SCAILFIN

Scalable CyberInfrastructure for Artificial Intelligence and Likelihood Free Inference

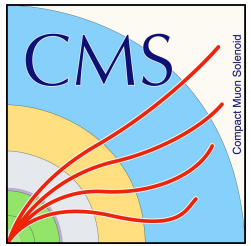
VC3, REANA, HPC-sites

Kenyi Hurtado: khurtado@nd.edu, Cody Kankel: ckankel@nd.edu

Motivation

Major Multi-User Research Facilities involve the comparison of data collected from experiments with “synthetic” data, produced from computationally-intensive simulations.

Comparisons of experimental data and predictions from simulations are abstractions of the specific data analysis techniques developed by the respective communities over several decades. E.g.:



Motivation

Many of these data analysis tasks are often conducted manually or through *ad hoc scripts* that might not be well maintained, difficulting reproducibility and reusability. Many of these tasks do have a well-defined workflow that make automation possible though.

REANA was created (in collaboration with DASPOS, DIANA and CERN) to address the reproducibility and reusability of the analysis pipeline.

Reproducible research data analysis platform

Motivation

In parallel:

Interest in leveraging Machine Learning (ML) and Artificial Intelligence (AI) techniques, to enhance the analysis of data from these facilities.

In particular, its application with emergent **Likelihood-Free Inference (LFI) techniques** when the predictions for the data are implicitly defined by the simulation, often leading to an intractable likelihood function. This can apply to analysis of data from LHC, LIGO, etc, but such Likelihood-Free algorithms have so far been **implemented mostly on individual machines and in ad hoc scripts.**

Motivation

SCAILFIN: Scalable CyberInfrastructure for Artificial Intelligence and Likelihood Free Inference

The SCAILFIN project aims to deploy artificial intelligence and likelihood-free inference techniques and software using scalable cyberinfrastructure (CI) that is developed to be integrated into existing CI elements, such as the REANA system.

The topic today:

SCAILFIN goals

- ...
- One of the goals of the project is **extending the REANA** platform to allow remote **submission** of workflows to **HPC facilities.**
- ...

Background

As mentioned before, SCAILFIN plans to use:

- **REANA** as the Cyber Infrastructure element to deploy AI and Likelihood-Free inference techniques.

- We are also leveraging on **VC3** (Virtual Clusters for Community Computation) in order to scale REANA to HPC resources.

Let's see a bit about these 2 components...

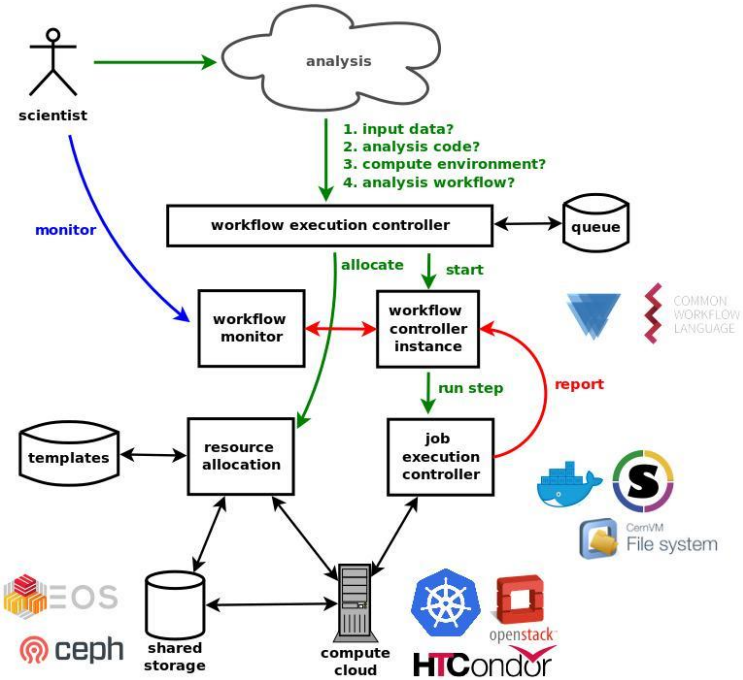
REANA: Reproducible Research Data Analysis Platform

Features

- Allows creation of tightly defined, container encapsulated workflows
- Whole purpose is to allow complete reproducibility
- Sharing workflows is as easy as sharing a specification
  - (and inputs!)
- Different workflow engines supported. E.g.:
  - CWL (Common Workflow Language) : https://www.commonwl.org/
  - Yadage (YAML based adage): https://yadage.readthedocs.io

REANA: Reproducible Research Data Analysis Platform

1. Create workflow specification
   (Yadage, CWL, Serial)
2. Upload workflow and inputs to REANA cloud
3. Start workflow
4. Download / pull down results
5. Share workflow specs with others

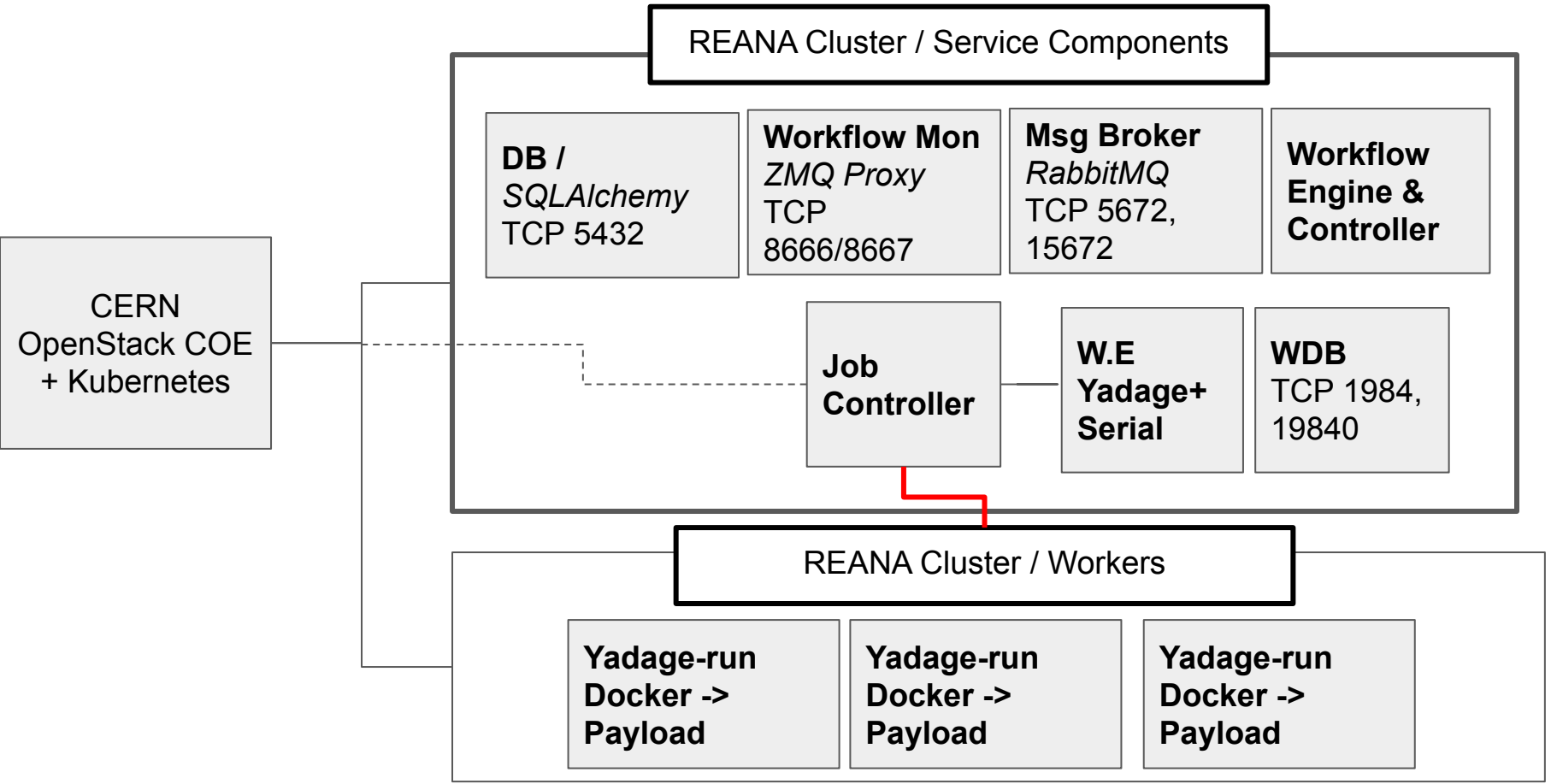


REANA: Reproducible Research Data Analysis Platform

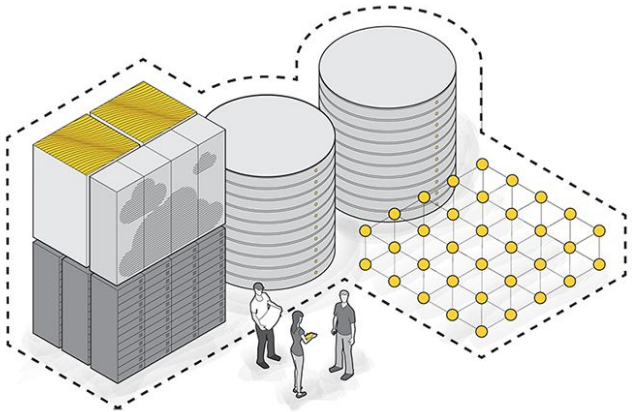
Components

- Two major components each consisting of many sub-components.
  - reana-client: User facing component.
    - Accepts workflows and and is used as interface to entire REANA system (for user).
  - reana-cluster: Workhorse.
    - Consists of many small pieces which handle workflows, dish out jobs, coordinates results, can be thought of as the job scheduler. Jobs are scheduled via Kubernetes.

