DevOps School for HPC

CARLA Conference
September 26, 2025

CARLA 2025

Globus Compute: Reliable Remote Computation at Scale

Lev Gorenstein (lev@globus.org)

THE UNIVERSITY OF CHICAGO

Globus is …

a non-profit service developed and operated by

THE UNIVERSITY OF CHICAGO

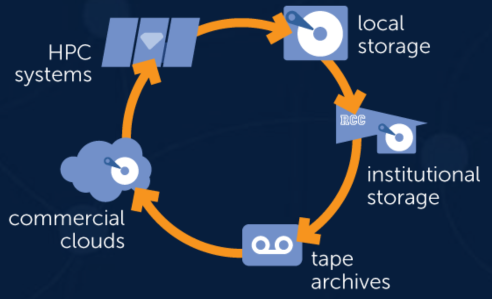
Globus: Platform for Data Driven Research

A
fast
secure transfer
reliable
B

Managed transfer & sync

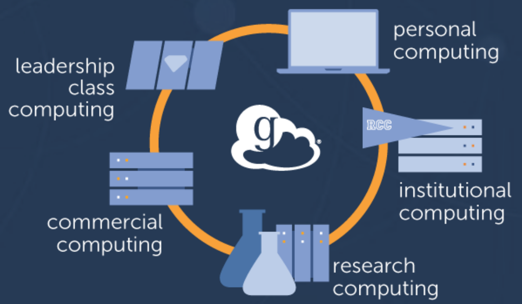
collaborators
colleagues

Collaborative data sharing

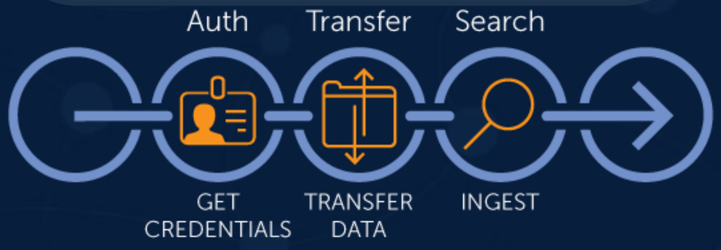


Unified data access

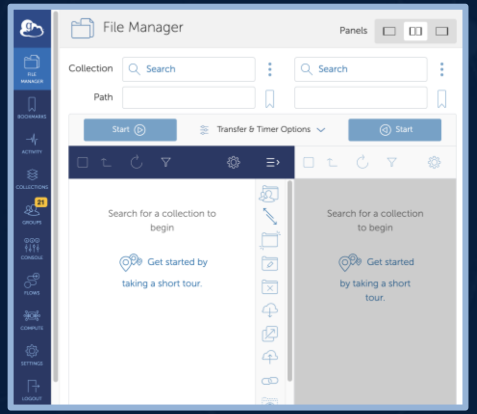
Publication & discovery



Managed remote execution



Reliable automation


File Manager

Panels

Collection: Search

Path

Search

Start

Transfer & Timer Options

Start

Search for a collection to begin

Get started by taking a short tour.

Search for a collection to begin

Get started by taking a short tour.

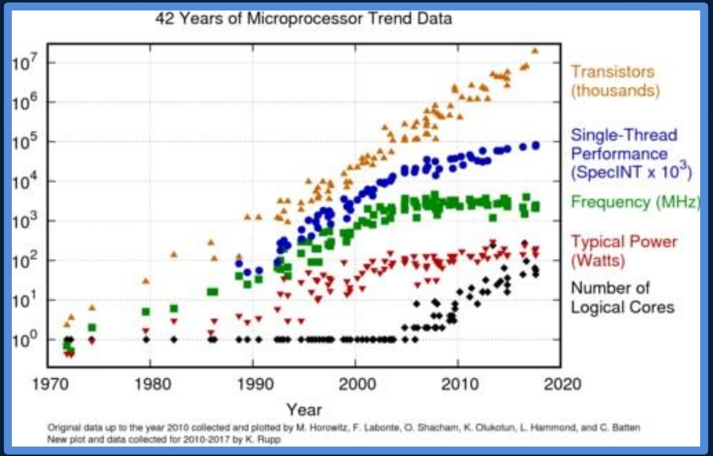
Software-as-a-Service

Platform-as-a-Service

The research computing ecosystem is rapidly evolving

Resources

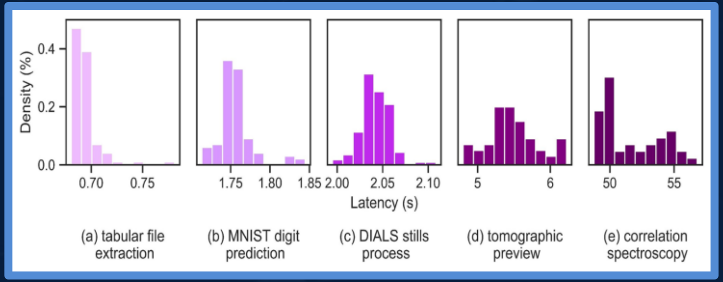
- Hardware specialization
- Specialization leads to distribution



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp

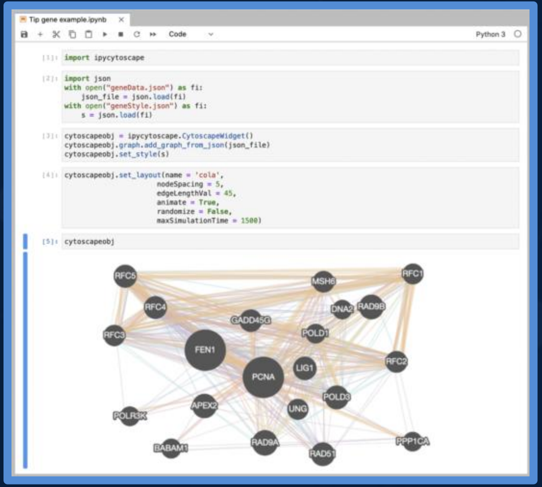
Workloads

- Interactive, real-time workloads
- Machine learning training and inference
- Components may best be executed in different places



Users

- Diverse backgrounds and expertise
- Different user interfaces (e.g., notebooks)



```
[1]: import ipycytoscape

[2]: import json
     with open("geneData.json") as fi:
         json_file = json.load(fi)
     with open("geneStyle.json") as fi:
         s = json.load(fi)

[3]: cytoscapeobj = ipycytoscape.CytoscapeWidget()
     cytoscapeobj.graph.add_graph_from_json(json_file)
     cytoscapeobj.set_style(s)

[4]: cytoscapeobj.set_layout(name = 'cola',
                             nodeSpacing = 5,
                             edgeLengthVal = 45,
                             animate = True,
                             randomize = False,
                             maxSimulationTime = 1500)

[5]: cytoscapeobj
```

How do we support researchers navigate this?

Move closer to researchers’ environments

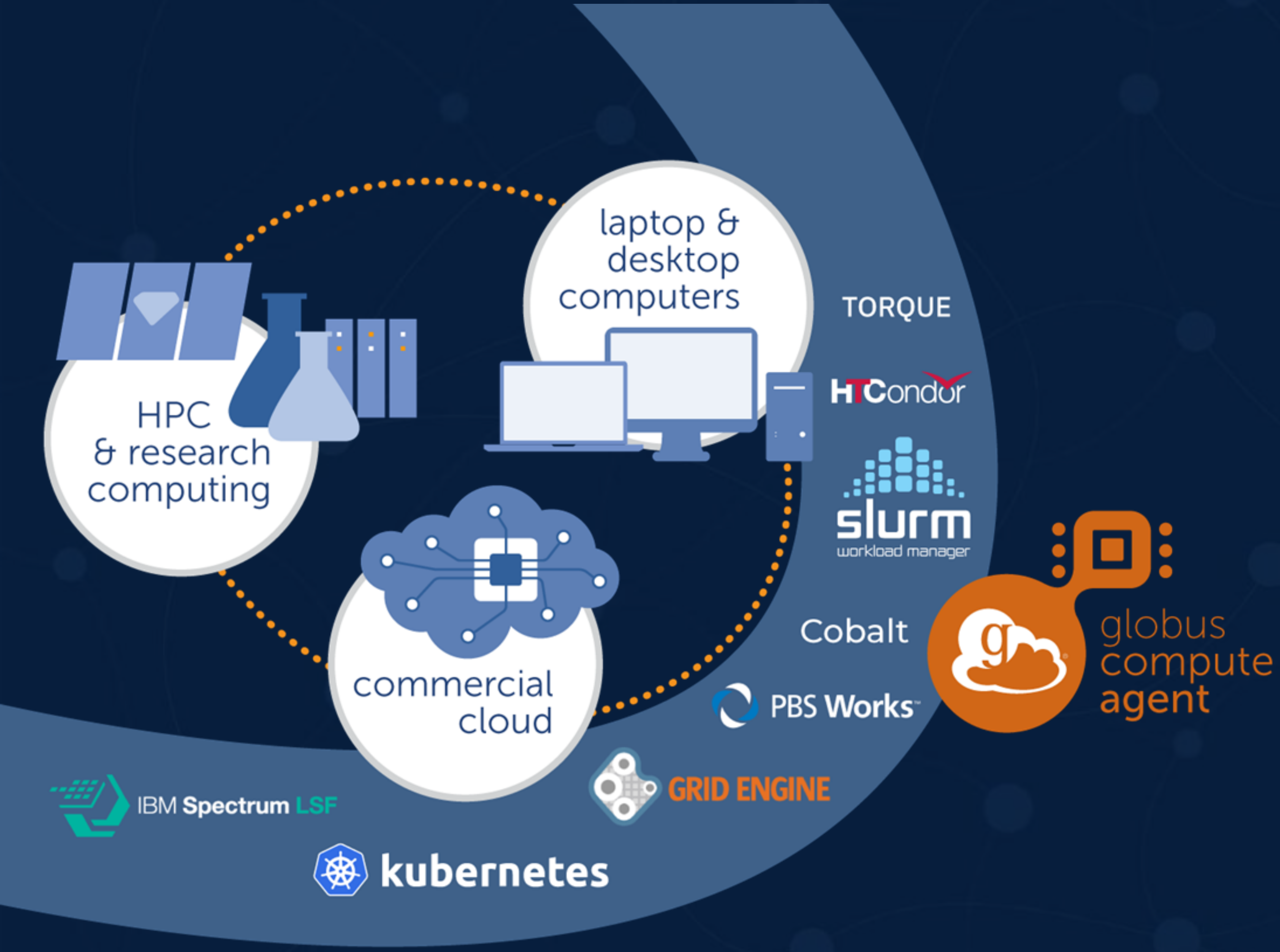
- Researchers primarily work in high level languages
- Functions are a natural unit of computation
- The Function-as-a-Service (FaaS) model allows researchers to work in a familiar language (e.g., Python) using familiar interfaces (e.g., Jupyter)

We asked: How can Globus deliver this?

Borrow page from data management playbook

- "Fire-and-forget" computation
- Uniform access interface
- Federated access control
- Programmatic interface to compute resource
- Administration interfaces for management and monitoring

Globus Compute?

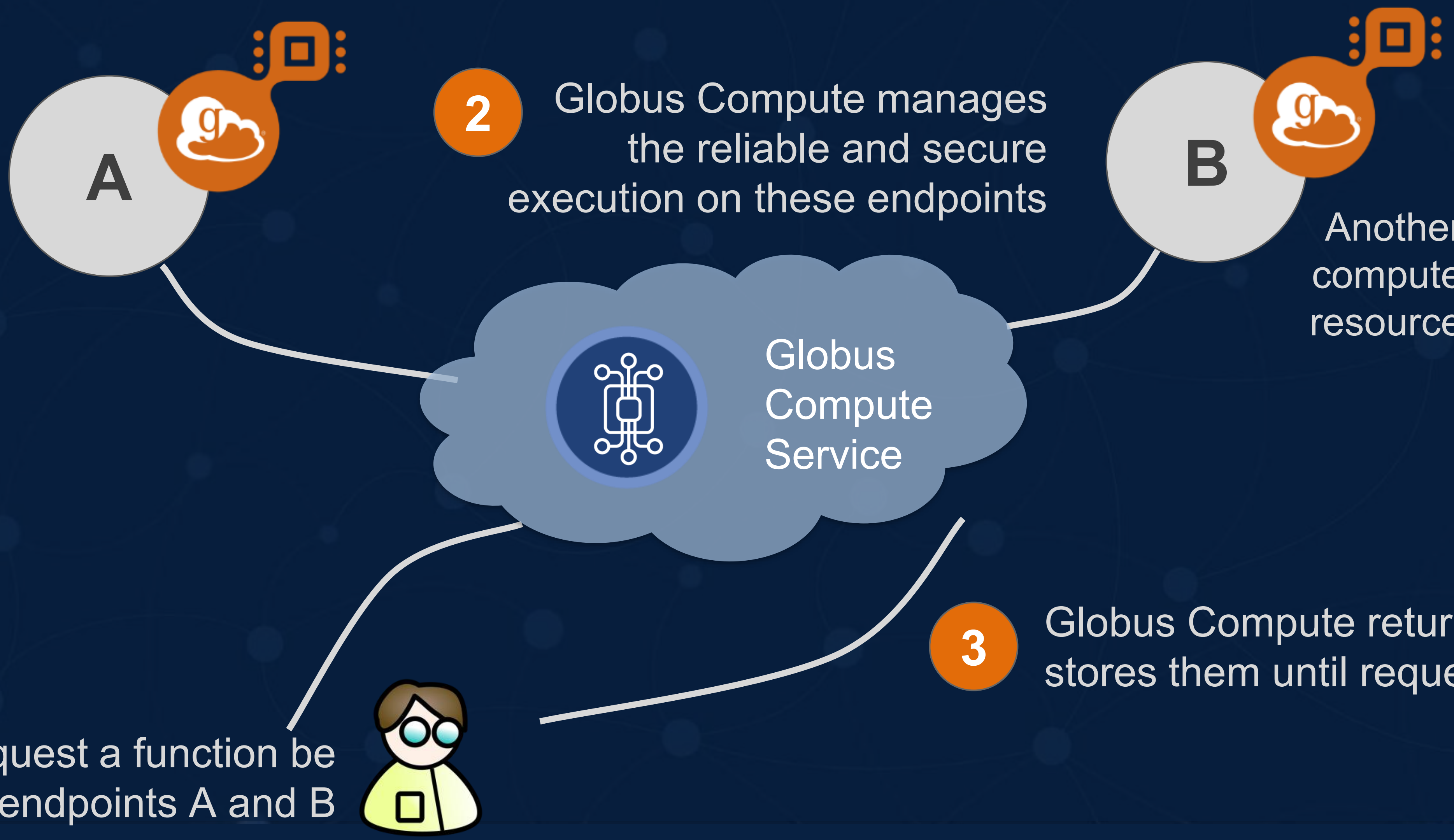


- FaaS for any compute resource
- Programmatic access to compute resources
- “Fire and forget” reliable execution
- Consistent user interface across diverse execution systems

Globus Compute components

- Compute **service** — Highly available cloud-hosted service for managed function execution
- Compute **endpoint** — Abstracts access to compute resources, from edge device to supercomputer
- Compute **SDK** — Python interface for interacting with the service, using the familiar (to many) Globus look and feel

How does it look from the researcher’s PoV?



