial.sif

eflows4hpc.yaml

package.py

Further Information

- Project page: http://www.bsc.es/compss
  - Documentation
  - Virtual Appliance for testing & sample applications
  - Tutorials
- Source Code

  https://github.com/bsc-wdc/compss
- Docker Image

  https://hub.docker.com/r/compss/compss
- Applications

  https://github.com/bsc-wdc/apps

  https://github.com/bsc-wdc/dislib
- Dislib
  - https://dislib.readthedocs.io/en/latest/

ACKs

DT-GEO

CEEC

Colmena

HP2C-DT

CyclOps

DISCOVER US

CAELESTIS

ICOS

Joint Laboratory for Extreme-Scale Computing
Inria
NCSA
Argonne
JÜLICH
RIKEN

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Generalitat de Catalunya
Departament de Recerca i Universitats

MINISTERIO DE CIENCIA E INNOVACIÓN

Financiado por la Unión Europea
NextGenerationEU

Plan de Recuperación, Transformación y Resiliencia

AGENCIA ESTATAL DE INVESTIGACIÓN

BSC Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Thanks!

COMPSs

rosa.m.badia@bsc.es