reana-cluster - Simplified Diagram



VC3: Virtual Clusters for Community Computation



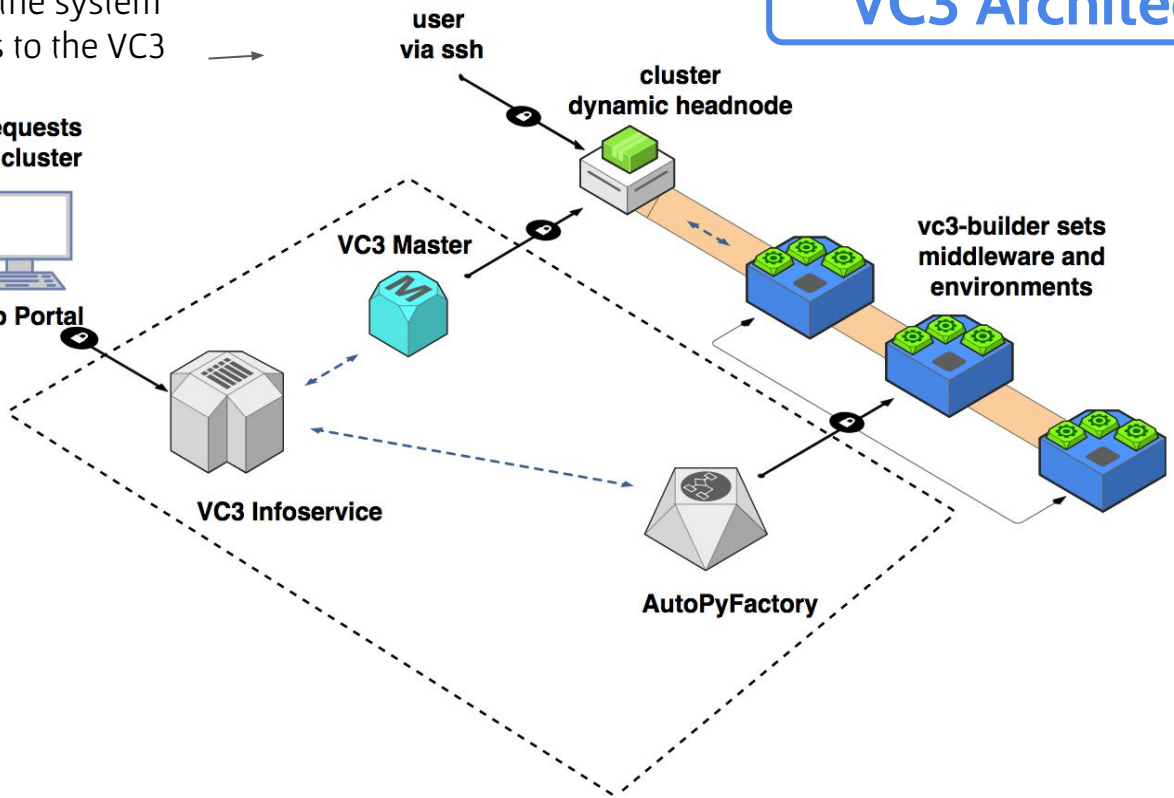
VC3: A platform for provisioning cluster frameworks over heterogeneous resources for collaborative science teams

https://www.virtualclusters.org

VC3: Virtual Clusters for Community Computation

User adds an SSH public key to the system that will be used to grant access to the VC3 dynamic headnode

VC3 Architecture



User defines an allocation. Two authentication mechanisms are supported:

- SSH keys: the user is handed a public ssh key to add to his/her submit node resource.
- GSI-SSH proxies: The user provides a proxy that VC3 can use.

VC3: Virtual Clusters for Community Computation

Features

- The user can select its own middleware for submission (E.g.: HTCondor, WorkQueue, Spark, REANA+HTCondor).
- It doesn't matter what the resource target batch system is (as long as it is supported by glite/blah, our translation layer for submission).

E.g.: Torque (Bluewaters), SLURM (NERSC, PSC–Bridges, Stampede2), HTCondor, LSF, SGE, PBS.

VC3: Virtual Clusters for Community Computation

Constraints

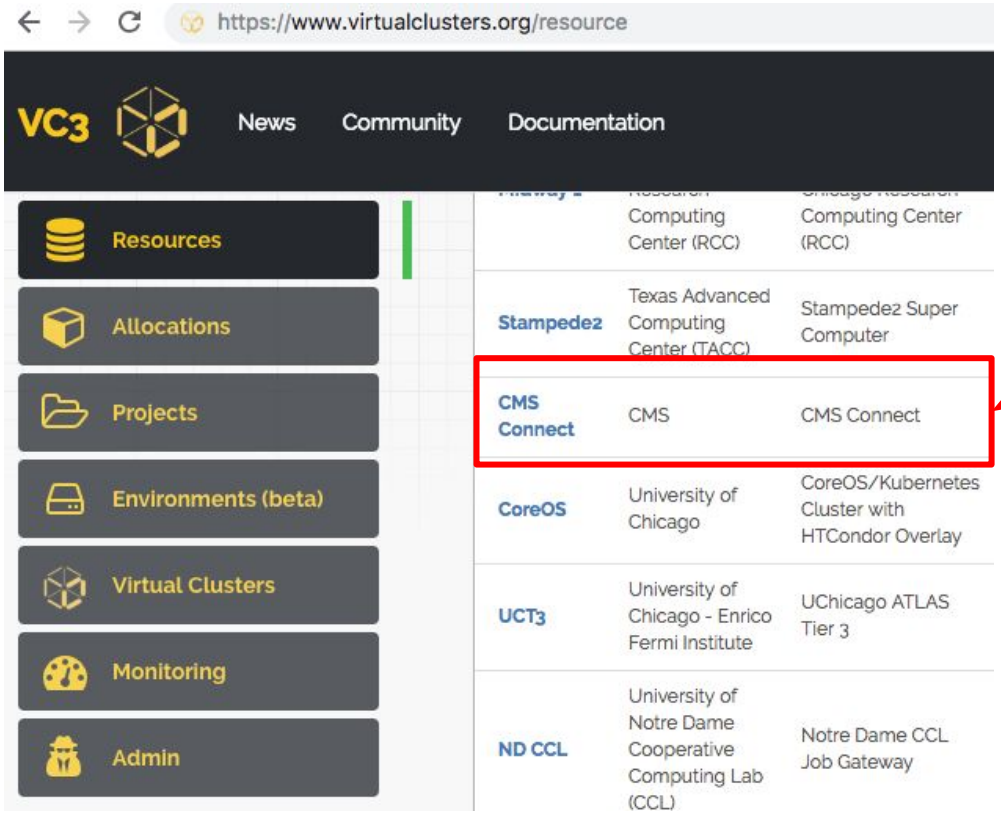
- At present, workers need outgoing network connection for a virtual cluster to work.
  - So, resources like ALCF/Theta are out of the scope with this approach.
  - But e.g.: XSEDE resources like NERSC, Bluewaters, Stampede or PSC–Bridges do meet the outgoing network requirement for example.