A compute resource

2 Globus Compute manages the reliable and secure execution on these endpoints

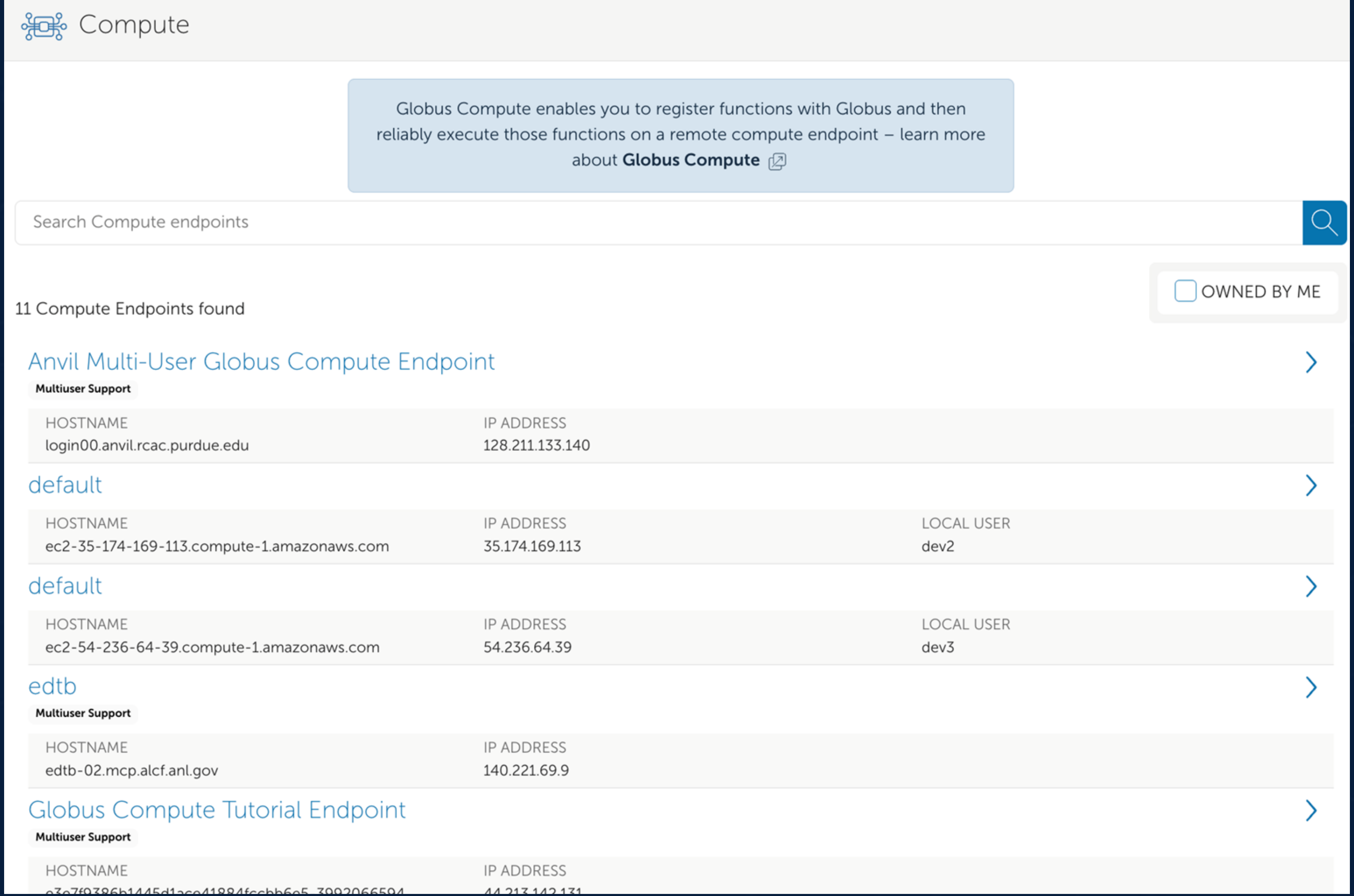
Another compute resource

Globus Compute Service

3 Globus Compute returns results or stores them until requested

1 You request a function be executed on endpoints A and B

Globus Web App interfaces



Compute

Globus Compute enables you to register functions with Globus and then reliably execute those functions on a remote compute endpoint – learn more about **Globus Compute**

Search Compute endpoints

□ OWNED BY ME

11 Compute Endpoints found

Anvil Multi-User Globus Compute Endpoint

Multiuser Support

HOSTNAME	IP ADDRESS
login00.anvil.rcac.purdue.edu	128.211.133.140

default

HOSTNAME	IP ADDRESS	LOCAL USER
ec2-35-174-169-113.compute-1.amazonaws.com	35.174.169.113	dev2

default

HOSTNAME	IP ADDRESS	LOCAL USER
ec2-54-236-64-39.compute-1.amazonaws.com	54.236.64.39	dev3

edtb

Multiuser Support

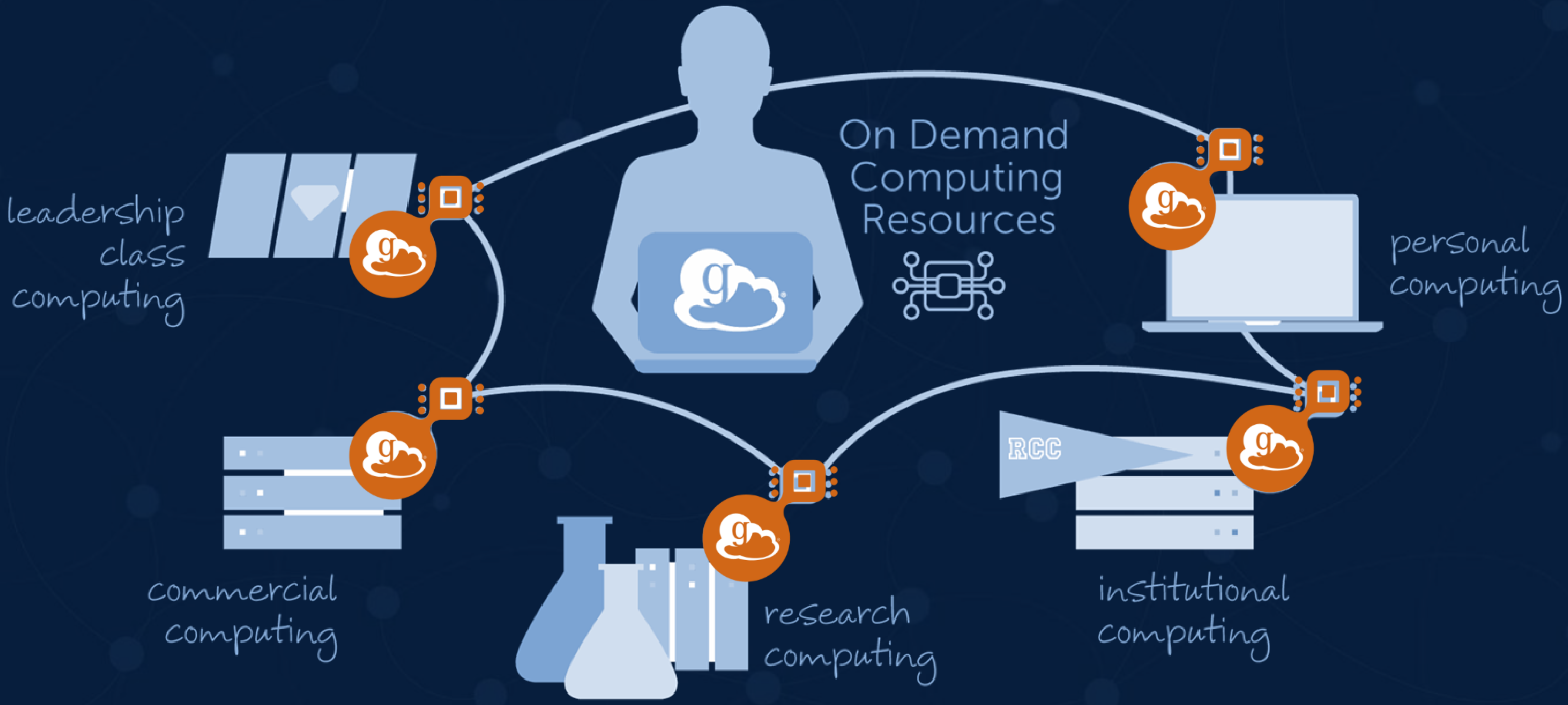
HOSTNAME	IP ADDRESS
edtb-02.mcp.alcf.anl.gov	140.221.69.9

Globus Compute Tutorial Endpoint

Multiuser Support

HOSTNAME	IP ADDRESS
e3e7f9386b1445d1ace41884fccbb6e5-3992066594	44.213.142.131

Globus Compute federates your existing CI



Globus Compute Agent

- **Python pip installable agent (or standard system package for admins)**
- **Elastic resource provisioning from local, cluster, or cloud system (via Parsl)**
- **Parallel execution using local fork or via common schedulers**
  - Slurm, PBS, LSF, Cobalt, K8s

Globus Compute Agent – Single User

- “Personal” endpoint
- Requires no administrative access for install
- No inbound connections

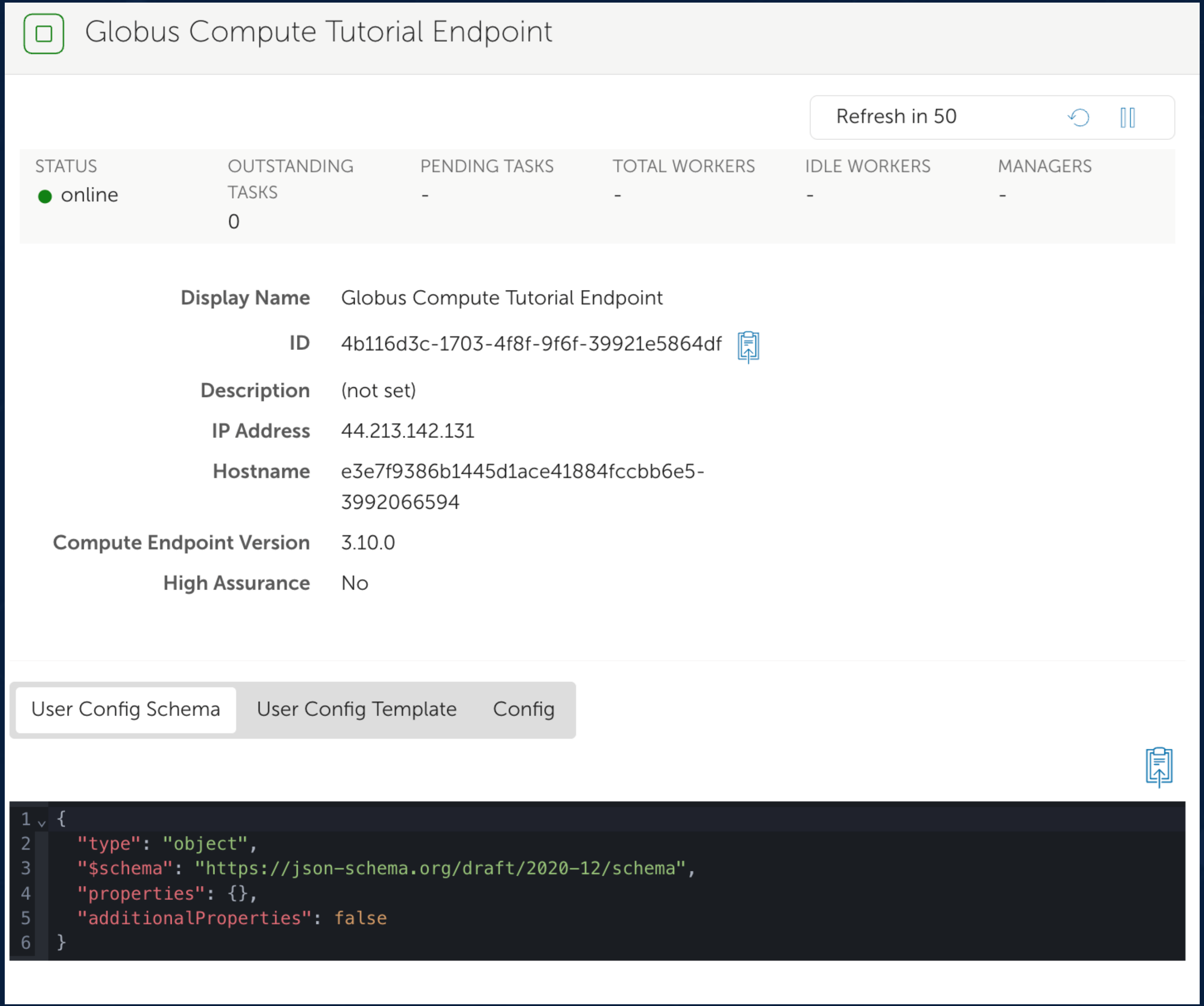
```
$ pip install globus-compute-endpoint
$ globus-compute-endpoint configure my-endpoint

Created profile for endpoint named <my-endpoint>
```

```
$ globus-compute-endpoint start my-endpoint
Starting endpoint; registered ID: 54460200-b652-4f43-a918-02882fa6114a
```

Globus Compute Agent – Multi User

- **Administrator installed and managed**
- **Templatable (controllable) user endpoint configurations**
  - E.g., pre-choose SlurmProvider, PBSProvider; enforce limits
  - User specifies configuration at task submission
- **Access control**
  - Authentication policies (Compute Service)
  - Identity mapping (Endpoint level)
- **Restrict functions that can be run**
- **High assurance deployments for protected data (including BAA)**
- **No inbound connections**

Globus Compute Tutorial Endpoint



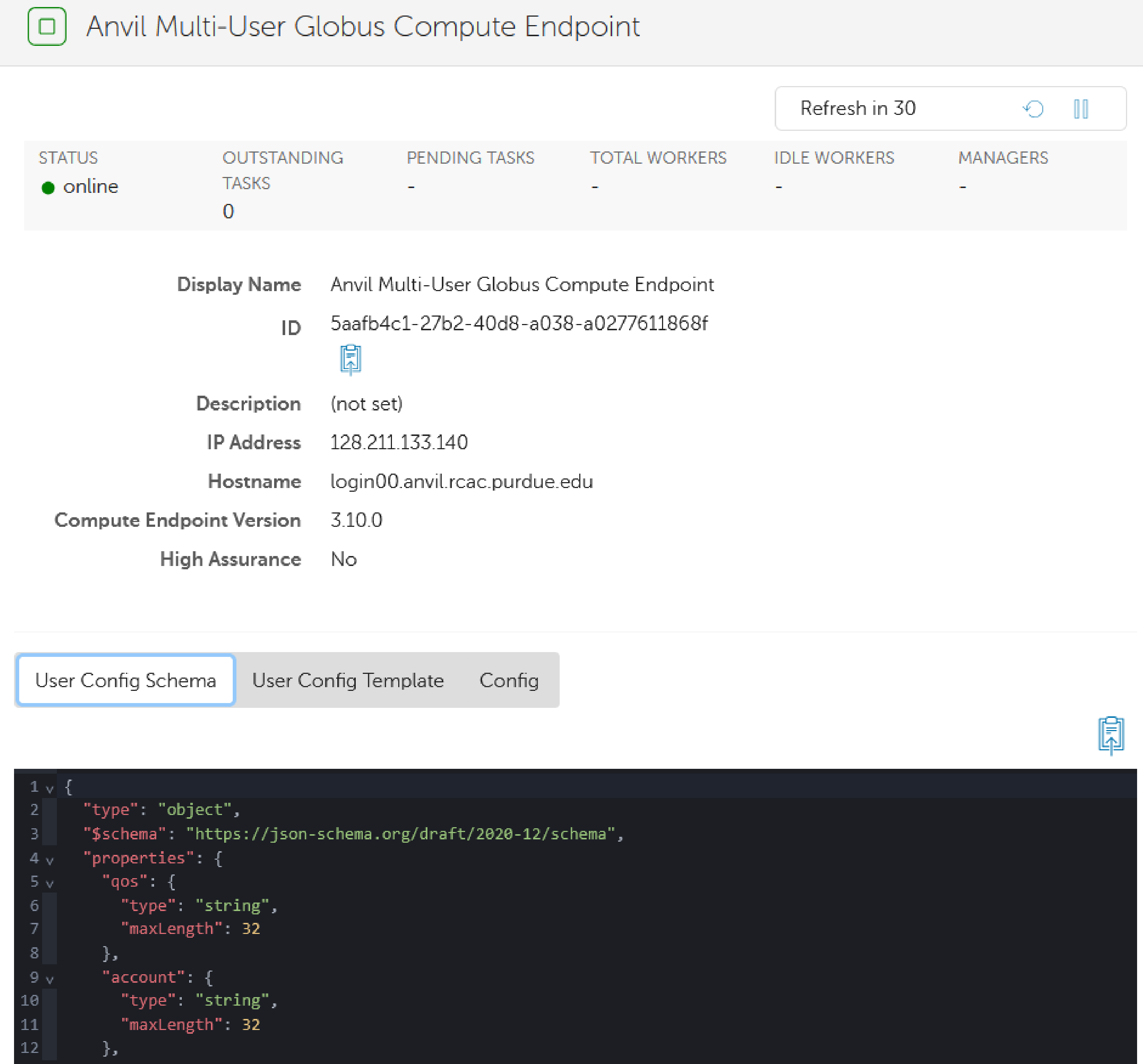
Globus Compute Tutorial Endpoint

STATUS	OUTSTANDING TASKS	PENDING TASKS	TOTAL WORKERS	IDLE WORKERS	MANAGERS
● online	0	-	-	-	-

Display Name	Globus Compute Tutorial Endpoint
ID	4b116d3c-1703-4f8f-9f6f-39921e5864df
Description	(not set)
IP Address	44.213.142.131
Hostname	e3e7f9386b1445d1ace41884fccbb6e5-3992066594
Compute Endpoint Version	3.10.0
High Assurance	No

```
{
  "type": "object",
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "properties": {},
  "additionalProperties": false
}
```

