Unified access to multiple clouds and HPC clusters using PROMINENCE

Andrew Lahiff

UK Atomic Energy Authority

EGI Conference 2022

Dissemination level: Public

Disclosing Party: UKAEA

Recipient Party: EGI Conference 2022

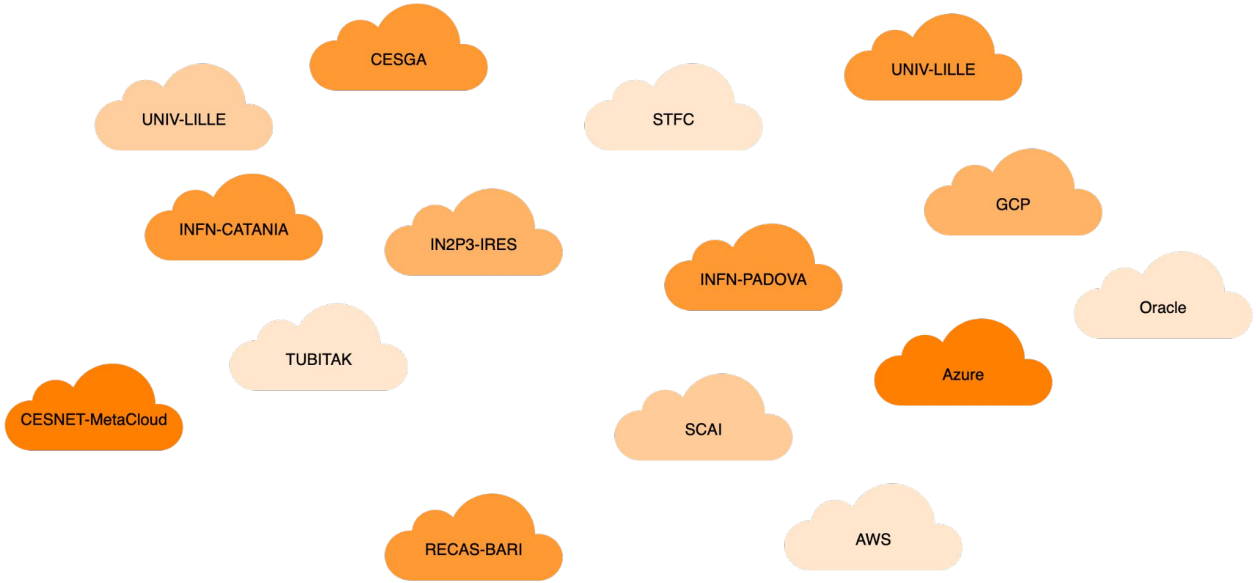
EGI-ACE receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 101017567.

Introduction to PROMINENCE

Introduction to PROMINENCE

Platform for running containerised HTC & HPC applications across multiple clouds simultaneously