VC3: Virtual Clusters for Community Computation

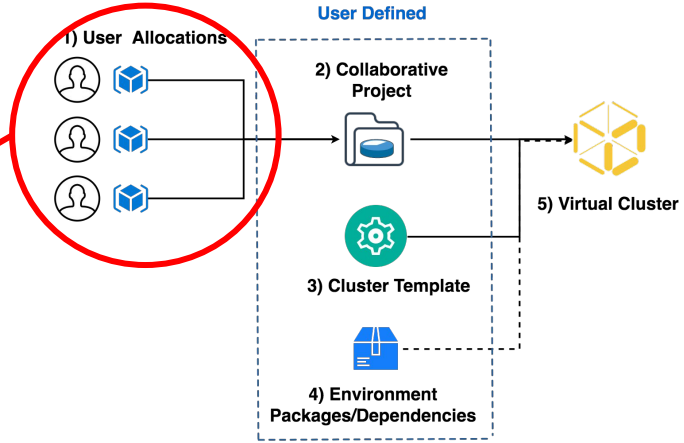
Use-case example

Let's say we have an allocation to an HPC resource, or a pilot-based HTCondor pool like OSG Connect / CMS Connect / ATLAS Connect.
We want to use our allocation in this resource in order to create a SPARK cluster.

Creating a Spark Cluster – Step 1

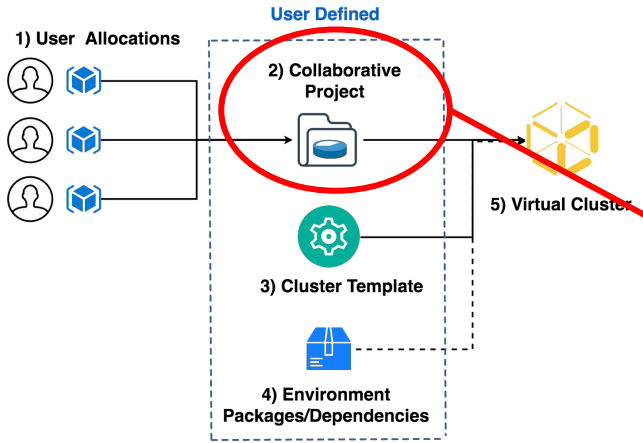


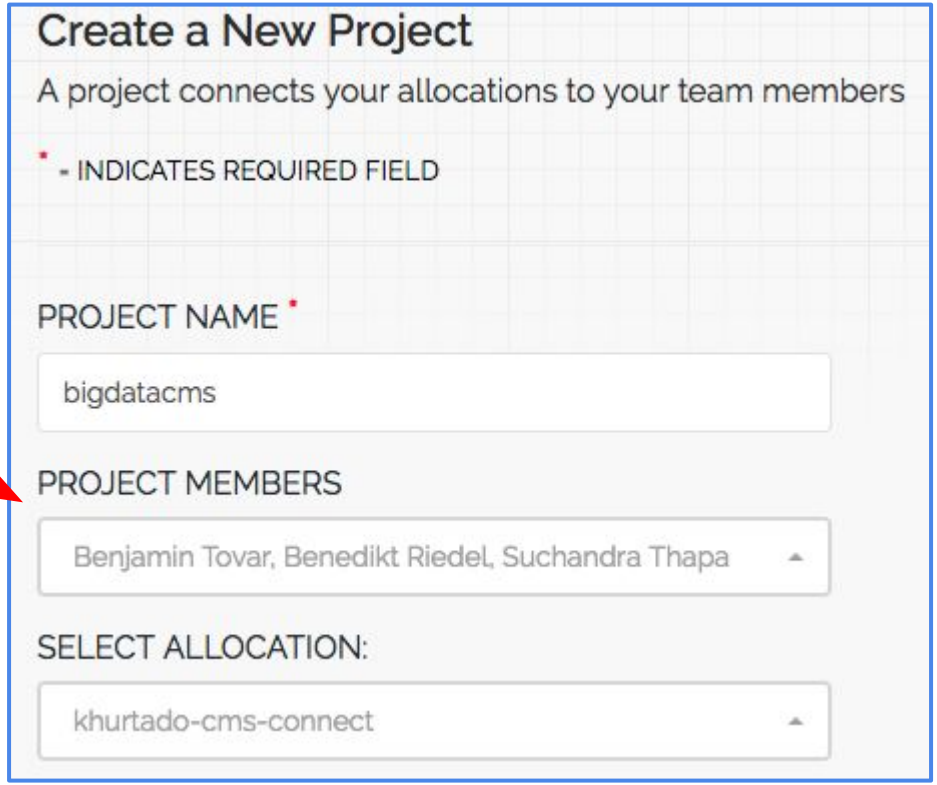
VC3 website:
https://www.virtualclusters.org/



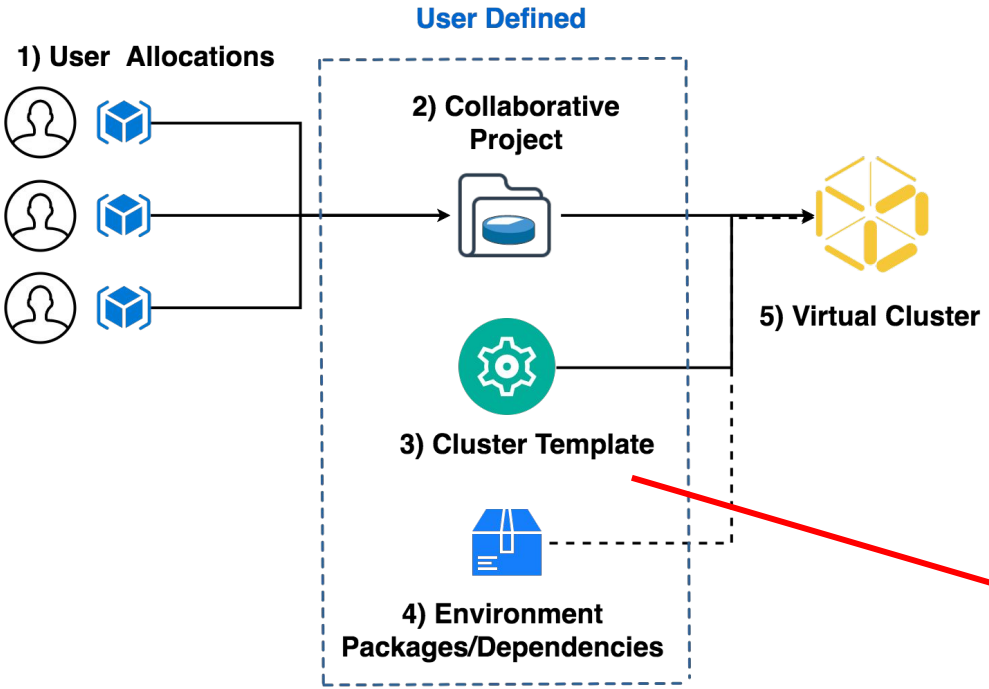
Using CMS Connect to access the CMS Global Pool