Purdue Anvil Compute Endpoint



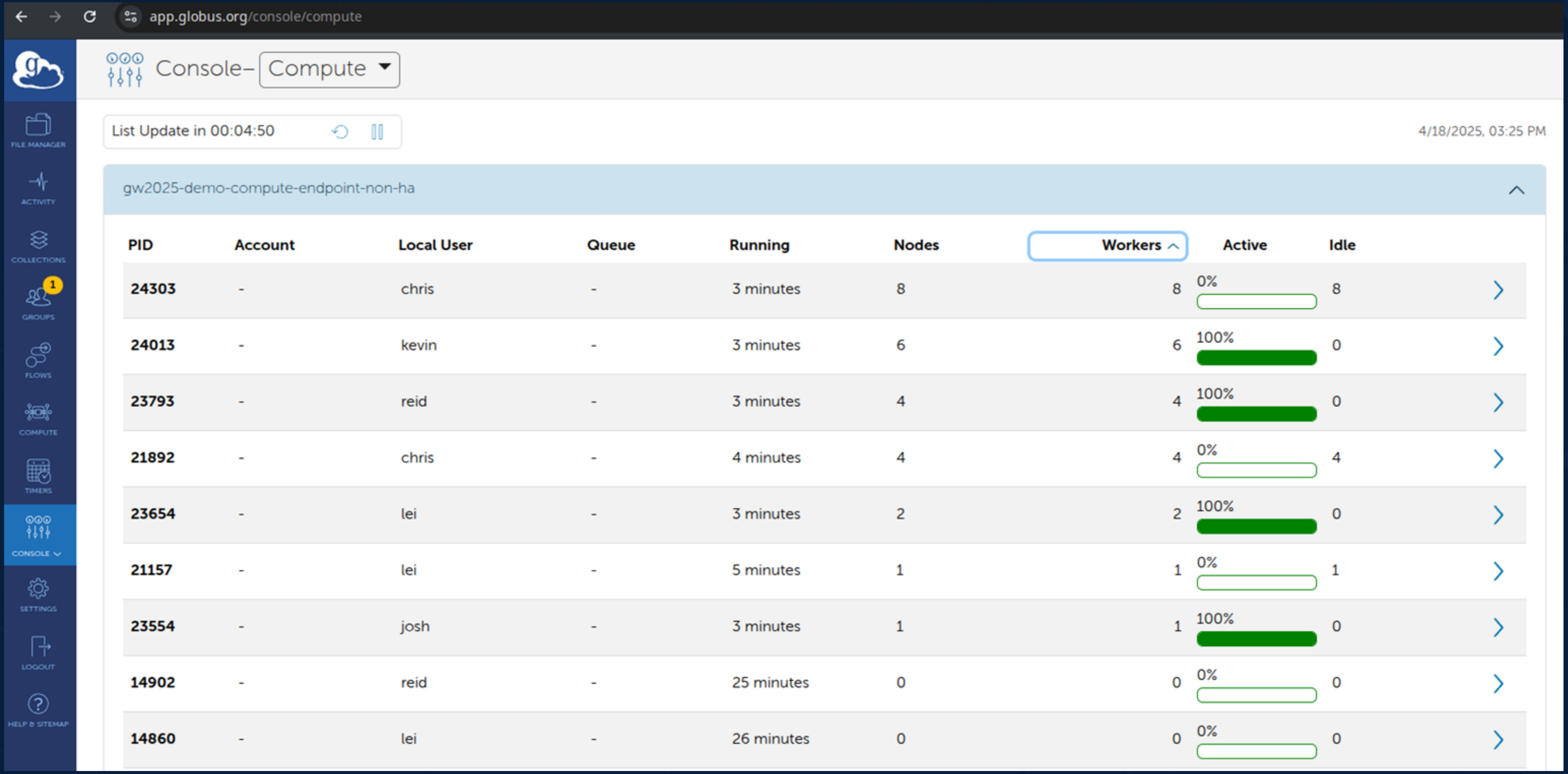
Anvil Multi-User Globus Compute Endpoint

STATUS	OUTSTANDING TASKS	PENDING TASKS	TOTAL WORKERS	IDLE WORKERS	MANAGERS
● online	0	-	-	-	-

Display Name	Anvil Multi-User Globus Compute Endpoint
ID	5aafb4c1-27b2-40d8-a038-a0277611868f
Description	(not set)
IP Address	128.211.133.140
Hostname	login00.anvil.rcac.purdue.edu
Compute Endpoint Version	3.10.0
High Assurance	No

```
{
  "type": "object",
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "properties": {
    "qos": {
      "type": "string",
      "maxLength": 32
    },
    "account": {
      "type": "string",
      "maxLength": 32
    },
```

Administrative Console



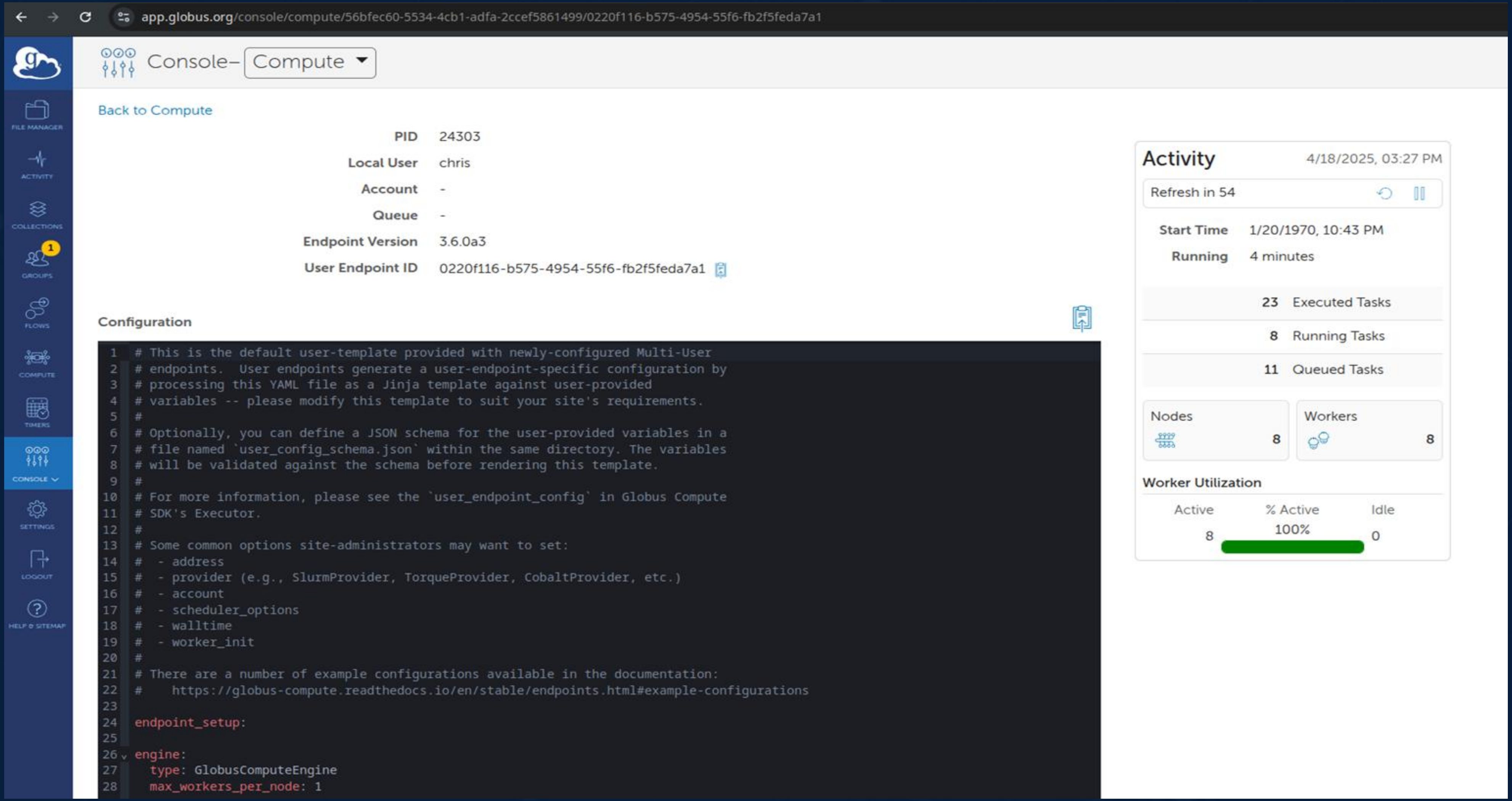
List Update in 00:04:50

4/18/2025, 03:25 PM

gw2025-demo-compute-endpoint-non-ha

PID	Account	Local User	Queue	Running	Nodes	Workers	Active	Idle
24303	-	chris	-	3 minutes	8	8	0%	8
24013	-	kevin	-	3 minutes	6	6	100%	0
23793	-	reid	-	3 minutes	4	4	100%	0
21892	-	chris	-	4 minutes	4	4	0%	4
23654	-	lei	-	3 minutes	2	2	100%	0
21157	-	lei	-	5 minutes	1	1	0%	1
23554	-	josh	-	3 minutes	1	1	100%	0
14902	-	reid	-	25 minutes	0	0	0%	0
14860	-	lei	-	26 minutes	0	0	0%	0

Administrative Console



PID	24303
Local User	chris
Account	-
Queue	-
Endpoint Version	3.6.0a3
User Endpoint ID	0220f116-b575-4954-55f6-fb2f5feda7a1

Configuration

```
# This is the default user-template provided with newly-configured Multi-User
# endpoints.  User endpoints generate a user-endpoint-specific configuration by
# processing this YAML file as a Jinja template against user-provided
# variables -- please modify this template to suit your site's requirements.
#
# Optionally, you can define a JSON schema for the user-provided variables in a
# file named `user_config_schema.json` within the same directory. The variables
# will be validated against the schema before rendering this template.
#
# For more information, please see the `user_endpoint_config` in Globus Compute
# SDK's Executor.
#
# Some common options site-administrators may want to set:
#  - address
#  - provider (e.g., SlurmProvider, TorqueProvider, CobaltProvider, etc.)
#  - account
#  - scheduler_options
#  - walltime
#  - worker_init
#
# There are a number of example configurations available in the documentation:
#    https://globus-compute.readthedocs.io/en/stable/endpoints.html#example-configurations

endpoint_setup:

engine:
  type: GlobusComputeEngine
  max_workers_per_node: 1
```

Activity 4/18/2025, 03:27 PM

Refresh in 54

Start Time	1/20/1970, 10:43 PM
Running	4 minutes

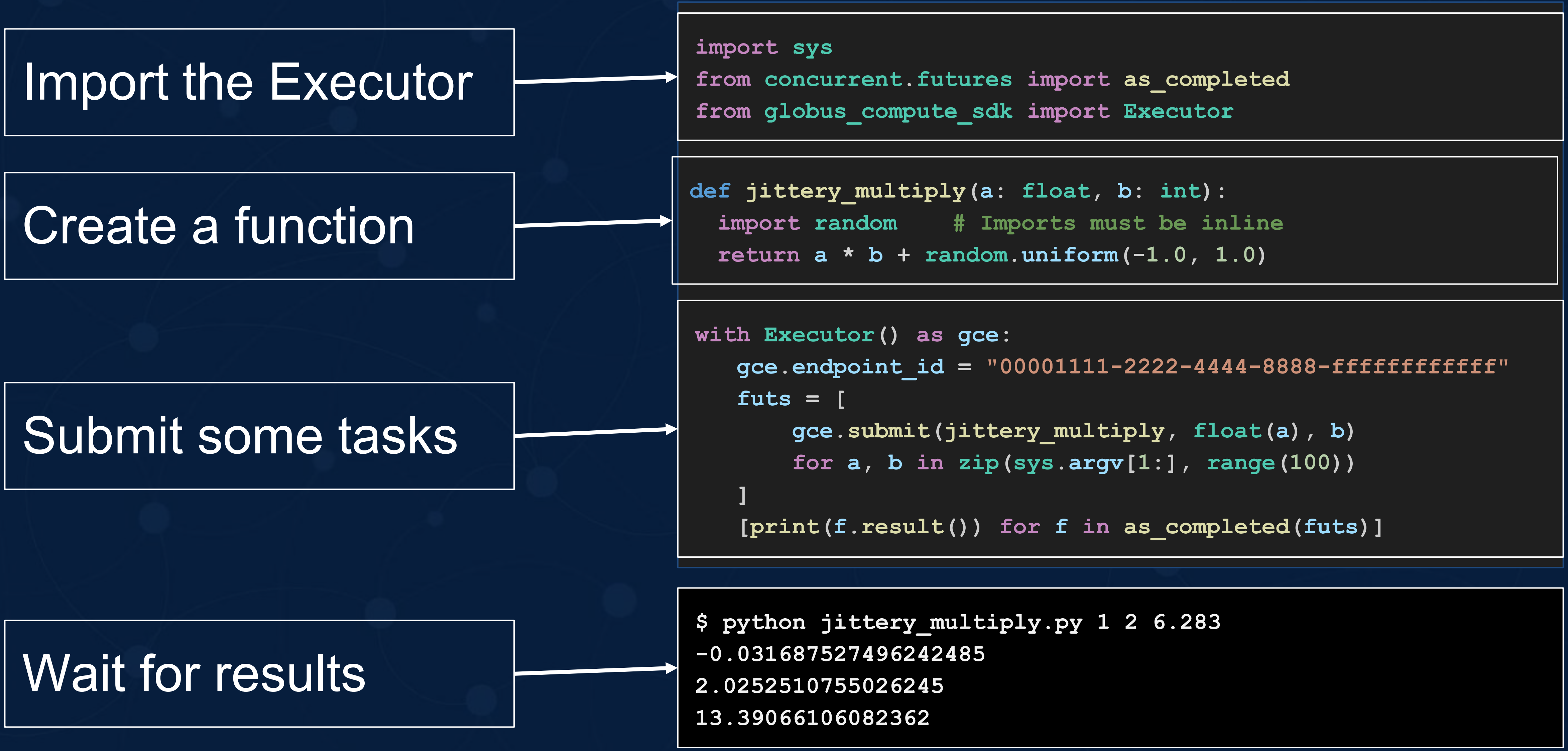
23	Executed Tasks
8	Running Tasks
11	Queued Tasks

Nodes	Workers
8	8

Worker Utilization

Active	% Active	Idle
8	100%	0

Using the SDK



Import the Executor

```
import sys
from concurrent.futures import as_completed
from globus_compute_sdk import Executor
```

Create a function

```
def jittery_multiply(a: float, b: int):
  import random     # Imports must be inline
  return a * b + random.uniform(-1.0, 1.0)
```

Submit some tasks

```
with Executor() as gce:
   gce.endpoint_id = "00001111-2222-4444-8888-ffffffffffff"
   futs = [
       gce.submit(jittery_multiply, float(a), b)
       for a, b in zip(sys.argv[1:], range(100))
   ]
   [print(f.result()) for f in as_completed(futs)]
```