- Originally designed for opportunistic usage of idle cloud resources

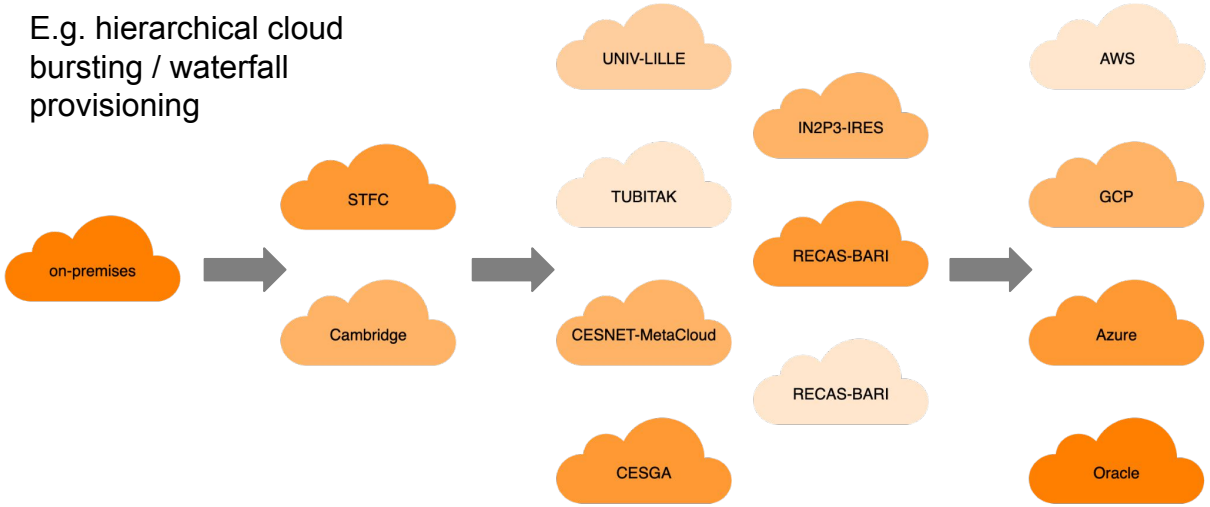


Introduction to PROMINENCE

From the user's perspective

- Appears like a normal batch system
- Jobs are directed automatically to the appropriate resources
- Don't need to worry about (or know about) clouds, clusters or infrastructure

E.g. hierarchical cloud bursting / waterfall provisioning



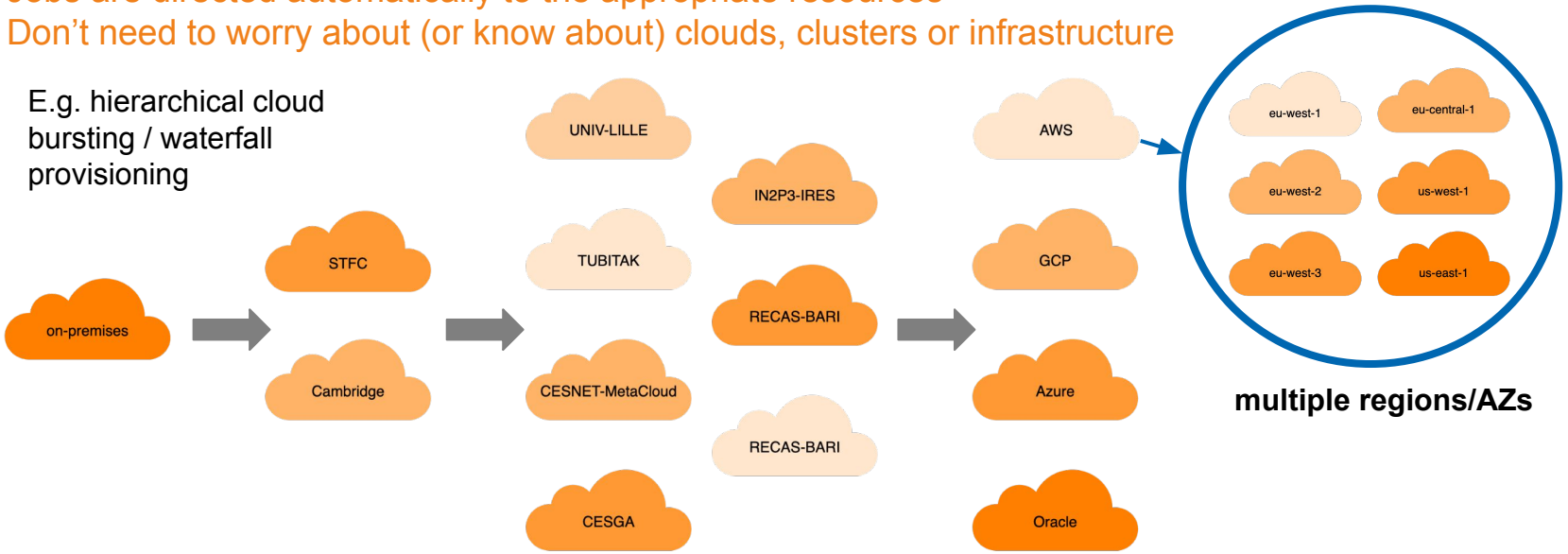
on-prem **national research clouds** **EGI** **public clouds**

Introduction to PROMINENCE

From the user's perspective

- Appears like a normal batch system
- Jobs are directed automatically to the appropriate resources
- Don't need to worry about (or know about) clouds, clusters or infrastructure

E.g. hierarchical cloud bursting / waterfall provisioning



Introduction to PROMINENCE

Interact via REST API and JSON
- Token-based authentication

EGI Check-in

INDIGO IAM

Simple CLI provides a batch-system like experience
- Submit jobs, list jobs, delete jobs, …
- Run anywhere - no need to ssh into a login node

```
pip install prominence-cli
```