Creating a Spark Cluster – Step 2

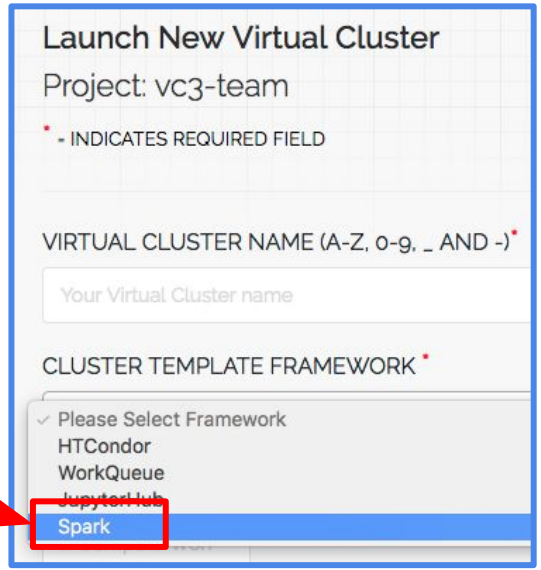




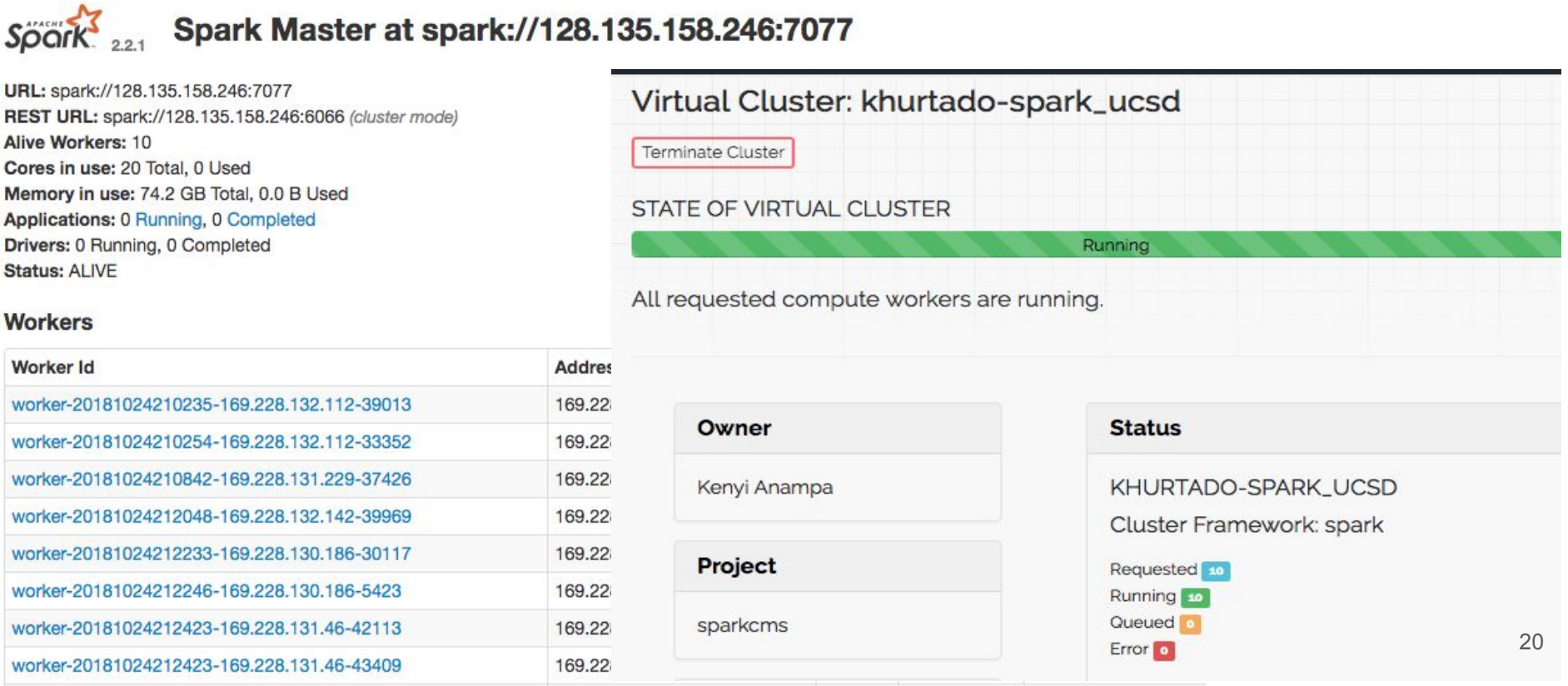
Creating a Spark Cluster – Step 3



Use CMS Connect to access the CMS Global Pool



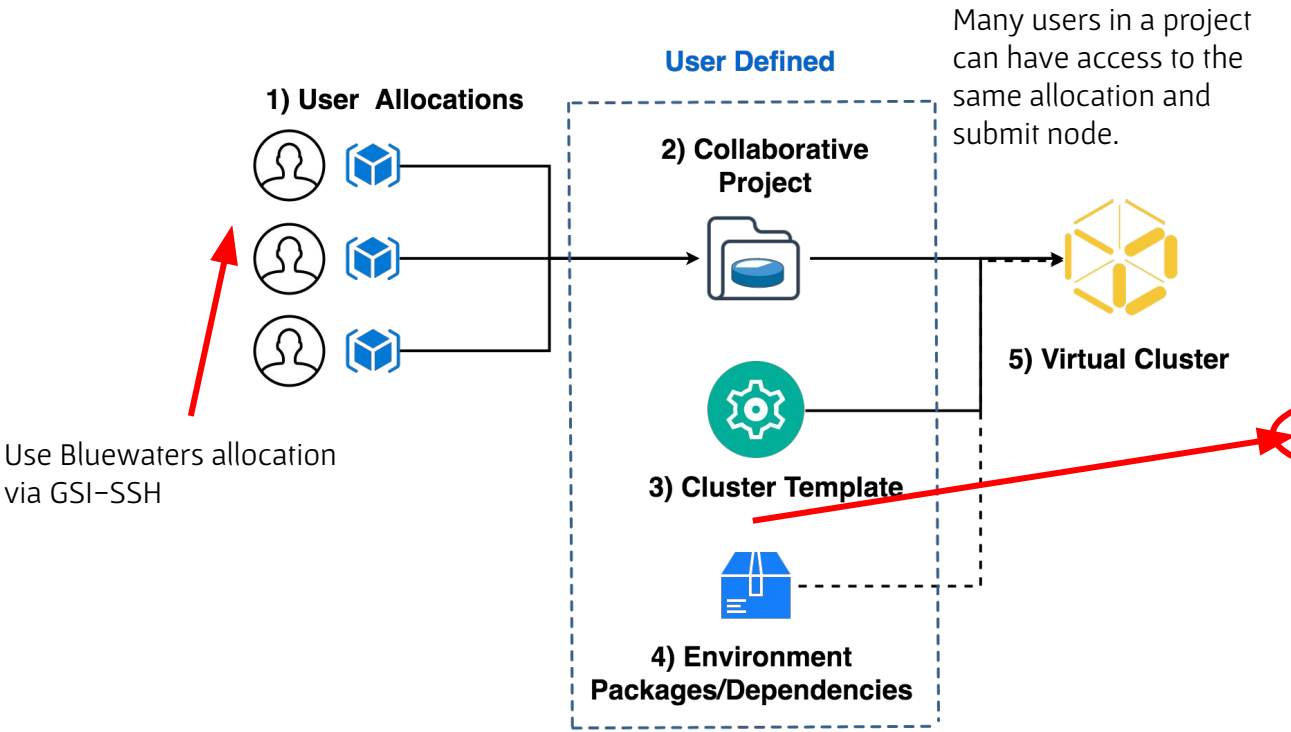
Creating a Spark Cluster – Step 4



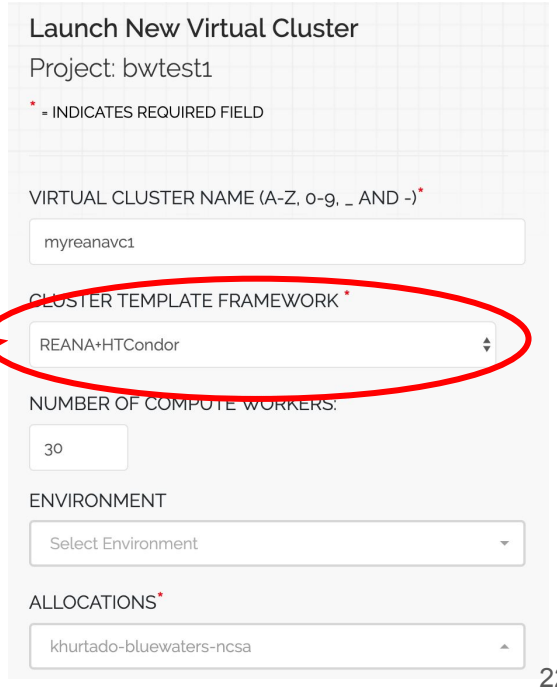
Putting it all together

REANA + VC3 with HTCondor on Blue Waters (in progress)

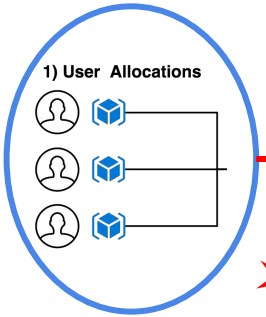
Creating a REANA Cluster Template for VC3

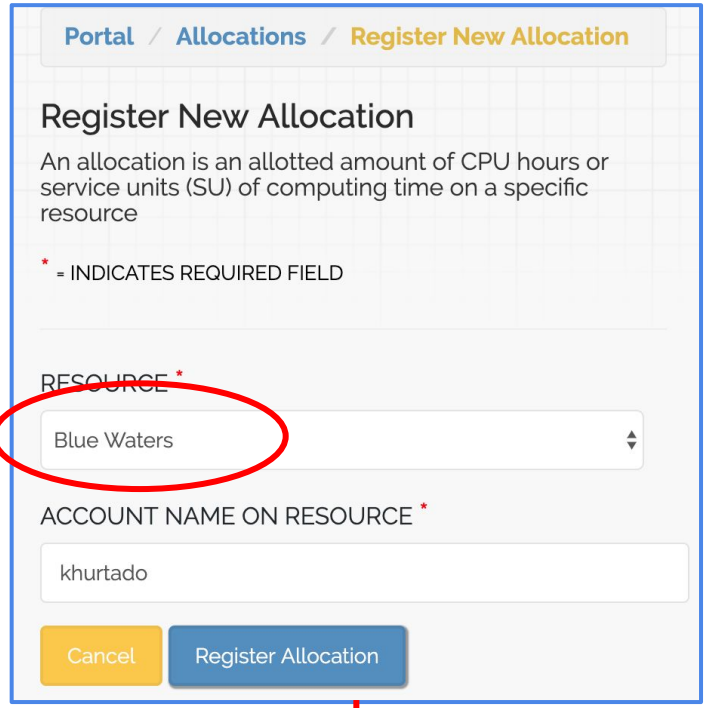


Many users in a project can have access to the same allocation and submit node.