Wait for results

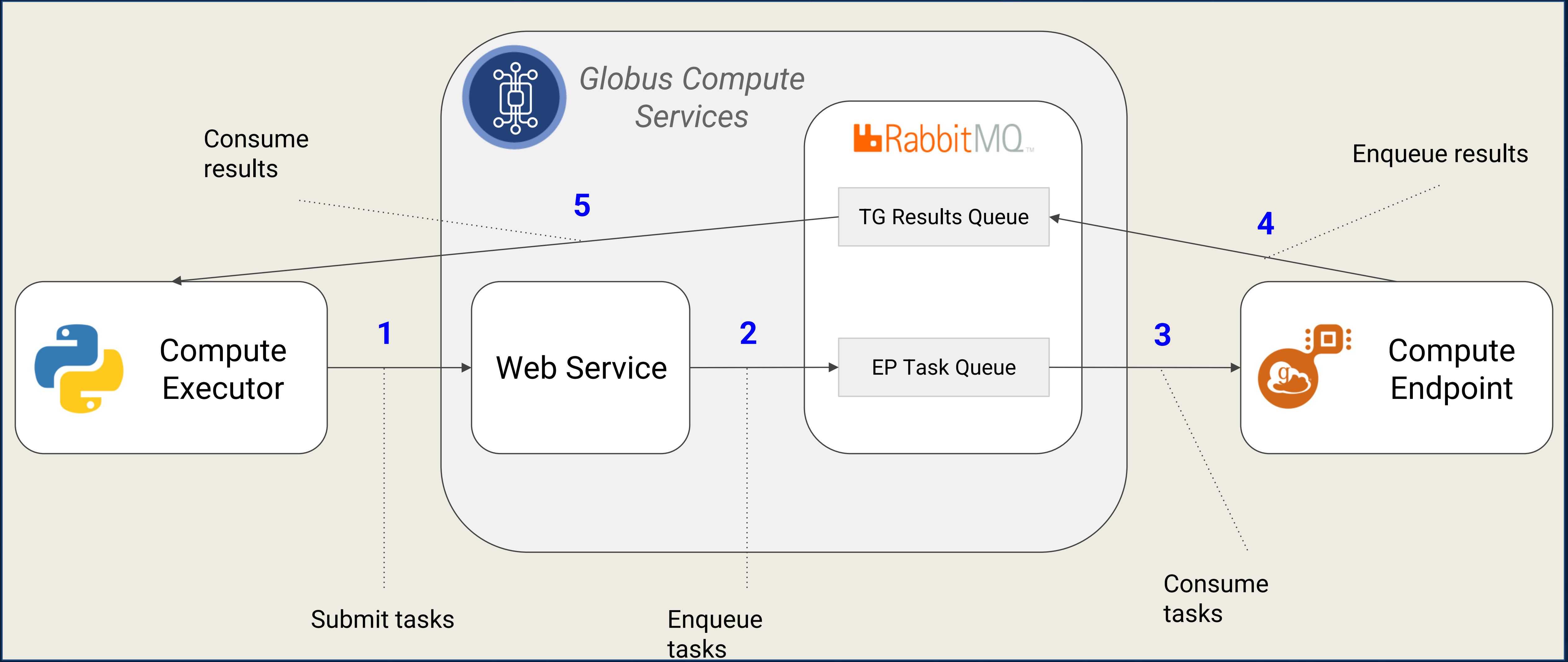
```
$ python jittery_multiply.py 1 2 6.283
-0.031687527496242485
2.0252510755026245
13.39066106082362
```

Using the SDK

Follow link to https://jupyter.demo.globus.org/

Open `globus-jupyter-notebooks/Compute_Introduction.ipynb`

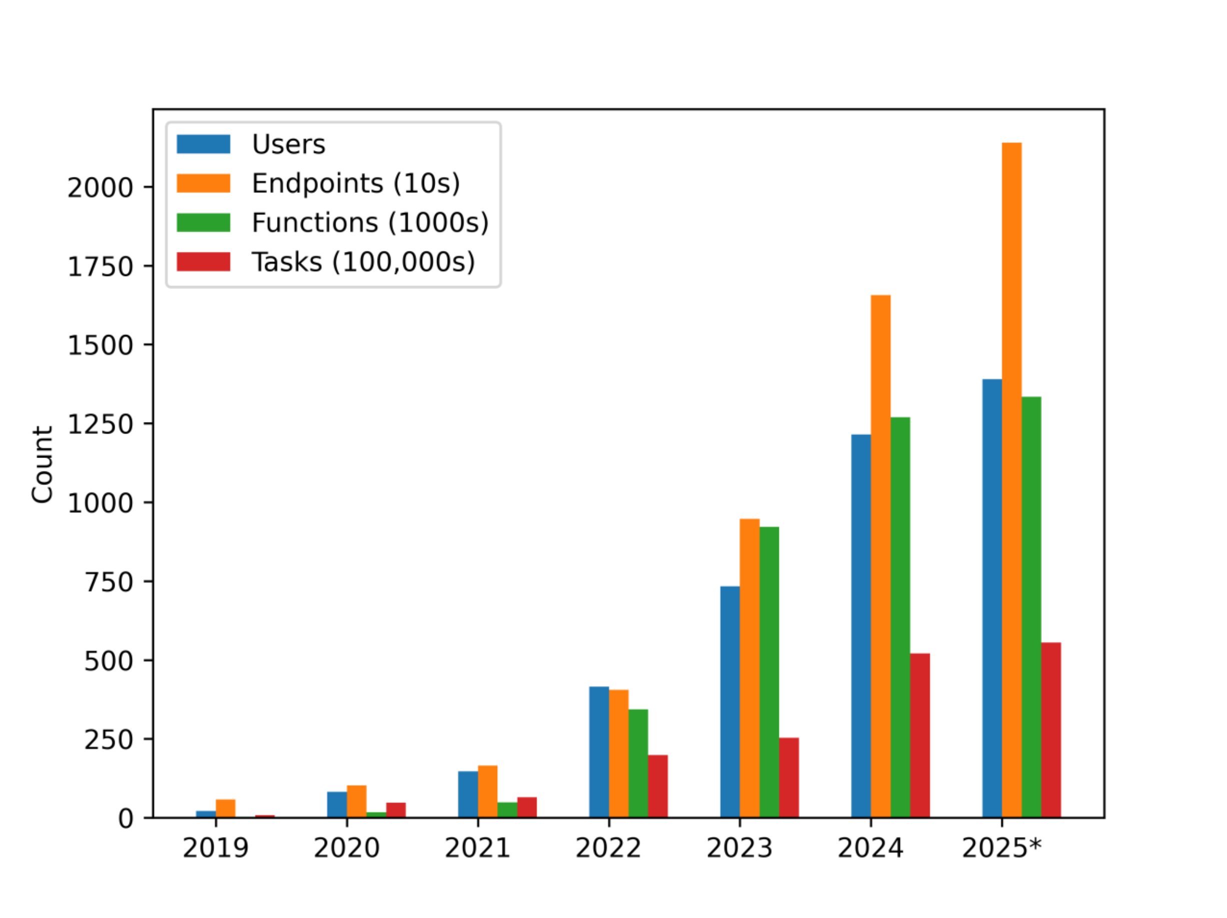
A peak under the hood



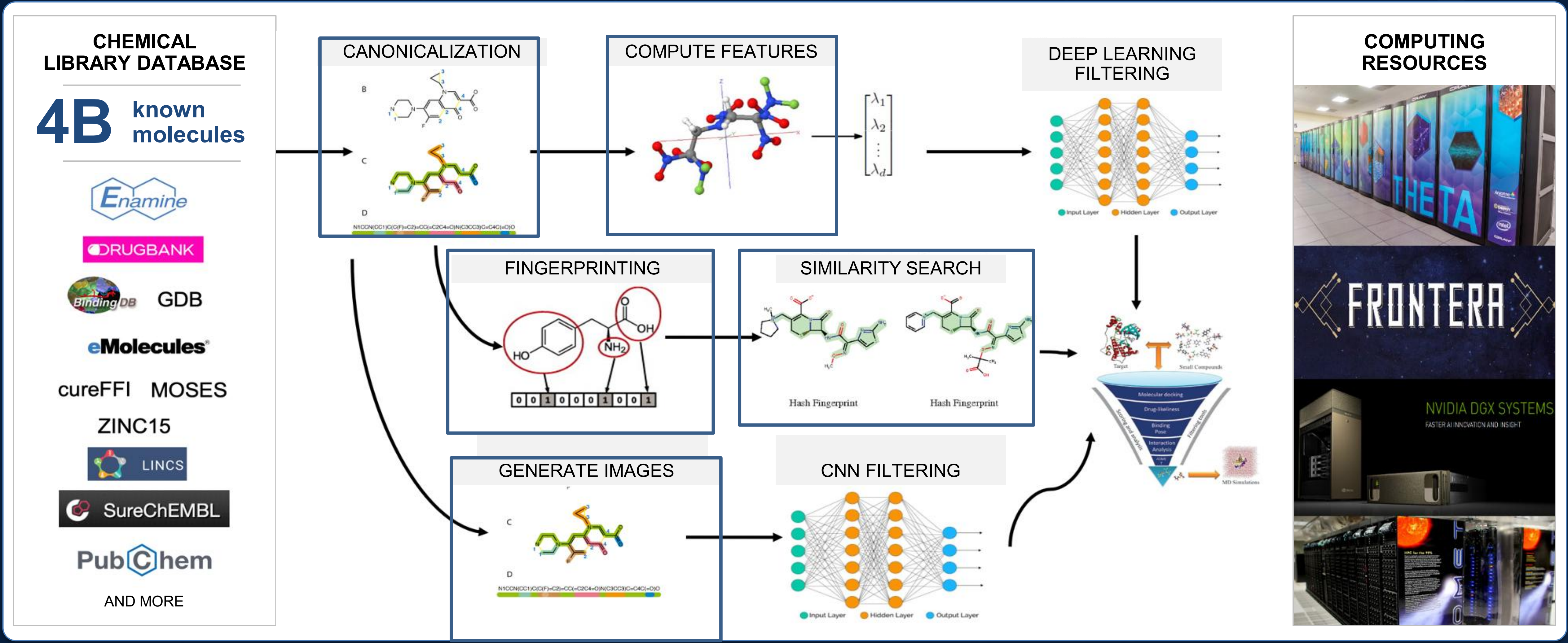
Usage is growing rapidly

Adopters generally in one of three categories:

- **Remote (bag-of-tasks) execution**
- **Research automation**
- **Platform for building other services**



Application: Using AI and supercomputers to accelerate drug development



CHEMICAL LIBRARY DATABASE

4B known molecules

Enamine

DRUGBANK

BindingDB GDB

eMolecules

cureFFI MOSES

ZINC15

LINCS

SureChEMBL

PubChem

AND MORE

CANONICALIZATION

COMPUTE FEATURES

DEEP LEARNING FILTERING

FINGERPRINTING

SIMILARITY SEARCH

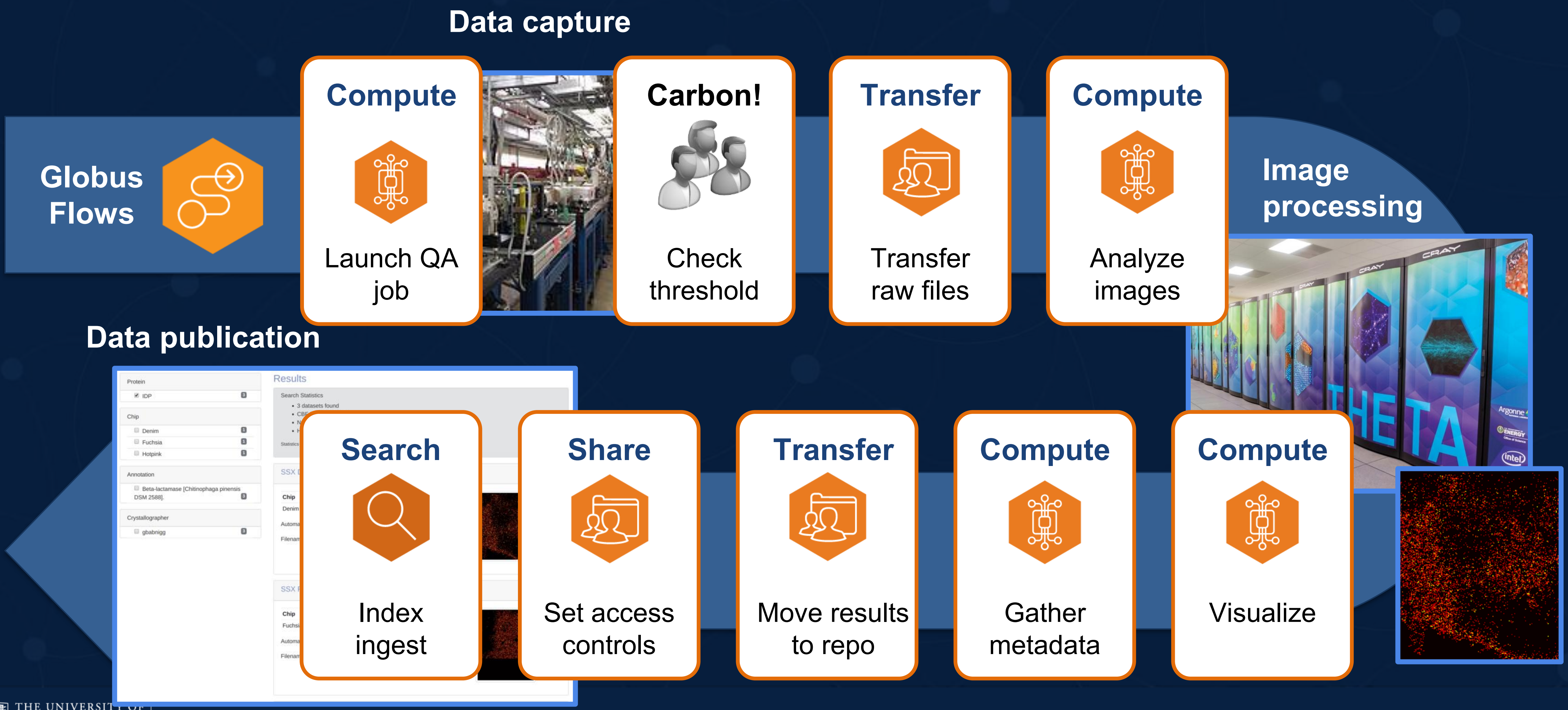
GENERATE IMAGES

CNN FILTERING

COMPUTING RESOURCES

Babuji, et al. Targeting SARS-CoV2 with AI- and HPC-enabled Lead Generation: A First Data Release.