Installing the CLI

```
prominence login
```

Get an access token

```
prominence stdout <job id>
```

View stdout/err in real time

Introduction to PROMINENCE

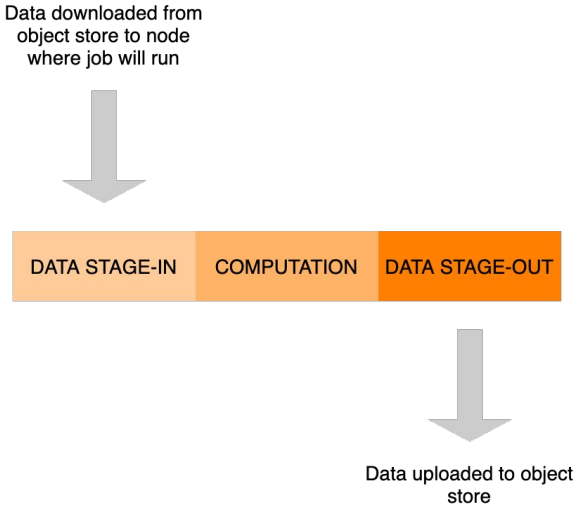
Supports individual jobs, large numbers of similar jobs, DAG workflows

In Fusion users are typically running:

- Individual jobs
- Multi-dimensional parameter sweeps
- More complex workflows, e.g. genetic algorithms (using PROMINENCE Python API)

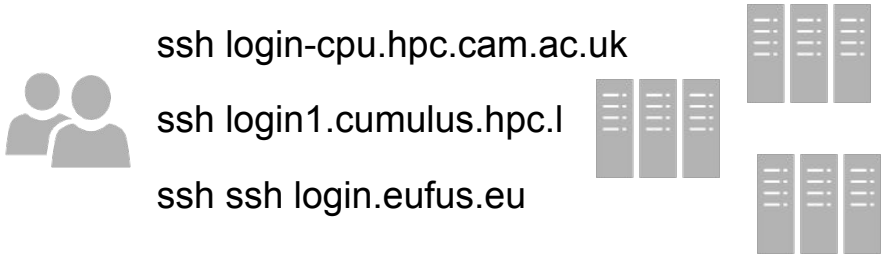
Data

- Object storage (Ceph with S3 API)
- OneData REST API



Introduction to PROMINENCE

Computing in the fusion energy research community mostly uses HPC clusters



HPC integration

Simplest idea

- Submit PROMINENCE worker nodes as jobs over ssh as needed
- udocker can be used to run jobs on any HPC system

However, need ssh keys

- Some of the main HPC facilities we use have policies prohibiting **any** single system submitting jobs on behalf of multiple users
- Integration of OIDC & ssh not yet common

Alternative: each user can be provided with a script, run as a cron on a login node, which:

- Queries the PROMINENCE API
- Submits workers nodes as jobs directly when needed

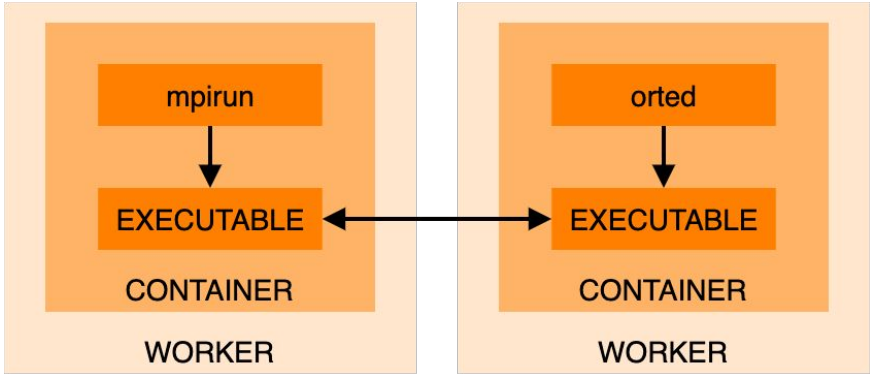
CLI can help users set this up:

```
prominence resource add --type slurm --name myhpc-1
```