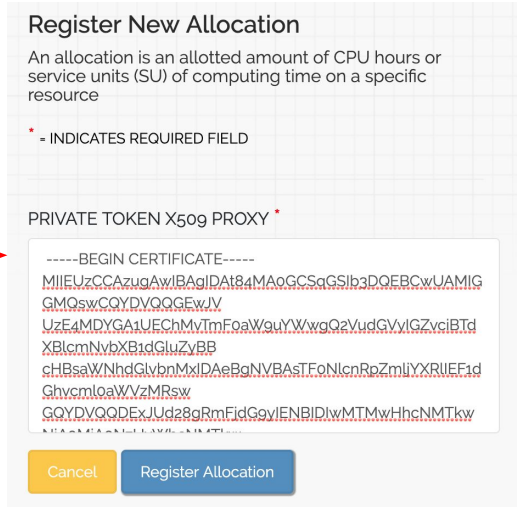


Creating A REANA cluster on Blue Waters





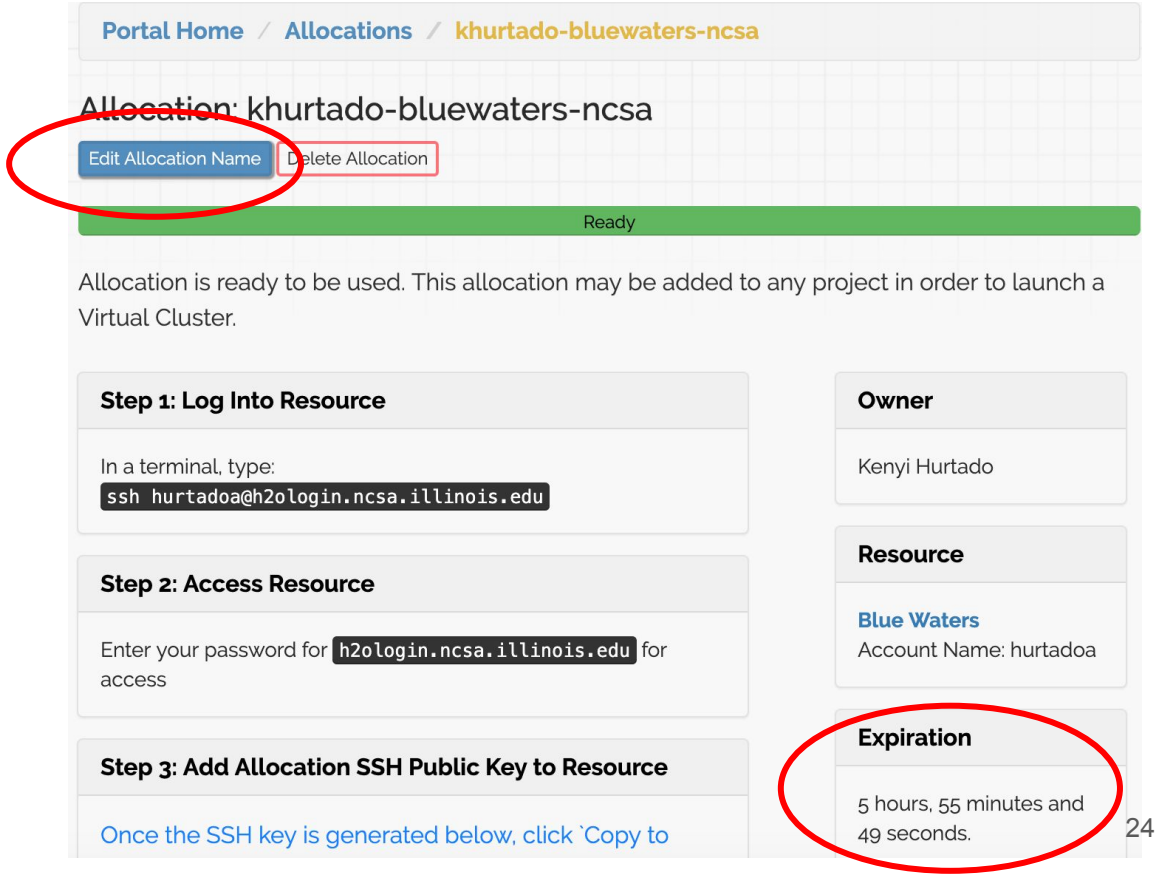

```
hurtadoa@h2ologin1:~> myproxy-logon -s
tfca.ncsa.illinois.edu -p 7512
Enter MyProxy pass phrase:
A credential has been received for user hurtadoa in
/tmp/x509up_u57980.
hurtadoa@h2ologin1:~> cat /tmp/x509up_u57980
```



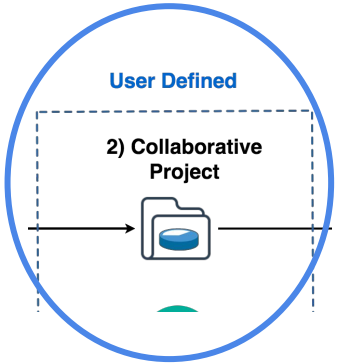
Allocation authentication mechanisms

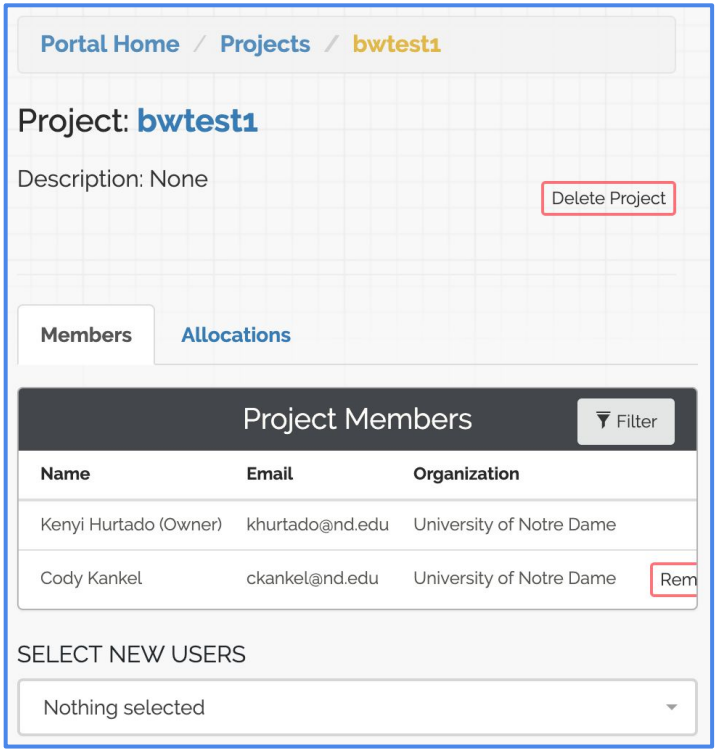
The auth mechanisms available in VC3 are:

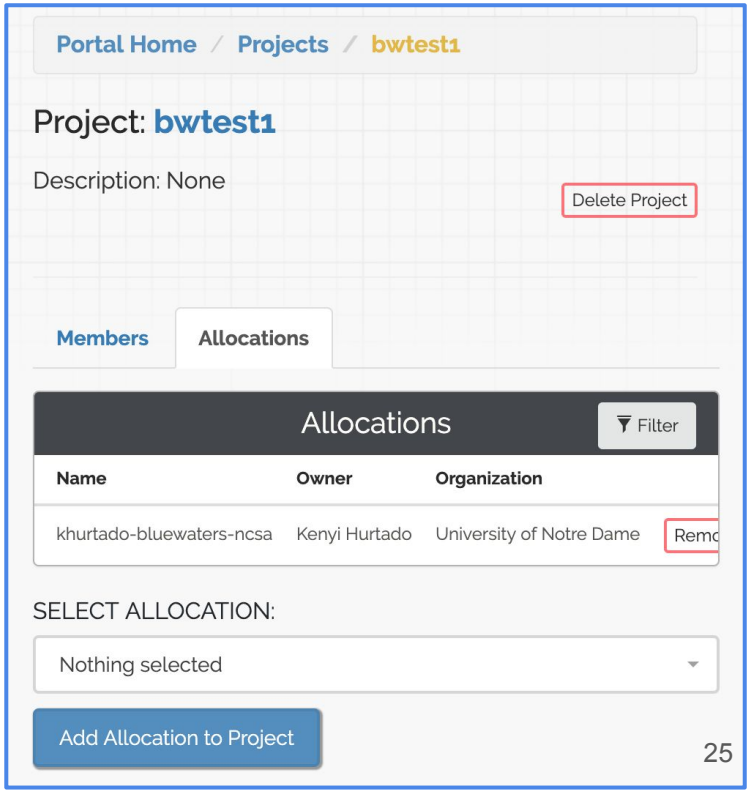
- SSH keys
- GSI tokens
  - Available on XSEDE HPC Systems
  - Requires renewal. The website shows when the proxy expires and the "Edit Allocation Name" button allows users to change
  - Some HPC centers have procedures to increase the default 12 hours expiration time to e.g.: 10 days (NERSC case, and previously done at BW)