Automation: Serial crystallography



Data capture

Globus Flows

Compute Launch QA job

Carbon! Check threshold

Transfer Transfer raw files

Compute Analyze images

Image processing

Compute Visualize

Compute Gather metadata

Transfer Move results to repo

Share Set access controls

Search Index ingest

Data publication

Protein

☑ IDP 3

Chip

□ Denim 1

□ Fuchsia 1

□ Hotpink 1

Annotation

□ Beta-lactamase [Chitinophaga pinensis DSM 2588]. 3

Crystallographer

□ gbabnigg 3

Results

Search Statistics

- 3 datasets found
- CBF
- N
- H

Statistics

SSX D

Chip

Denim

Automa

Filenam

SSX F

Chip

Fuchsi

Automa

Filenam

Common use cases

- Easily scale from laptop to cluster to cloud to supercomputer
- Seamlessly move between allocations on different systems
- Drive compute from a laptop (e.g., via Jupyter)
- Outsource management of a batch of tasks and retrieve results at some much later stage
- Gateways, community accounts via sharing of endpoints and functions
- Part of automated flows (often to perform actions for which there is no action provider)

Next: Globus Compute Single User Endpoint

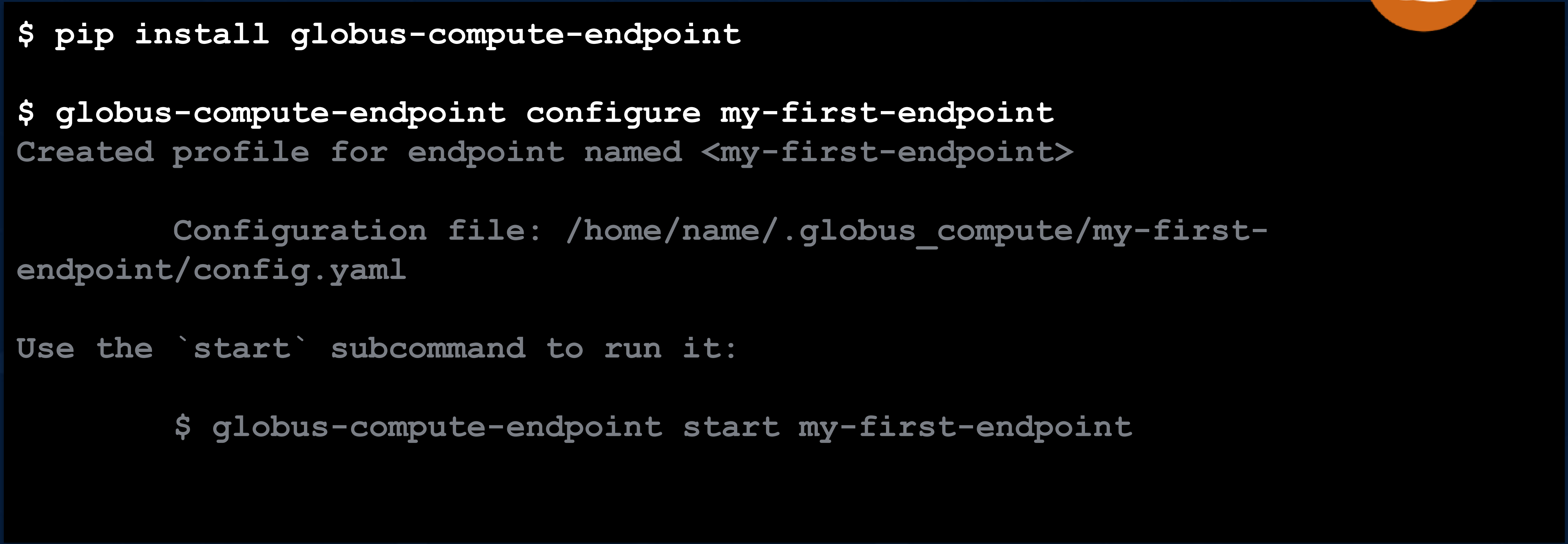
Types of Globus Compute endpoints

- **Single-user endpoint (SEP)**
  - No admin privileges to install
  - Runs as the owning user
  - Upon receiving a task, just works it (no identity mapping involved)
  - Fixed job configuration - one-trick pony (but easy to have a stable!)*

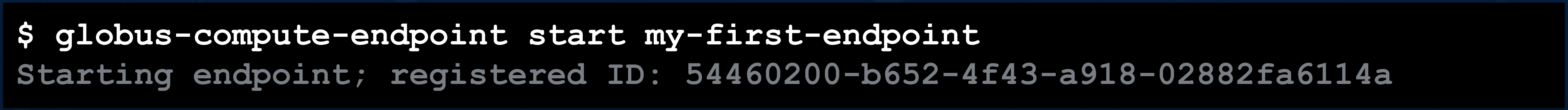
- **Multi-user endpoint (MEP)**
  - Installed by sysadmins
  - Runs as root
  - Upon receiving a task, maps identity to a local user, drops privileges and forks a transient User Endpoint as this user.
  - The UEP then works the task(s)
  - Template-able, allows flexible specification of job parameters by users*

  - Outbound connections only
  - * SEPs can be made template-able and can behave like MEPs, too (*sans* identity mapping)

Installing single user compute endpoint



```
$ pip install globus-compute-endpoint

$ globus-compute-endpoint configure my-first-endpoint
Created profile for endpoint named <my-first-endpoint>

        Configuration file: /home/name/.globus_compute/my-first-
endpoint/config.yaml

Use the `start` subcommand to run it:

        $ globus-compute-endpoint start my-first-endpoint
```

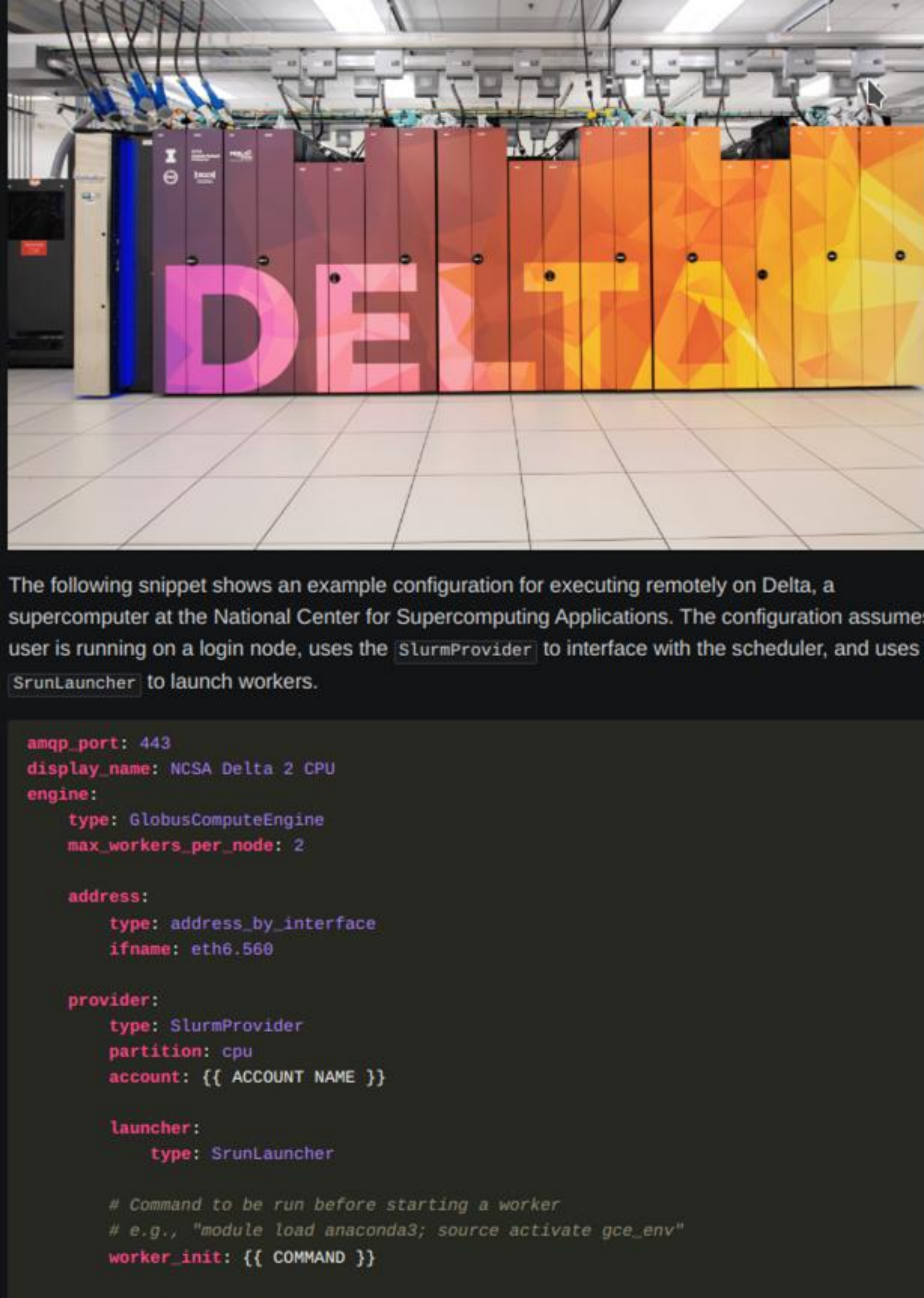


```
$ globus-compute-endpoint start my-first-endpoint
Starting endpoint; registered ID: 54460200-b652-4f43-a918-02882fa6114a
```

Configuring a single user compute endpoint

```
# ~/.globus_compute/my-first-endpoint/config.yaml
amqp_port: 443
display_name: My Endpoint
engine:
 type: GlobusComputeEngine
 provider:
    type: LocalProvider
```

https://globus-compute.readthedocs.io/en/latest/endpoints.html#example-configurations

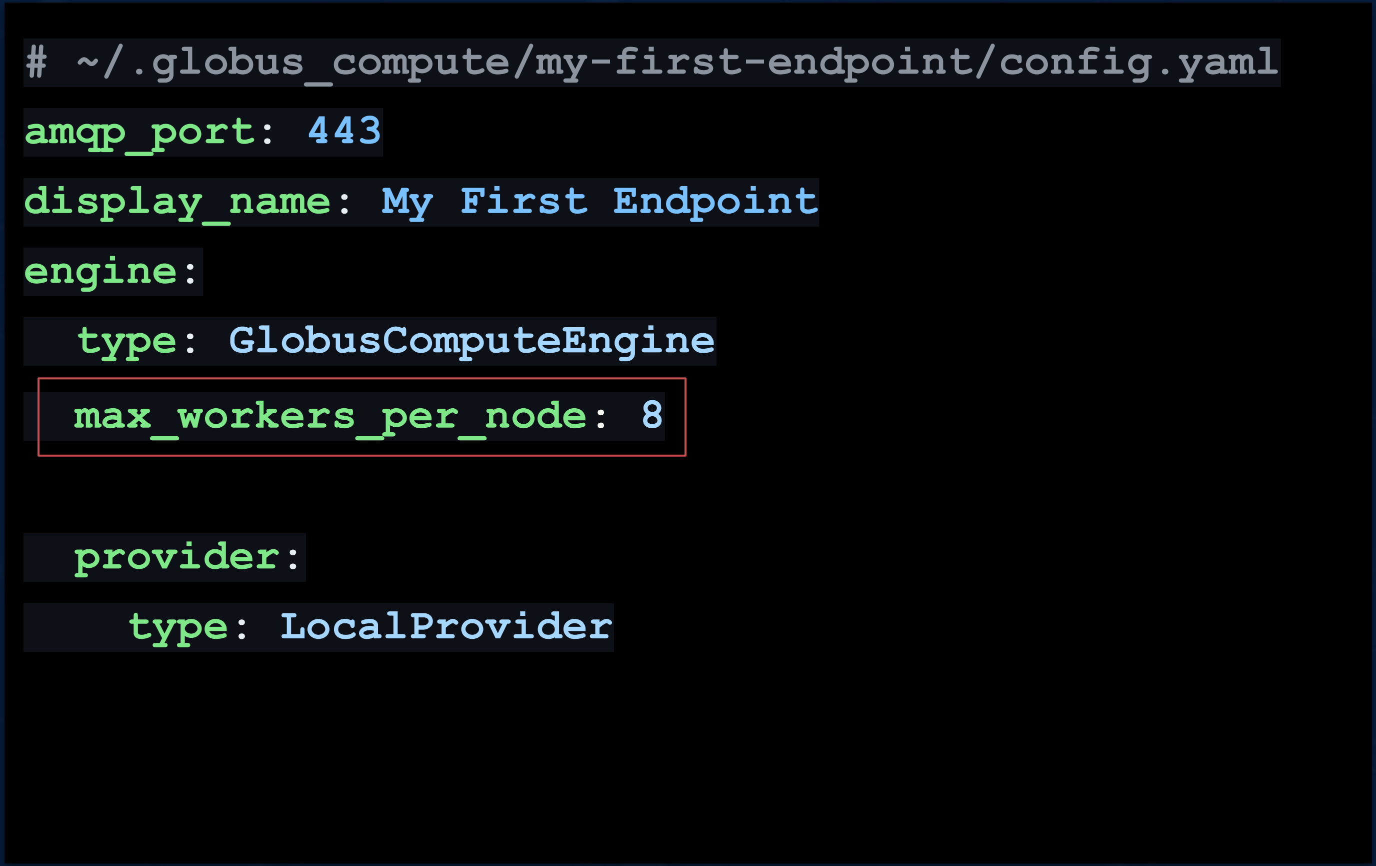


The following snippet shows an example configuration for executing remotely on Delta, a supercomputer at the National Center for Supercomputing Applications. The configuration assume user is running on a login node, uses the `SlurmProvider` to interface with the scheduler, and uses `SrunLauncher` to launch workers.

```
amqp_port: 443
display_name: NCSA Delta 2 CPU
engine:
    type: GlobusComputeEngine
    max_workers_per_node: 2

    address:
        type: address_by_interface
        ifname: eth6.560

    provider:
        type: SlurmProvider
        partition: cpu
        account: {{ ACCOUNT NAME }}

        launcher:
            type: SrunLauncher

        # Command to be run before starting a worker
        # e.g., "module load anaconda3; source activate gce_env"
        worker_init: {{ COMMAND }}
```

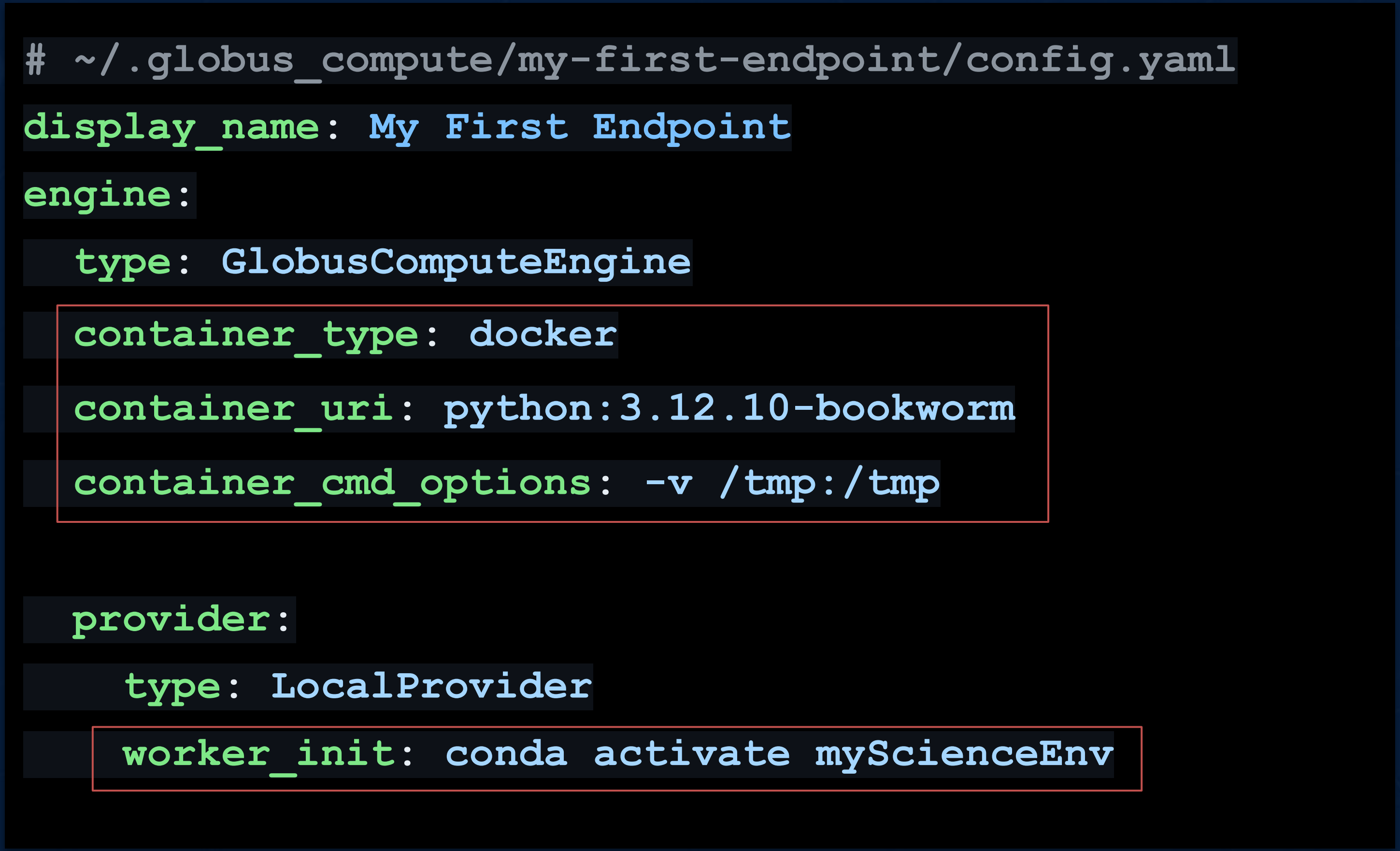
THE UNIVERSITY OF CHICAGO RESEARCH

Configuring endpoints - Scaling



```yaml
# ~/.globus_compute/my-first-endpoint/config.yaml
amqp_port: 443
display_name: My First Endpoint
engine:
  type: GlobusComputeEngine
  max_workers_per_node: 8

  provider:
    type: LocalProvider
```