HPC integration

Use MPI only in the container

- Avoids MPI version conflicts etc



OpenMPI example

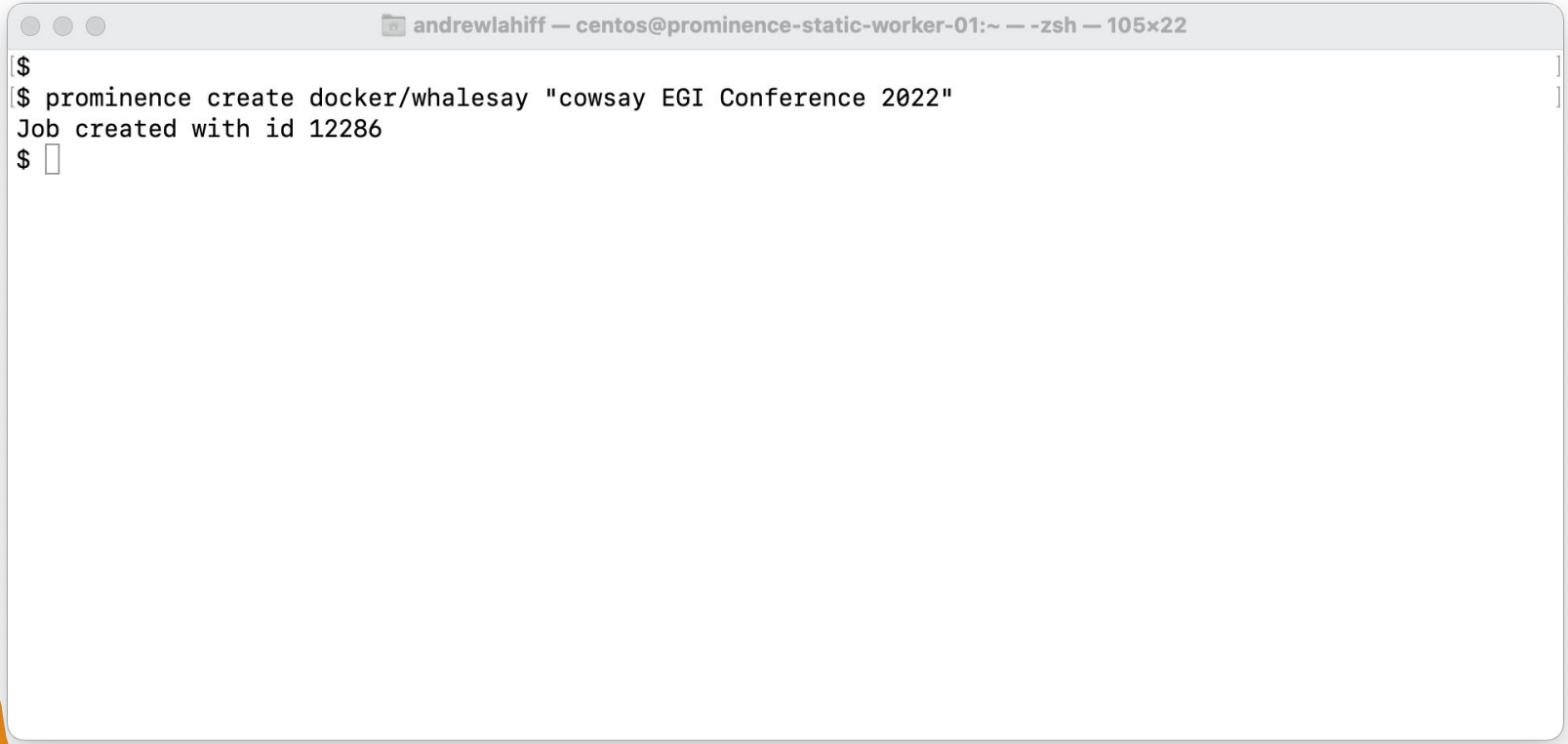
Container runtimes supported:

- udocker, Singularity

The demo…

Backup slides

Submitting a very basic job



Check job status

Get job standard output

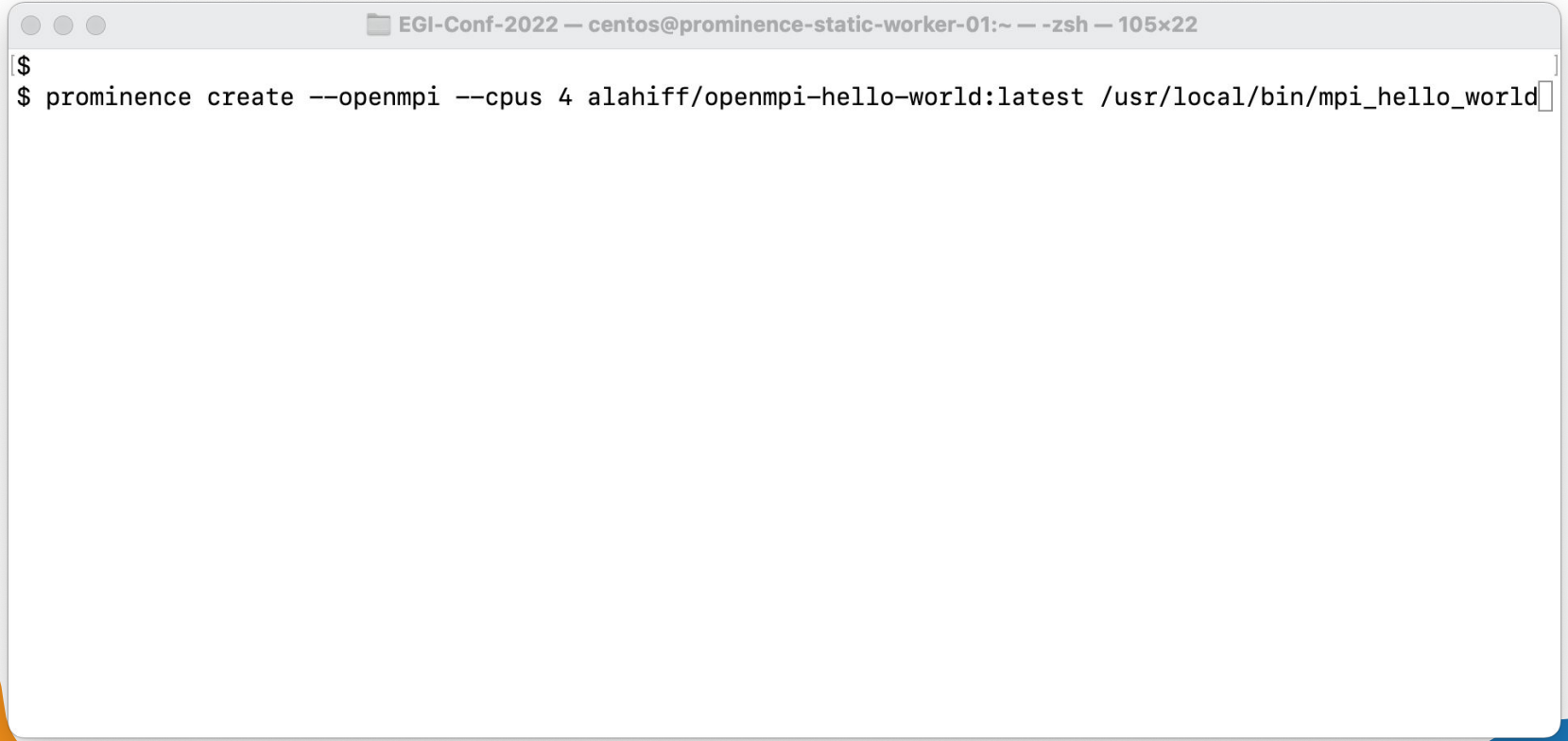
andrewlahiff — centos@prominence-static-worker-01:~ — -zsh — 105×22

```
$
$ prominence stdout 12286
 ______________________
< EGI Conference 2022 >
 ----------------------
    \
     \
      \
                    ##        .
              ## ## ##       ==
           ## ## ## ##      ===
       /""""""""""""""""___/ ===
  ~~~ {~~ ~~~~ ~~~ ~~~~ ~~ ~ /  ===- ~~~
       \______ o          __/
        \     \        __/
          \____\______/