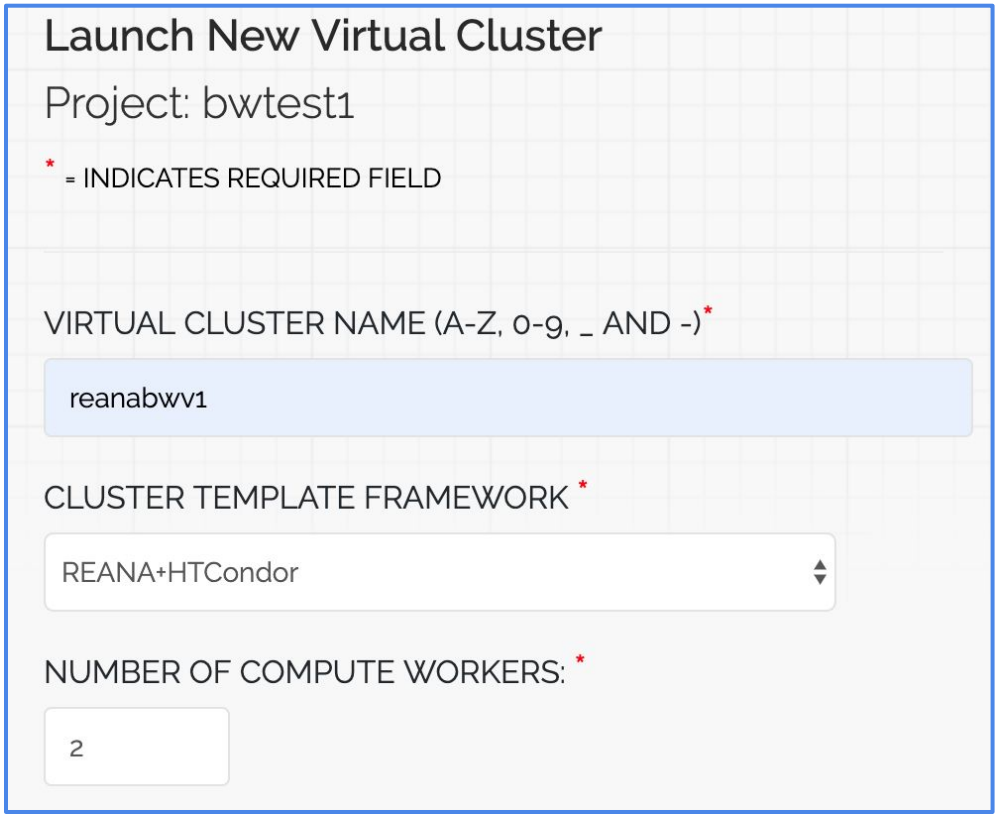
Creating A REANA cluster on Blue Waters

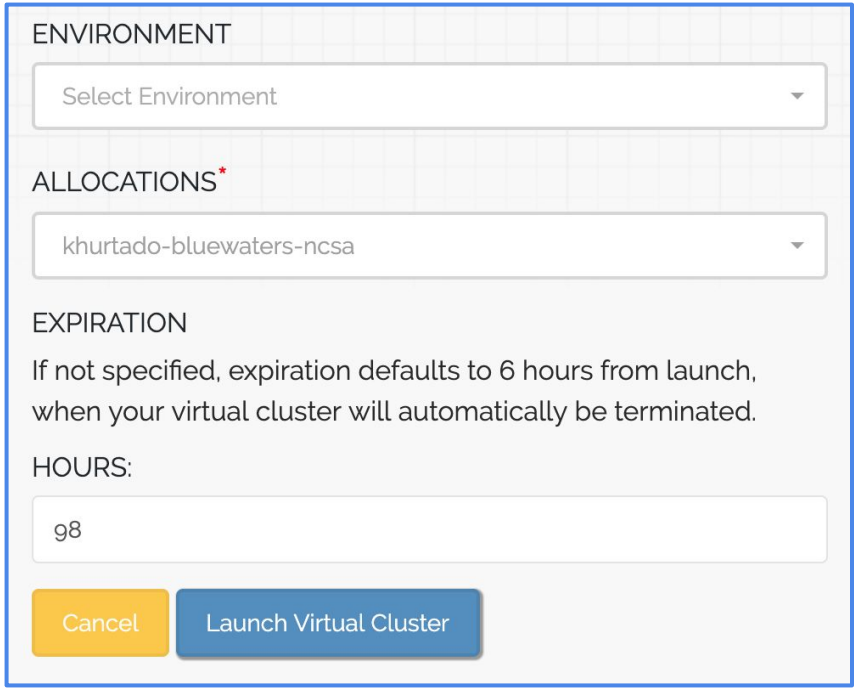




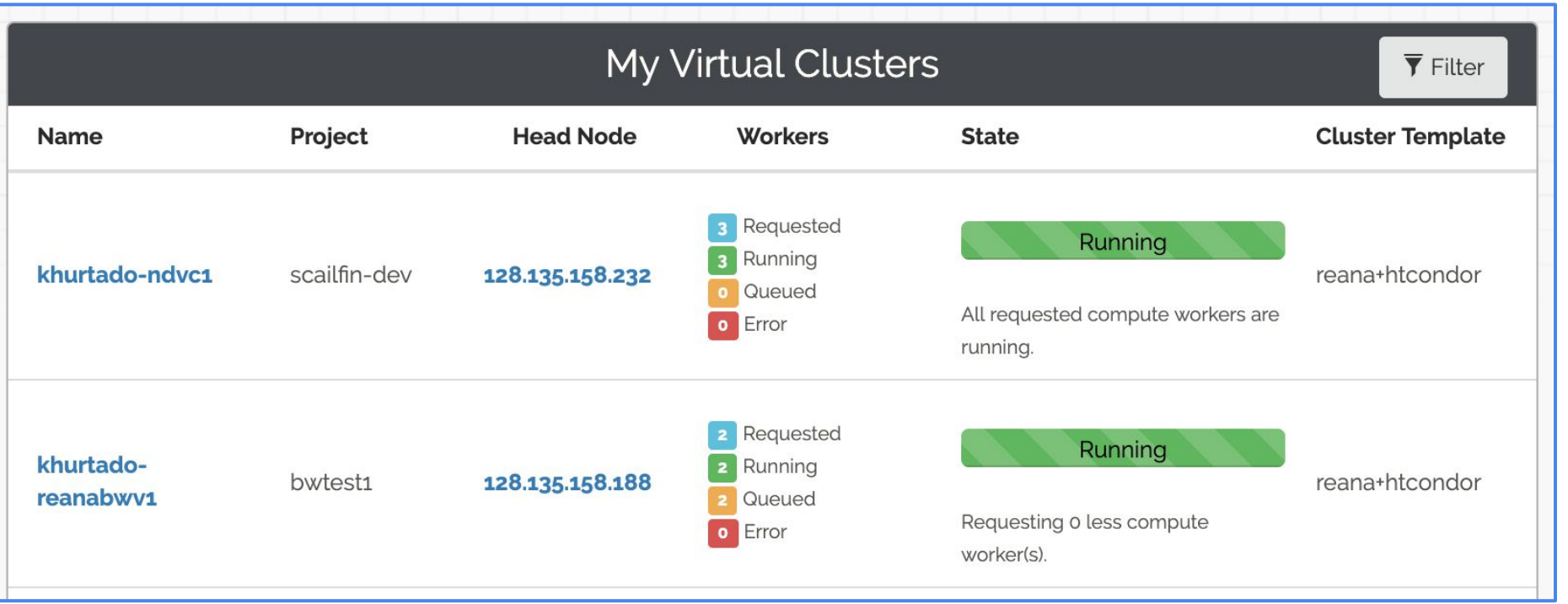


Creating A REANA cluster on Blue Waters





Creating A REANA cluster on Blue Waters



My Virtual Clusters

Filter

Name	Project	Head Node	Workers	State	Cluster Template
khurtado-ndvc1	scailfin-dev	128.135.158.232	3 Requested 3 Running 0 Queued 0 Error	Running All requested compute workers are running.	reana+htcondor
khurtado-reanabwv1	bwtest1	128.135.158.188	2 Requested 2 Running 2 Queued 0 Error	Running Requesting 0 less compute worker(s).	reana+htcondor