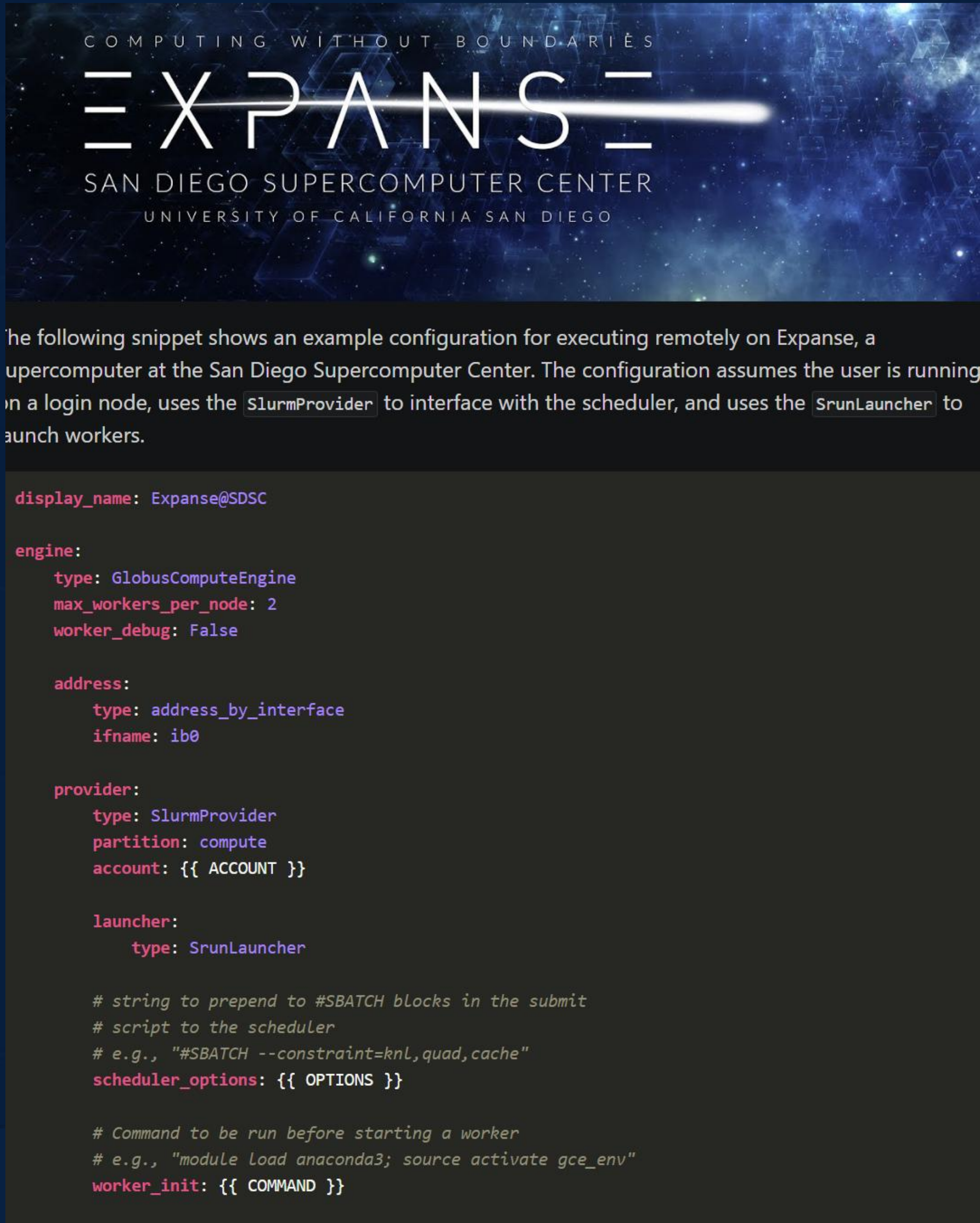
Managing the Execution Environment



```yaml
# ~/.globus_compute/my-first-endpoint/config.yaml
display_name: My First Endpoint
engine:
  type: GlobusComputeEngine
  container_type: docker
  container_uri: python:3.12.10-bookworm
  container_cmd_options: -v /tmp:/tmp

  provider:
    type: LocalProvider
    worker_init: conda activate myScienceEnv
```

Configuring endpoints - Batch Schedulers

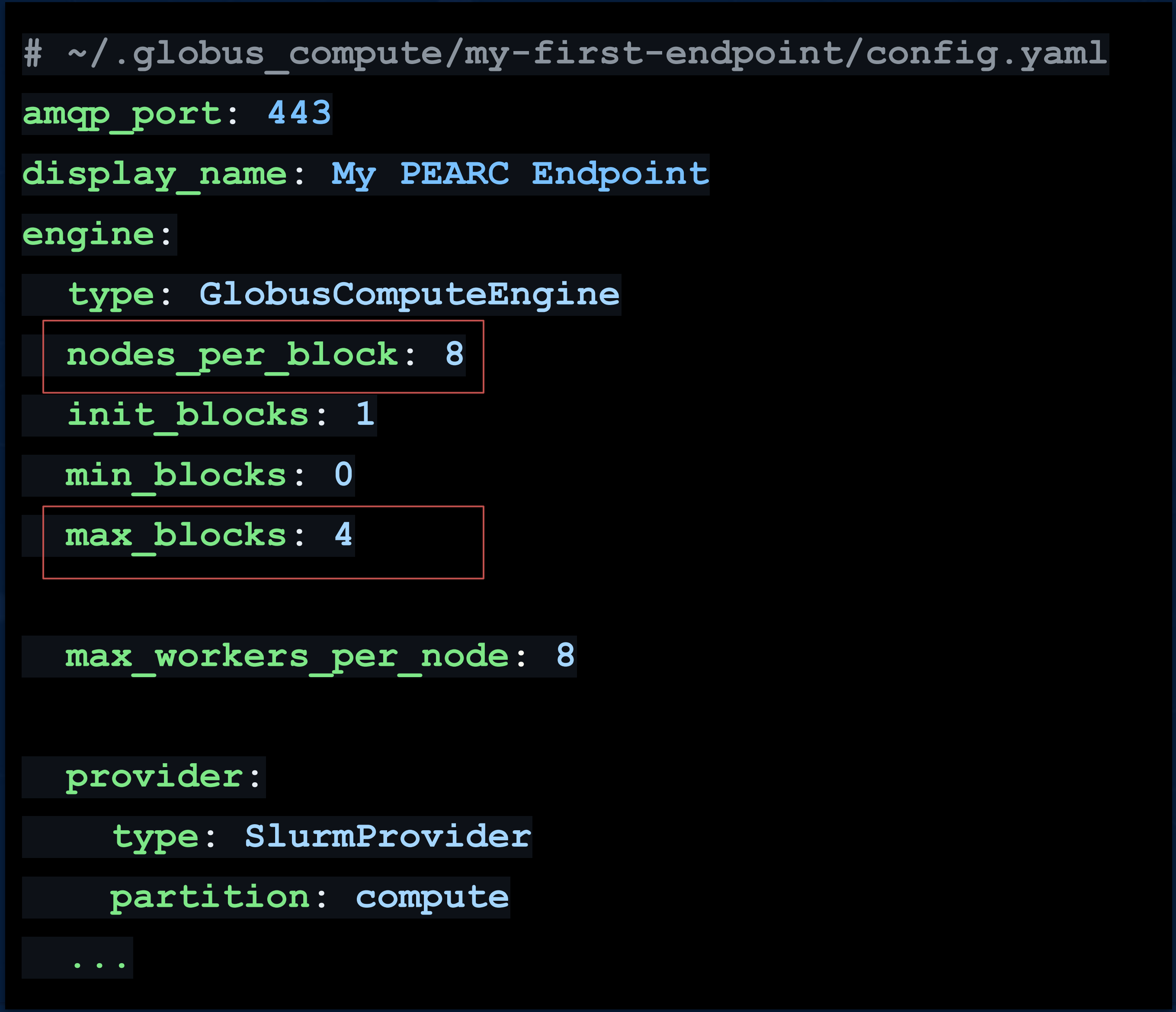
```yaml
# ~/.globus_compute/my-first-endpoint/config.yaml
amqp_port: 443
display_name: My First Endpoint
engine:
  provider:
    type: SlurmProvider
    partition: compute
    account: {{ ACCOUNT }}
    launcher:
      type: SrunLauncher
    scheduler_options: {{ OPTIONS}}
    worker_init: {{ COMMAND }}
    walltime: 01:00:00
    nodes_per_block: 1
  type: GlobusComputeEngine
  max_workers_per_node: 8
```


COMPUTING WITHOUT BOUNDARIES
EXPANSE
SAN DIEGO SUPERCOMPUTER CENTER
UNIVERSITY OF CALIFORNIA SAN DIEGO

he following snippet shows an example configuration for executing remotely on Expanse, a upercomputer at the San Diego Supercomputer Center. The configuration assumes the user is running n a login node, uses the `SlurmProvider` to interface with the scheduler, and uses the `SrunLauncher` to aunch workers.

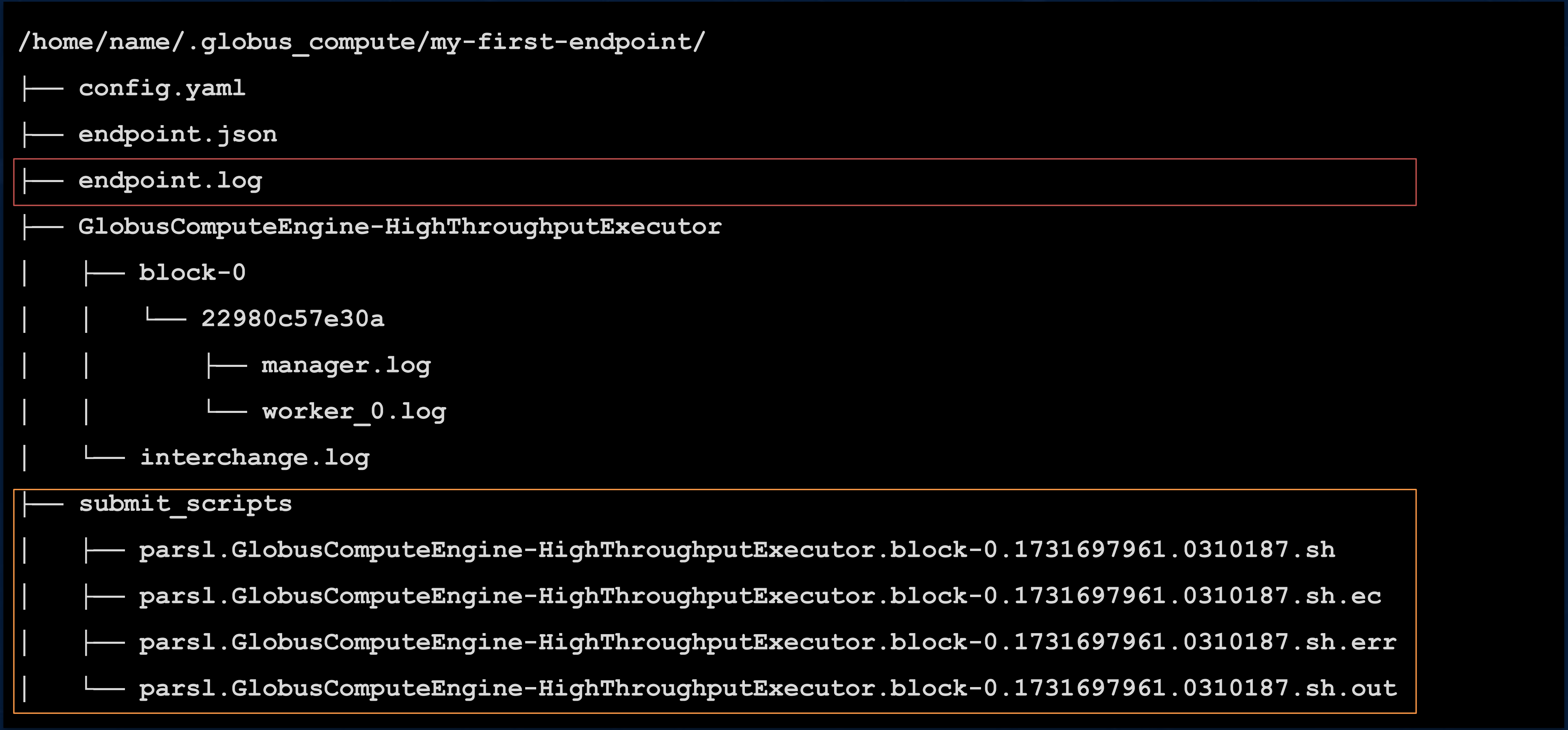
```yaml
display_name: Expanse@SDSC

engine:
    type: GlobusComputeEngine
    max_workers_per_node: 2
    worker_debug: False

    address:
        type: address_by_interface
        ifname: ib0

    provider:
        type: SlurmProvider
        partition: compute
        account: {{ ACCOUNT }}

        launcher:
            type: SrunLauncher

        # string to prepend to #SBATCH blocks in the submit
        # script to the scheduler
        # e.g., "#SBATCH --constraint=knl,quad,cache"
        scheduler_options: {{ OPTIONS }}

        # Command to be run before starting a worker
        # e.g., "module load anaconda3; source activate gce_env"
        worker_init: {{ COMMAND }}
```

Configuring endpoints - Scaling Batch Schedulers



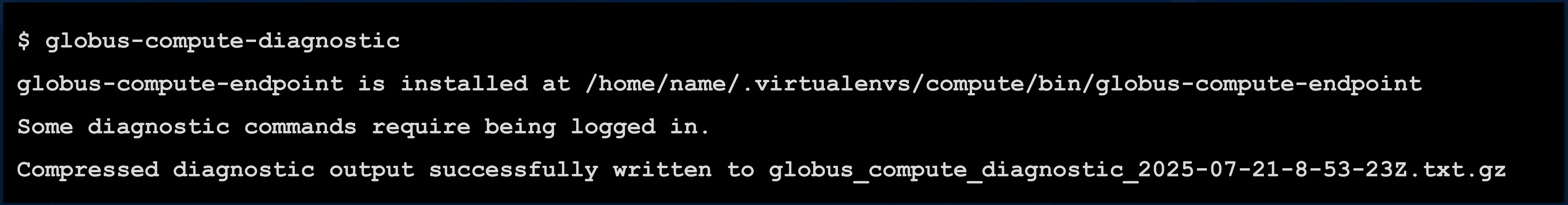
```
# ~/.globus_compute/my-first-endpoint/config.yaml
amqp_port: 443
display_name: My PEARC Endpoint
engine:
  type: GlobusComputeEngine
  nodes_per_block: 8
  init_blocks: 1
  min_blocks: 0
  max_blocks: 4

  max_workers_per_node: 8

  provider:
    type: SlurmProvider
    partition: compute
  ...
```

Debugging and Diagnostics



```
/home/name/.globus_compute/my-first-endpoint/
├── config.yaml
├── endpoint.json
├── endpoint.log
├── GlobusComputeEngine-HighThroughputExecutor
│   ├── block-0
│   │   └── 22980c57e30a
│   │       ├── manager.log
│   │       └── worker_0.log
│   └── interchange.log
├── submit_scripts
│   ├── parsl.GlobusComputeEngine-HighThroughputExecutor.block-0.1731697961.0310187.sh
│   ├── parsl.GlobusComputeEngine-HighThroughputExecutor.block-0.1731697961.0310187.sh.ec
│   ├── parsl.GlobusComputeEngine-HighThroughputExecutor.block-0.1731697961.0310187.sh.err
│   └── parsl.GlobusComputeEngine-HighThroughputExecutor.block-0.1731697961.0310187.sh.out
```

Debugging and Diagnostics



```
$ globus-compute-diagnostic
globus-compute-endpoint is installed at /home/name/.virtualenvs/compute/bin/globus-compute-endpoint
Some diagnostic commands require being logged in.
Compressed diagnostic output successfully written to globus_compute_diagnostic_2025-07-21-8-53-23Z.txt.gz
```