$
```

Small input files

MPI jobs - static resource requirements



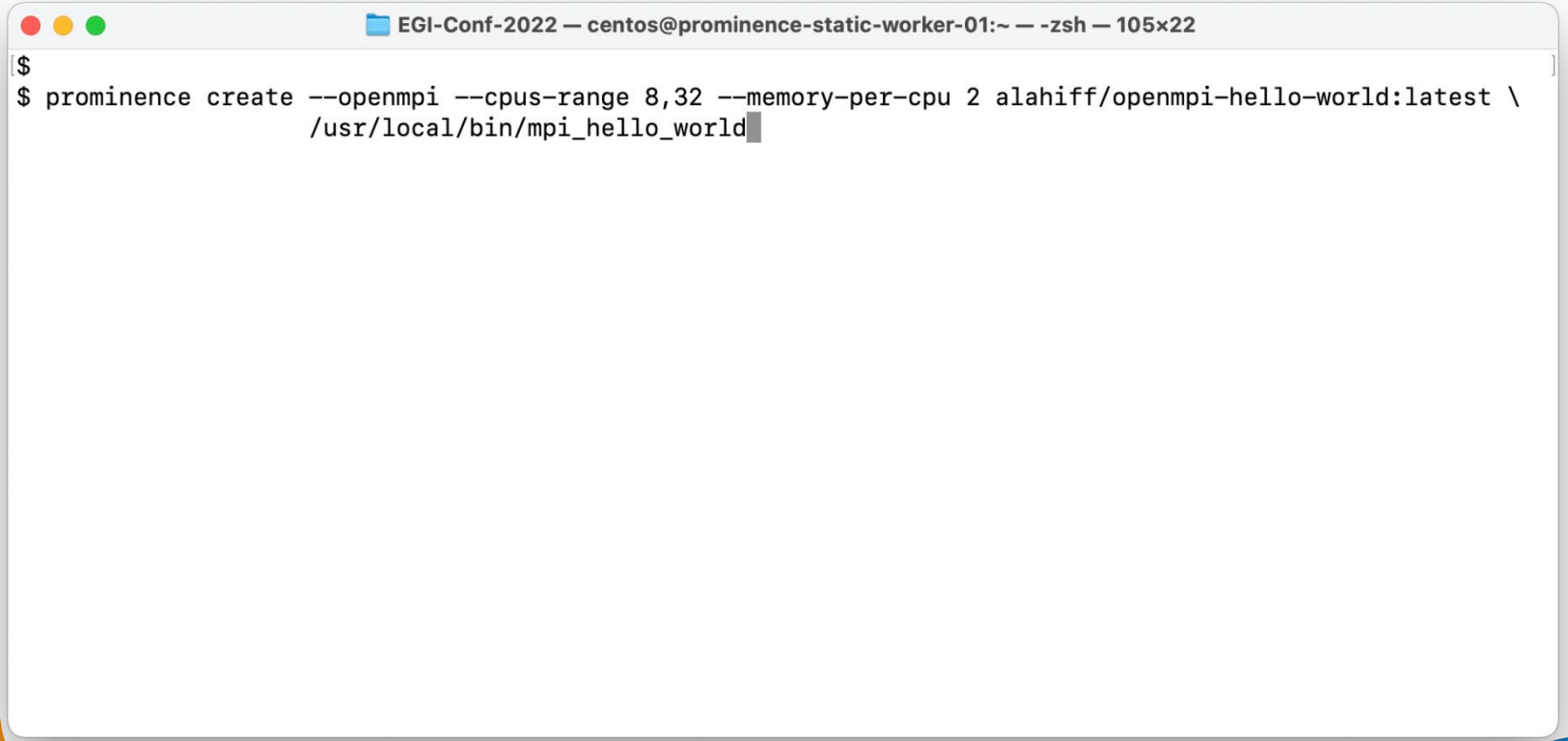
JSON descriptions of jobs

“prominence create --dry-run …”

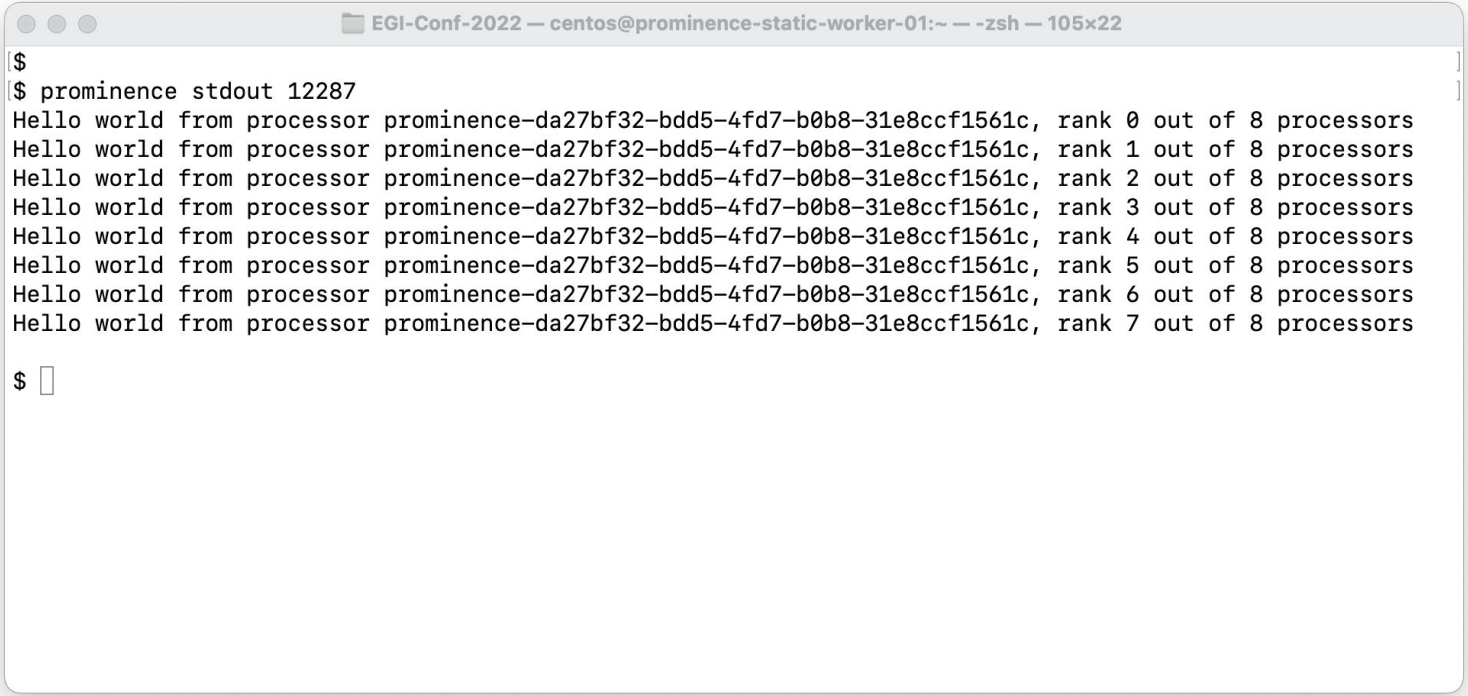
EGI-Conf-2022 — centos@prominence-static-worker-01:~ — -zsh — 105×22