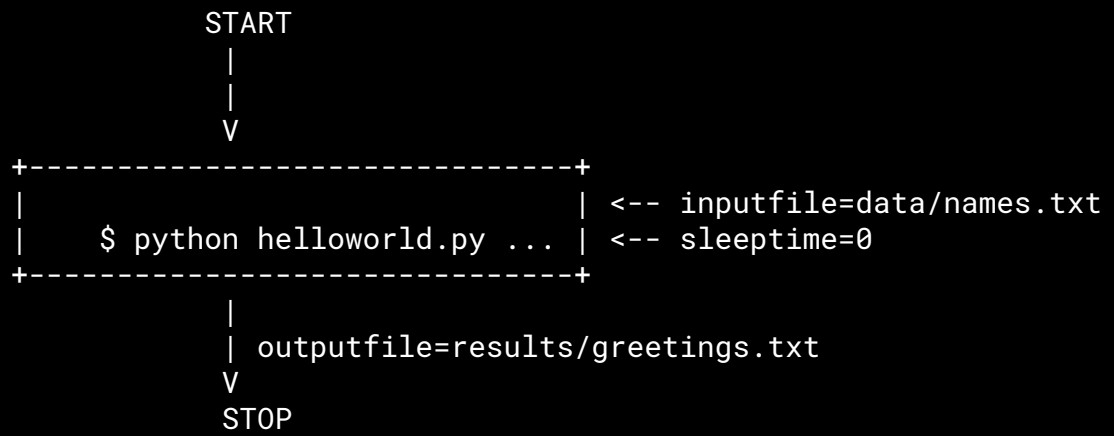
End result will be A REANA cluster deployed on the VC3 headnode

Components are deployed via Kubernetes (minikube)

```
(reana) [khurtado@khurtado-reanabwv1 ~]$ reana-cluster status
COMPONENT             STATUS
job-controller        Running
server                Running
db                    Running
workflow-controller   Running
message-broker        Running
REANA cluster is ready.
(reana) [khurtado@khurtado-reanabwv1 ~]$ kubectl get pods
NAME                                                  READY   STATUS      RESTARTS   AGE
batch-serial-7e79ee48-036f-4049-87ee-a3dc66d8a1da-tl7zd   0/1     Completed   0          5h54m
db-69744557df-wg4mt                                   1/1     Running     0          5h55m
job-controller-5c7f4c8b4f-sgnj6                       1/1     Running     0          5h55m
message-broker-b7d66cf55-m9p4n                        1/1     Running     0          5h55m
server-58dc985c77-n2qpn                               2/2     Running     0          5h55m
workflow-controller-668f69d4bc-x62w7                  2/2     Running     0          5h55m
```

reana-client

Basic Example



1. Create workflow in reana-client
2. Upload code (helloworld.py) and data (names.txt)
3. Start workflow
    a. Executes Python code and creates a result directory with the names found in names.txt
4. Download results

reana-client

Basic Example

```
$ reana-client create -n my_stuff
my_stuff.1
$ export REANA_WORKON=my_stuff
$ reana-client upload ./code ./data
File code/helloworld.py was successfully uploaded.
File data/names.txt was successfully uploaded.

$ reana-client start
my_stuff is running
$ reana-client status
NAME       RUN_NUMBER   CREATED             STATUS     PROGRESS
my_stuff   1            2019-06-04T21:13:21   running    0/1
$ reana-client status
NAME       RUN_NUMBER   CREATED             STATUS     PROGRESS
my_stuff   1            2019-06-04T21:13:21   finished    1/1

$ reana-client ls

NAME                  SIZE   LAST-MODIFIED
code/helloworld.py    2991   2019-06-04T20:24:36
data/names.txt          20   2019-06-04T20:24:36
results/greetings.txt   67   2019-06-04T20:25:40
```

reana-client

Results from Blue Waters

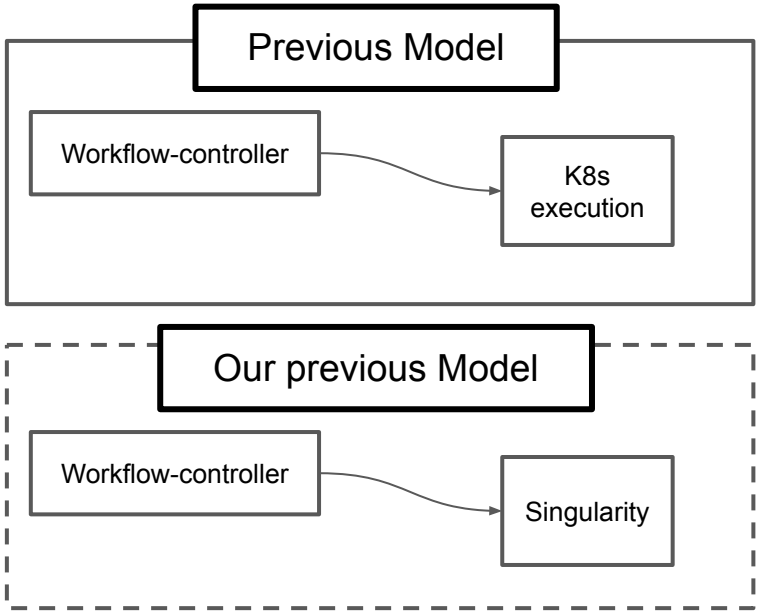
```
(reana)[ckankel@ckankel-bw_scailfin_test1 reana-demo-helloworld]$ reana-client ls
NAME                  SIZE   LAST-MODIFIED
code/helloworld.py    2991   2019-06-04T20:24:36
data/names.txt          20   2019-06-04T20:24:36
results/greetings.txt   67   2019-06-04T20:25:40

(reana)[ckankel@ckankel-bw_scailfin_test1 reana-demo-helloworld]$ reana-client download

File results/greetings.txt downloaded to /home/ckankel/reana-demo-helloworld.

(reana)[ckankel@ckankel-bw_scailfin_test1 reana-demo-helloworld]$ cat results/greetings.txt

Working from hostname: nid25354

Hello Jane Doe!

Hello Joe Bloggs!
```

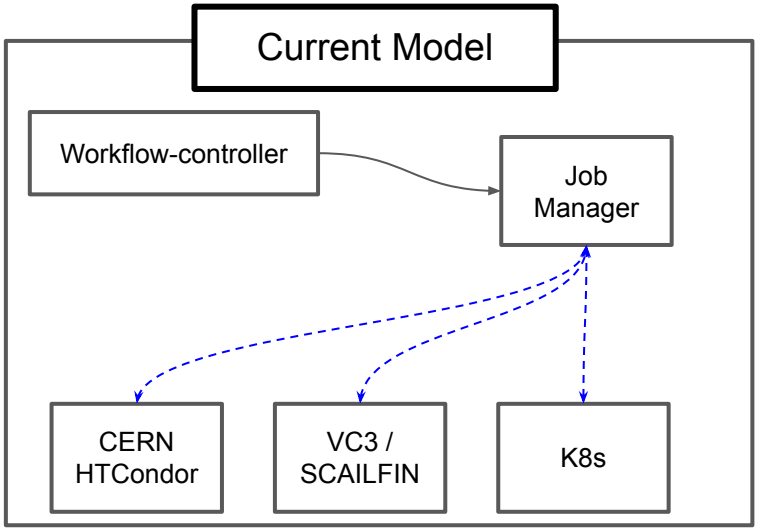
SCAILFIN's REANA modifications

- REANA requires some form of docker supporting container technology
  - Singularity and Shifter support in the works. Possibly CharlieCloud
- REANA expects to submit to a kubernetes cluster
  - Added support for VC3 specialized HTCondor submissions through a modified reana-job-controller and a job_wrapper for every workflow step.
  - The modified reana-job-controller submits each workflow step to a local condor scheduler
- Job wrapper
  - Each workflow step is wrapped by a script which searches for container technology and launches each workflow step into the available container (shifter, singularity)