Next: Globus Compute Multi-User Endpoint

Multi-user Endpoint: Value add for users

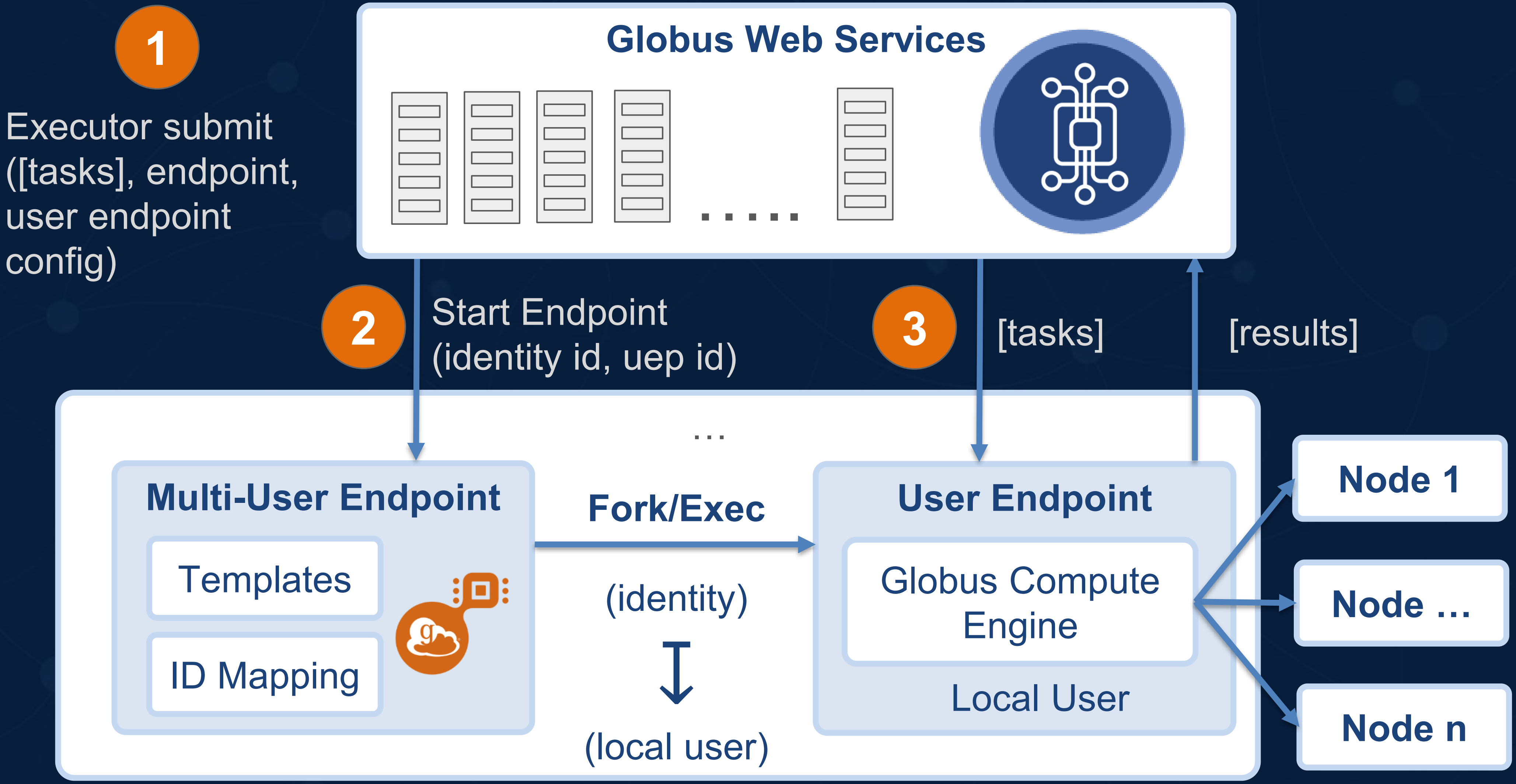
- No need to maintain multiple endpoints for different configurations
- Specify configuration **at task submission**
- No need to log into the target computer

* SEPs can be made template-able and flexible

Multi-user Endpoint: Value add for Admins

- **<u>Lower barrier for users</u>**
- Templatable (controllable) user endpoint configurations
  - E.g., pre-choose SlurmProvider, PBSProvider; enforce limits
- No orphaned user compute endpoints
  - Enforced process tree
  - Idle endpoints are shut down (per endpoint configuration)
- Standard Globus Identity Mapping
- Advanced authentication and authorization policies

Multi-user Endpoints: Architecture



Multi-user: Installation

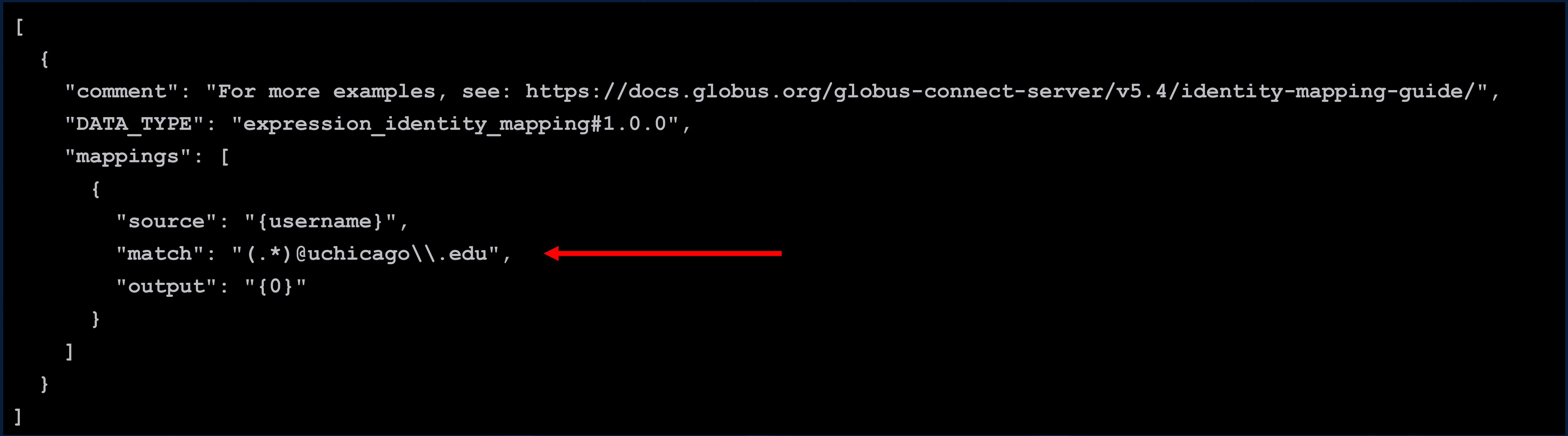
```
$ curl -LOs https://downloads.globus.org/globus-connect-server/stable/installers/repo/deb/globus-repo_latest_all.deb
$ dpkg -i globus-repo_latest_all.deb
$ apt-key add /usr/share/globus-repo/RPM-GPG-KEY-Globus

$ apt update
$ apt install globus-compute-agent
```

```
$ globus-compute-endpoint configure mycluster-endpoint-mu --multi-user
Created multi-user profile for endpoint named <mycluster-endpoint-mu>

 Configuration file: /root/.globus_compute/mycluster-endpoint-mu/config.yaml

 Example identity mapping configuration: /root/.globus_compute/mycluster-endpoint-
mu/example_identity_mapping_config.json

 User endpoint configuration template: /root/.globus_compute/mycluster-endpoint-mu/user_config_template.yaml.j2
 User endpoint configuration schema: /root/.globus_compute/mycluster-endpoint-mu/user_config_schema.json
 User endpoint environment variables: /root/.globus_compute/mycluster-endpoint-mu/user_environment.yaml

Use the `start` subcommand to run it:

        $ globus-compute-endpoint start mycluster-endpoint-mu
```

Multi-user: Identity Mapping

Same format as GCSv5

/root/.globus_compute/mycluster-endpoint-mu/example_identity_mapping_config.json



```
[
  {
    "comment": "For more examples, see: https://docs.globus.org/globus-connect-server/v5.4/identity-mapping-guide/",
    "DATA_TYPE": "expression_identity_mapping#1.0.0",
    "mappings": [
      {
        "source": "{username}",
        "match": "(.*)@uchicago\\.edu",
        "output": "{0}"
      }
    ]
  }
]
```

https://docs.globus.org/globus-connect-server/v5.4/identity-mapping-guide/#default_identity_to_username_mapping

Using the multi-user endpoint

```
# globus-compute-endpoint start mycluster-endpoint-mu
```

```python
def hello_world():
    return "Hello, World!"

with Executor(endpoint_id="...") as gce:
    future = gce.submit(hello_world)
    print(future.result())
```

```
$ python hello_world.py
Hello, World!
```

Multi-user: User Configuration Template

/root/.globus_compute/mycluster-endpoint-mu/user_config_template.yaml.j2

```
engine:
  type: GlobusComputeEngine
  max_workers_per_node: {{ WORKERS }}

  provider:
    type: LocalProvider
```

```
from globus_compute_sdk import Executor

uep_conf = {
  "WORKERS": 5,
}

with Executor(endpoint_id="...") as gce:
   gce.user_endpoint_config = uep_conf
   futures = []
   for i in range(5):

futures.append(gce.submit(hello_world))
   for f in futures:
       f.result()
```

Multi-user: User Configuration Template

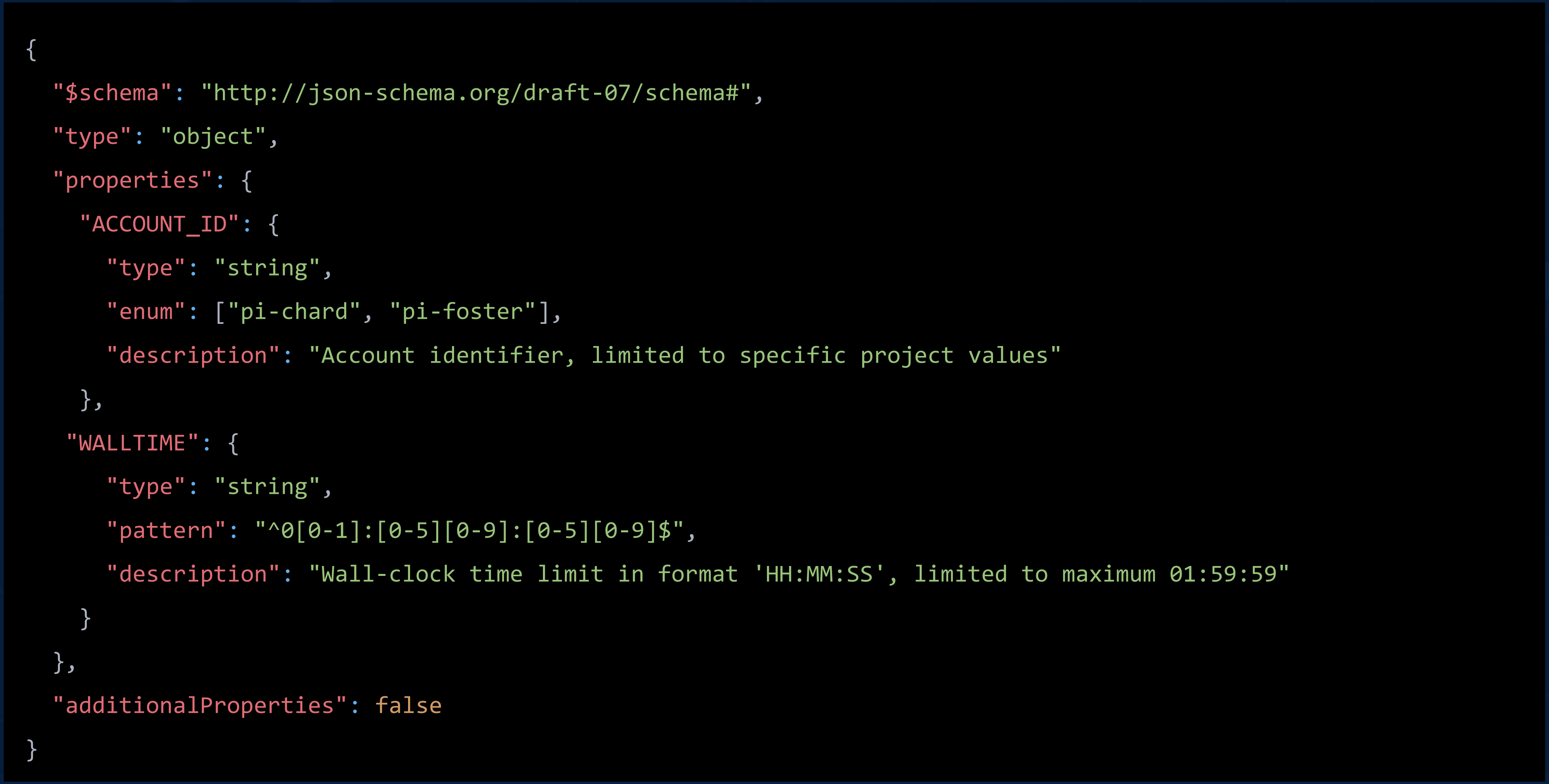
/root/.globus_compute/mycluster-endpoint-mu/user_config_template.yaml.j2

```
engine:
  type: GlobusComputeEngine

  provider:
    type: SlurmProvider
    partition: cpu
    account: {{ ACCOUNT_ID }}
    walltime: {{ WALLTIME|default("00:30:00") }}

    launcher:
      type: SrunLauncher
```

```
from globus_compute_sdk import Executor

uep_conf = {
  "ACCOUNT_ID": "314159265",
  "WALLTIME": "00:02:00"
}

with Executor(endpoint_id="...") as gce:
   gce.user_endpoint_config = uep_conf
   fut = gce.submit(hello_world)
   res = fut.result()
```

Multi-user: User Configuration Schema

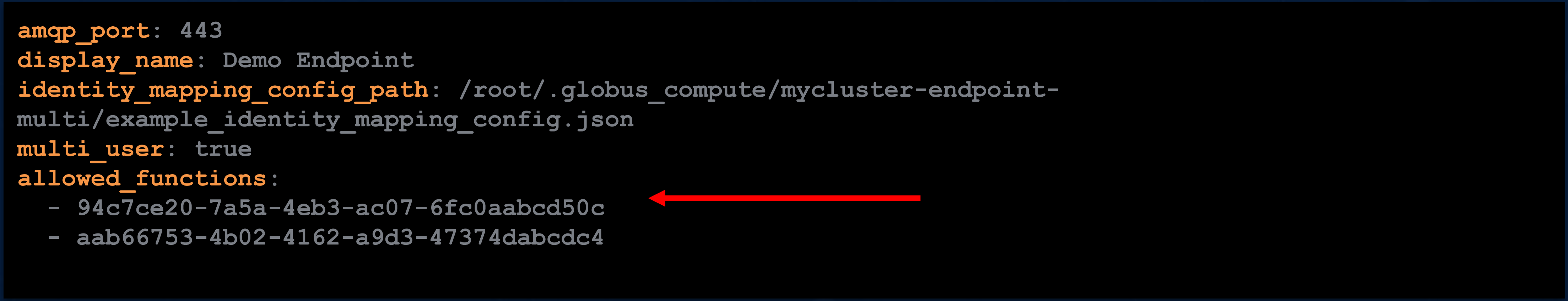
/root/.globus_compute/mycluster-endpoint-mu/user_config_schema.json