```
$
$ prominence create --dry-run --openmpi --cpus 4 alahiff/openmpi-hello-world:latest \
          /usr/local/bin/mpi_hello_world
{
  "resources": {
    "nodes": 1,
    "disk": 10,
    "cpus": 4,
    "memory": 1
  },
  "name": "",
  "tasks": [
    {
      "image": "alahiff/openmpi-hello-world:latest",
      "runtime": "singularity",
      "type": "openmpi",
      "cmd": "/usr/local/bin/mpi_hello_world"
    }
  ]
}
$
```

MPI jobs - dynamic resource requirements (1)



MPI jobs - dynamic resource requirements (2)



Information about jobs (1)

Information about jobs (2)

EGI-Conf-2022 — centos@prominence-static-worker-01:~ — -zsh — 105×22


```
    }
  ],
  "events": {
    "createTime": "2022-09-22 06:28:04",
    "startTime": "2022-09-22 06:28:12",
    "endTime": "2022-09-22 06:28:15"
  },
  "execution": {
    "site": "OpenStack-CAM",
    "provisionedResources": {
      "cpus": 8,
      "disk": 11,
      "memory": 15,
      "nodes": 1
    },
    "cpu": {
      "clock": "2194.840",
      "model": "Intel Xeon Processor (Cascadelake)",
      "vendor": "GenuineIntel"
    },
    "runtimeVersion": {
      "singularity": "3.8.7-1.el7"
```

Larger input files (accessing data)

EGI-Conf-2022 — centos@prominence-static-worker-01:~ — -zsh — 105×22

```
$
$ cat lammps-intelmpi-singlenode.yaml
name: lammps-example01-01-intelmpi-singlenode
resources:
  cpus: 8
  disk: 10
  memory: 8
  nodes: 1
artifacts:
  - url: lammps-example01-01.tgz
tasks:
  - cmd: /usr/local/bin/lmp_intel_cpu_intelmpi -pk intel 0 -sf intel -in sample.in
    image: lammps-intel-avx512-2019_latest.sif
    runtime: singularity
    type: intelmpi
    workdir: lammps-example01-01
$
```

Multi-node MPI jobs

EGI-Conf-2022 — centos@prominence-static-worker-01:~ — -zsh — 105×22

```
$
$ cat lammps-intelmpi-multinode.yaml
name: lammps-example01-01-intelmpi-multinode
resources:
  cpus: 8
  disk: 10
  memory: 8
  nodes: 2
artifacts:
  - url: lammps-example01-01.tgz
tasks:
  - cmd: /usr/local/bin/lmp_intel_cpu_intelmpi -pk intel 0 -sf intel -in sample.in
    image: lammps-intel-avx512-2019_latest.sif
    runtime: singularity
    type: intelmpi
    workdir: lammps-example01-01
$
```

Monitor stdout/err in real-time

EGI-Conf-2022 — centos@prominence-static-worker-01:~ — -zsh — 105×22

```
$
$ prominence stdout 12288
LAMMPS (12 Dec 2018)
OMP_NUM_THREADS environment is not set. Defaulting to 1 thread. (../comm.cpp:87)
  using 1 OpenMP thread(s) per MPI task
Lattice spacing in x,y,z = 2.8552 2.8552 2.8552
Created orthogonal box = (0 0 0) to (285.52 285.52 285.52)
  2 by 2 by 2 MPI processor grid
Created 2000000 atoms
  Time spent = 0.0595694 secs
Deleted 256 atoms, new total = 1999744
WARNING: Using 'neigh_modify every 1 delay 0 check yes' setting during minimization (../min.cpp:168)
------------------------------------------------------------
Using Intel Package without Coprocessor.
Precision: mixed
------------------------------------------------------------
Neighbor list info ...
  update every 1 steps, delay 0 steps, check yes
  max neighbors/atom: 3000, page size: 100000
  master list distance cutoff = 8
  ghost atom cutoff = 8
  binsize = 4, bins = 72 72 72
```

Placement policies - preferences

EGI-Conf-2022 — centos@prominence-static-worker-01:~ — -zsh — 112×25

```
$
$ cat  lammps-intelmpi-multinode-policies.yaml
name: lammps-example01-01-intelmpi-multinode-policies
resources:
  cpus: 8
  disk: 10
  memory: 8
  nodes: 4
artifacts:
  - url: lammps-example01-01.tgz
tasks:
  - cmd: /usr/local/bin/lmp_intel_cpu_intelmpi -pk intel 0 -sf intel -in sample.in
    image: lammps-intel-avx512-2019_latest.sif
    runtime: singularity
    type: intelmpi
    workdir: lammps-example01-01
policies:
  placement:
    preferences:
      sites:
        - OpenStack-CAM
        - OpenStack-STFC
        - OpenStack-TUBITAK
$
```

Placement policies - requirements

EGI-Conf-2022 — centos@prominence-static-worker-01:~ — -zsh — 112×25

```
$
$ cat lammps-intelmpi-singlenode-hpc.yaml
name: lammps-example01-01-intelmpi-singlenode-hpc
resources:
  cpus: 8
  disk: 10
  memory: 8
  nodes: 1
artifacts:
  - url: lammps-example01-01.tgz
tasks:
  - cmd: /usr/local/bin/lmp_intel_cpu_intelmpi -pk intel 0 -sf intel -in sample.in
    image: lammps-intel-avx512-2019_latest.sif
    runtime: singularity
    type: intelmpi
    workdir: lammps-example01-01
policies:
  placement:
    requirements:
      sites:
        - SLURM-CSD3
$
```

HPC resource

Jobs running on HPC look no different to any other (1)

EGI-Conf-2022 — centos@prominence-static-worker-01:~ — -zsh — 112×25

```
$
$ prominence list
ID      NAME                                          CREATED              STATUS    ELAPSED      IMAGE
                      CMD
12289   lammps-example01-01-intelmpi-singlenode-hpc   2022-09-22 07:03:55   running   0+00:18:12   lammps-intel-
avx512-2019_latest.sif   /usr/local/bin/lmp_intel_cpu_intelmpi -pk intel 0 -sf intel -in sample.in
$
```

Jobs running on HPC look no different to any other (2)

EGI-Conf-2022 — centos@prominence-static-worker-01:~ — -zsh — 112×25

```
$
$ prominence stdout 12289
LAMMPS (12 Dec 2018)
OMP_NUM_THREADS environment is not set. Defaulting to 1 thread. (../comm.cpp:87)
  using 1 OpenMP thread(s) per MPI task
Lattice spacing in x,y,z = 2.8552 2.8552 2.8552
Created orthogonal box = (0 0 0) to (285.52 285.52 285.52)
  2 by 2 by 2 MPI processor grid
Created 2000000 atoms
  Time spent = 0.125903 secs
Deleted 256 atoms, new total = 1999744
WARNING: Using 'neigh_modify every 1 delay 0 check yes' setting during minimization (../min.cpp:168)
----------------------------------------------------------
Using Intel Package without Coprocessor.
Precision: mixed
----------------------------------------------------------
Neighbor list info ...
  update every 1 steps, delay 0 steps, check yes
  max neighbors/atom: 3000, page size: 100000
  master list distance cutoff = 8
  ghost atom cutoff = 8
  binsize = 4, bins = 72 72 72
  1 neighbor lists, perpetual/occasional/extra = 1 0 0
  (1) pair eam/fs/intel, perpetual
      attributes: half, newton on, intel
```

Extra slides (if needed)

Architecture