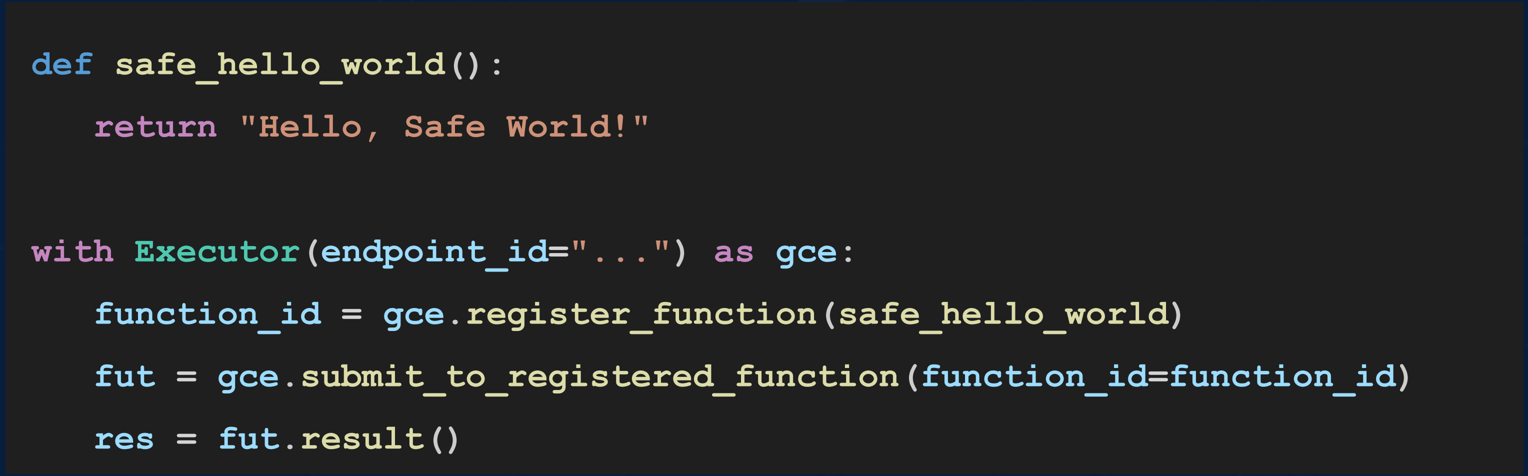
```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "object",
  "properties": {
    "ACCOUNT_ID": {
      "type": "string",
      "enum": ["pi-chard", "pi-foster"],
      "description": "Account identifier, limited to specific project values"
    },
   "WALLTIME": {
      "type": "string",
      "pattern": "^0[0-1]:[0-5][0-9]:[0-5][0-9]$",
      "description": "Wall-clock time limit in format 'HH:MM:SS', limited to maximum 01:59:59"
    }
  },
  "additionalProperties": false
}
```

Multi-user: Restricting Functions

/root/.globus_compute/mycluster-endpoint-mu/config.yaml



```yaml
amqp_port: 443
display_name: Demo Endpoint
identity_mapping_config_path: /root/.globus_compute/mycluster-endpoint-
multi/example_identity_mapping_config.json
multi_user: true
allowed_functions:
  - 94c7ce20-7a5a-4eb3-ac07-6fc0aabcd50c
  - aab66753-4b02-4162-a9d3-47374dabcdc4
```



```python
def safe_hello_world():
   return "Hello, Safe World!"

with Executor(endpoint_id="...") as gce:
   function_id = gce.register_function(safe_hello_world)
   fut = gce.submit_to_registered_function(function_id=function_id)
   res = fut.result()
```

Restricting access to endpoints

Cloud-enforced: Authentication policies

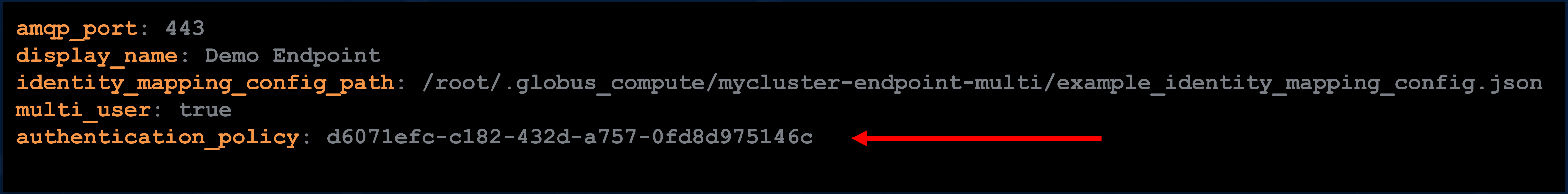
- Cloud gate-keeps submission to the endpoint
- E.g., domain restrictions, high assurance policies

Endpoint-enforced: Identity Mappings

- Map user identities to local accounts

Multi-user: Authentication Policies

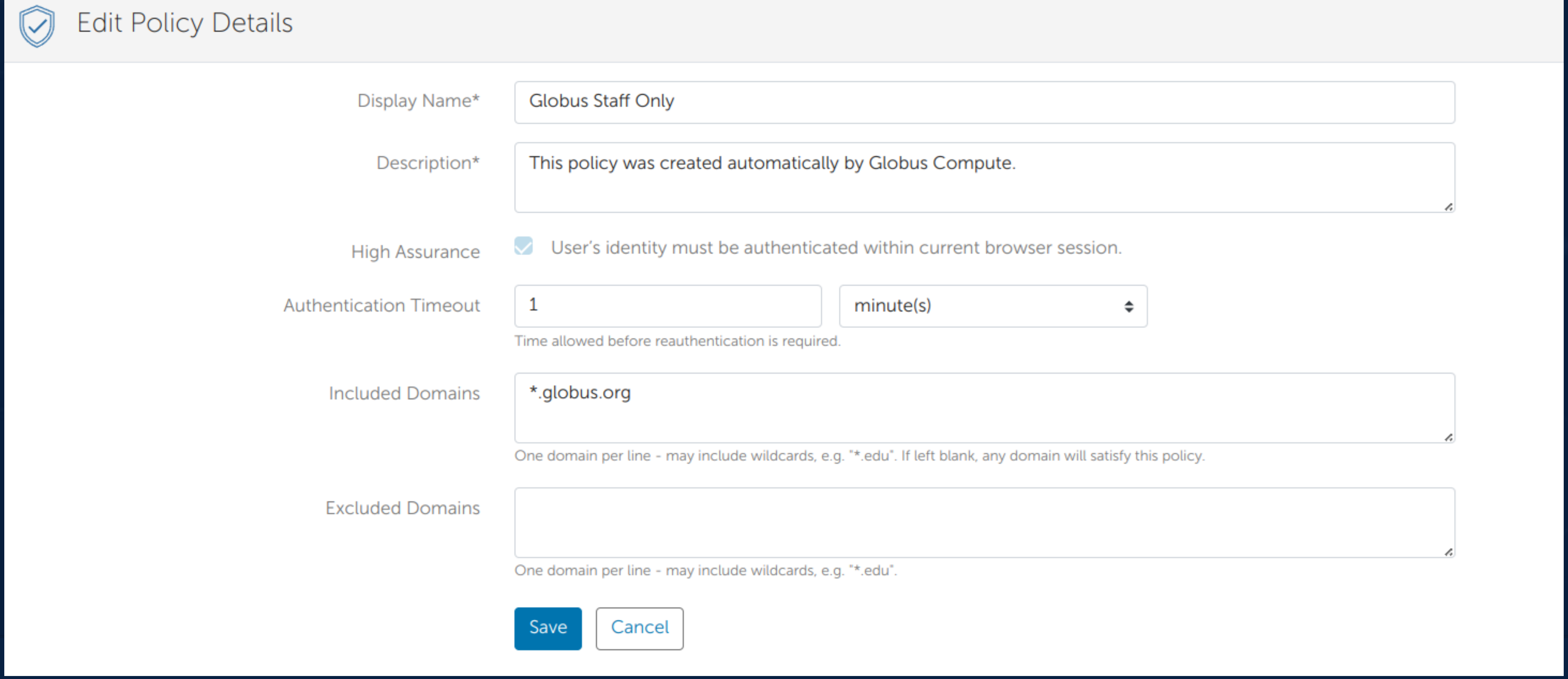
/root/.globus_compute/mycluster-endpoint-multi/config.yaml



```
amqp_port: 443
display_name: Demo Endpoint
identity_mapping_config_path: /root/.globus_compute/mycluster-endpoint-multi/example_identity_mapping_config.json
multi_user: true
authentication_policy: d6071efc-c182-432d-a757-0fd8d975146c
```

Multi-user: Authentication Policies

```
globus-compute-endpoint configure mycluster-endpoint-mu \
    --auth-policy-project-id 8236ad07-2801-468a-b262-9f1814988cc5 \
    --auth-policy-display-name "Globus Staff Only" \
    --allowed-domains "*.globus.org"  \
    --auth-timeout 60 \
    --subscription-id 964be8f5-5f9b-11e4-b64e-12313940394d \
    --multi-user
```



Edit Policy Details

Display Name* Globus Staff Only

Description* This policy was created automatically by Globus Compute.

High Assurance ☑ User's identity must be authenticated within current browser session.

Authentication Timeout 1 minute(s)

Time allowed before reauthentication is required.

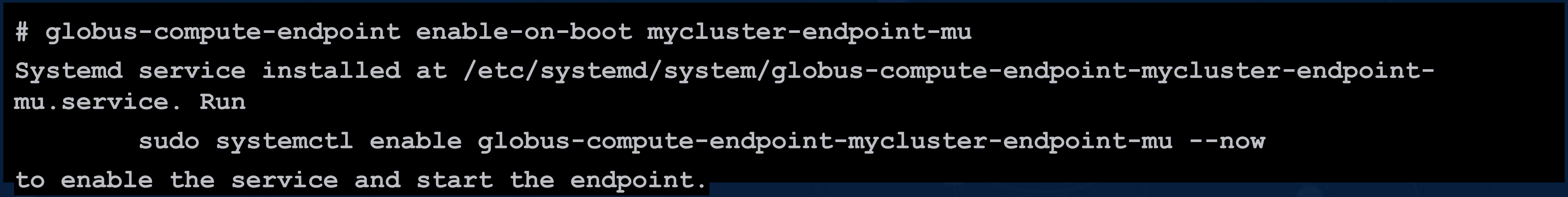
Included Domains *.globus.org

One domain per line - may include wildcards, e.g. "*.edu". If left blank, any domain will satisfy this policy.

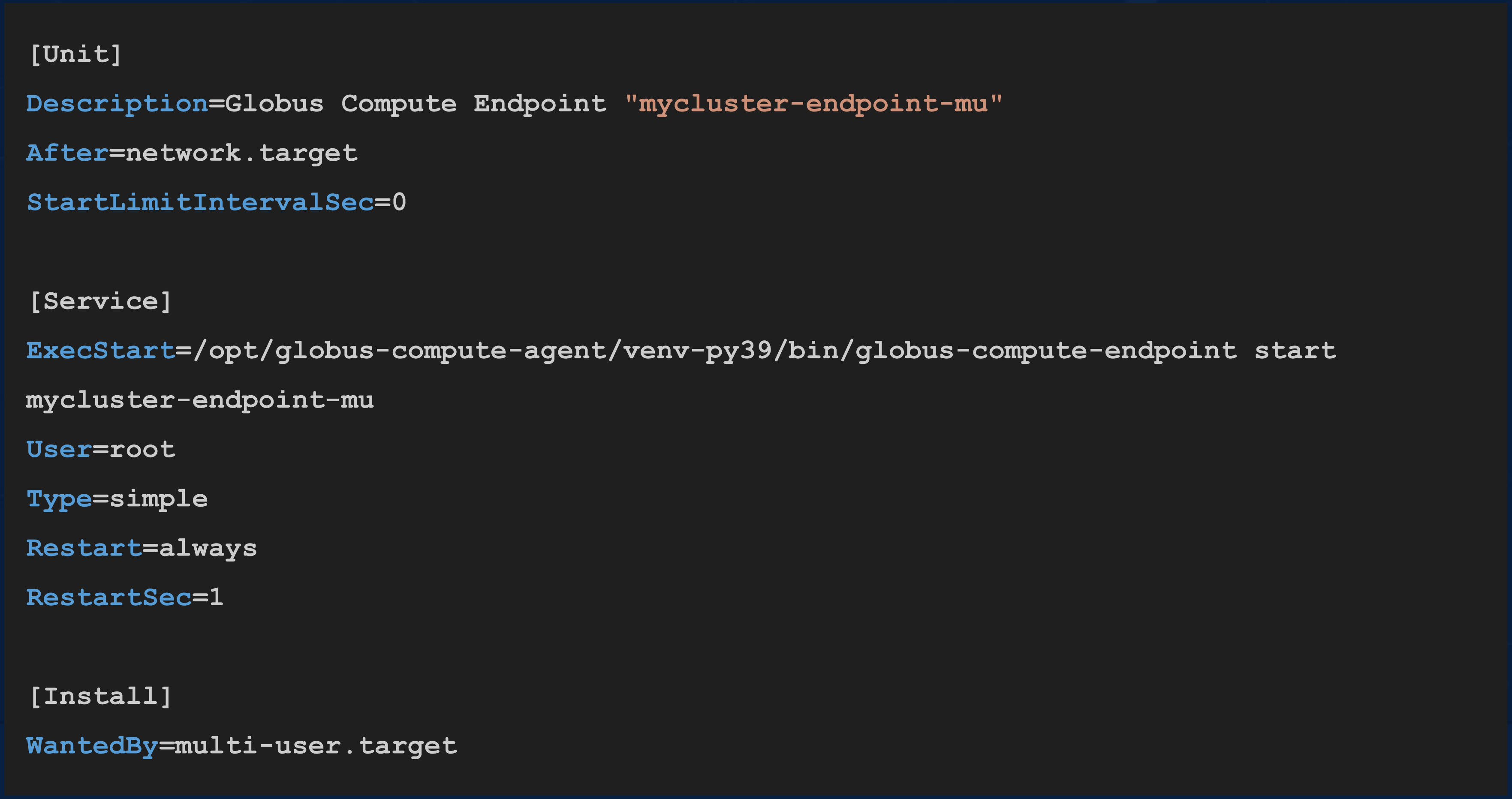
Excluded Domains

One domain per line - may include wildcards, e.g. "*.edu".

Multi-user: Enable on boot



```
# globus-compute-endpoint enable-on-boot mycluster-endpoint-mu
Systemd service installed at /etc/systemd/system/globus-compute-endpoint-mycluster-endpoint-
mu.service. Run
        sudo systemctl enable globus-compute-endpoint-mycluster-endpoint-mu --now
to enable the service and start the endpoint.
```



```
[Unit]
Description=Globus Compute Endpoint "mycluster-endpoint-mu"
After=network.target
StartLimitIntervalSec=0

[Service]
ExecStart=/opt/globus-compute-agent/venv-py39/bin/globus-compute-endpoint start
mycluster-endpoint-mu
User=root
Type=simple
Restart=always
RestartSec=1

[Install]
WantedBy=multi-user.target
```

Multi-user Endpoints: Requirements

- Ports
  - 443 outbound (optionally 5671) for both endpoint and SDK
  - No inbound ports
- Memory
  - ~200MB / active user
- Access to scheduler and shared filesystem

Compute High Assurance Features

- Additional authentication assurance
  - Authenticate with specific identity within session
- Isolation of applications
  - Authentication context per application, per session
- Enforced encryption of data in transit
- Local audit logging
- Option to require MFA
- Operations follow HIPAA, NIST SP 800-171, NIST SP 800-53 standards

Configuring HA Compute Endpoint

```
# globus-compute-endpoint configure \
         --multi-user \
         --auth-policy-project-id 81954df4-17af-4ef5-83e7-6d1ad251af4a \
         --display-name “My Cluster Compute Endpoint - HA" \
         --allowed-domains “example.edu" \
         --auth-timeout 36000 \
         --subscription-id AAAAAAAA-BBBB-CCCC-DDDD-EEEEEEEEEEEE \  # High Assurance Subscription
         --high-assurance \
         gw2025-demo-compute-endpoint-ha
```

```
authentication_policy: f47b946b-dd4b-4a2b-bfd1-984c374d67b7
display_name: My Cluster Compute Endpoint - HA
high_assurance: true
identity_mapping_config_path: /root/.globus_compute/mycluster-endpoint-
ha/example_identity_mapping_config.json
multi_user: true
subscription_id: AAAAAAAA-BBBB-CCCC-DDDD-EEEEEEEEEEEE
audit_log_path: /var/log/compute.log
```

Registering an HA Compute Function

- **HA functions must be registered with an HA endpoint**
- **HA functions will be deleted within 90 days of the last task submitted**

```
function_id = c.register_function(
    test_info,
    ha_endpoint_id=endpoint
)
```

Thank you, funders...

U.S. DEPARTMENT OF
ENERGY

NSF

THE UNIVERSITY OF
CHICAGO

NIH

NIST
National Institute of
Standards and Technology
U.S. Department of Commerce

ALFRED P. SLOAN FOUNDATION
1934

Argonne
NATIONAL LABORATORY

amazon
web services

Questions?

- **Documentation**
  - Compute home: **docs.globus.org/compute/**
  - Endpoints: **globus-compute.readthedocs.io/en/latest/endpoints/index.html**
  - Compute SDK: **globus-compute.readthedocs.io/en/latest/sdk/index.html**
- **Notebooks: github.com/globus/globus-jupyter-notebooks**
- **Helpdesk: support@globus.org**