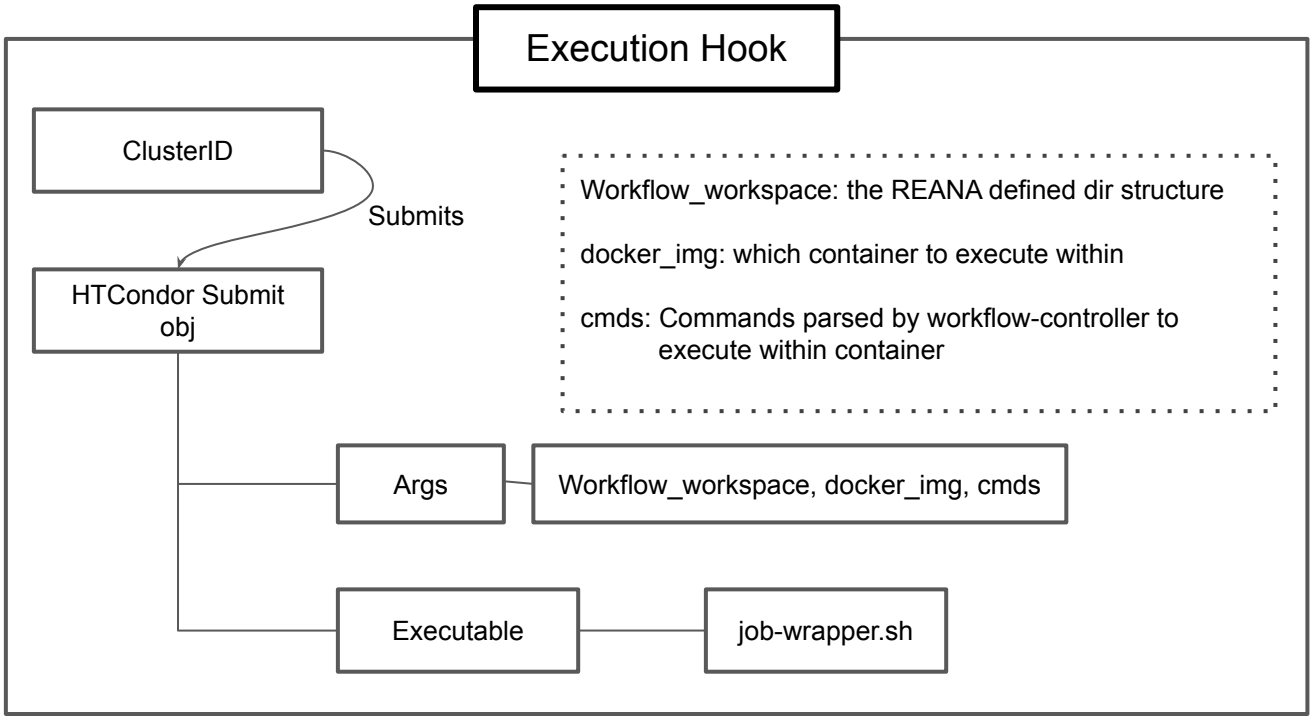
SCAILFIN's REANA modifications

- REANA developers allowed abstraction of reana-job-controller's backend
  - Allows for "plug and play" of backends / job execution component



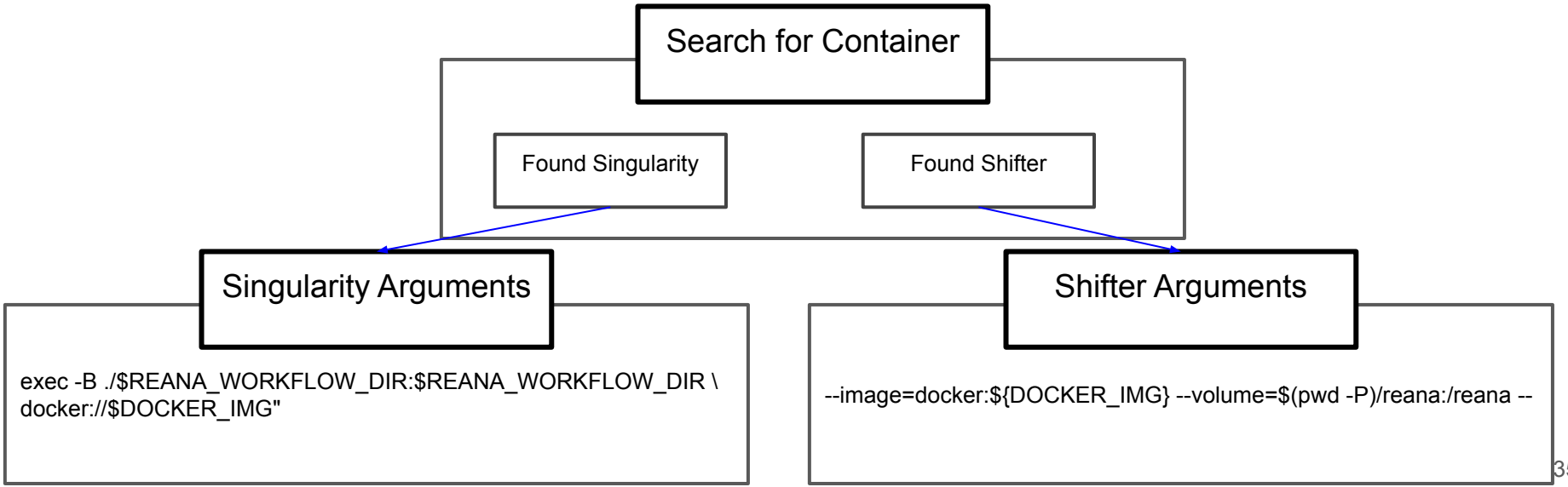


SCAILFIN's REANA modifications



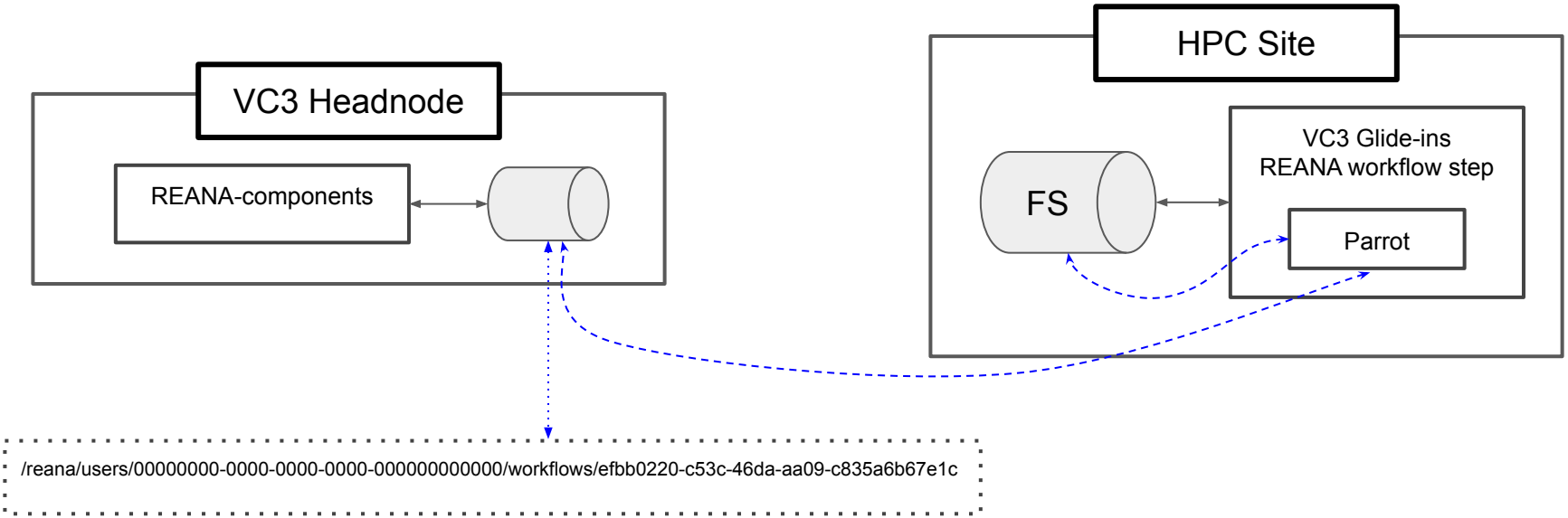
Job-wrapper Container management

- Performs check for available container technologies
  - Checks Binaries in $PATH (VC3 may auto-load these)
  - Attempts to load modules for Singularity and Shifter
  - Executes workflow-step within discovered container
    - Will choose depending on the currently set $default



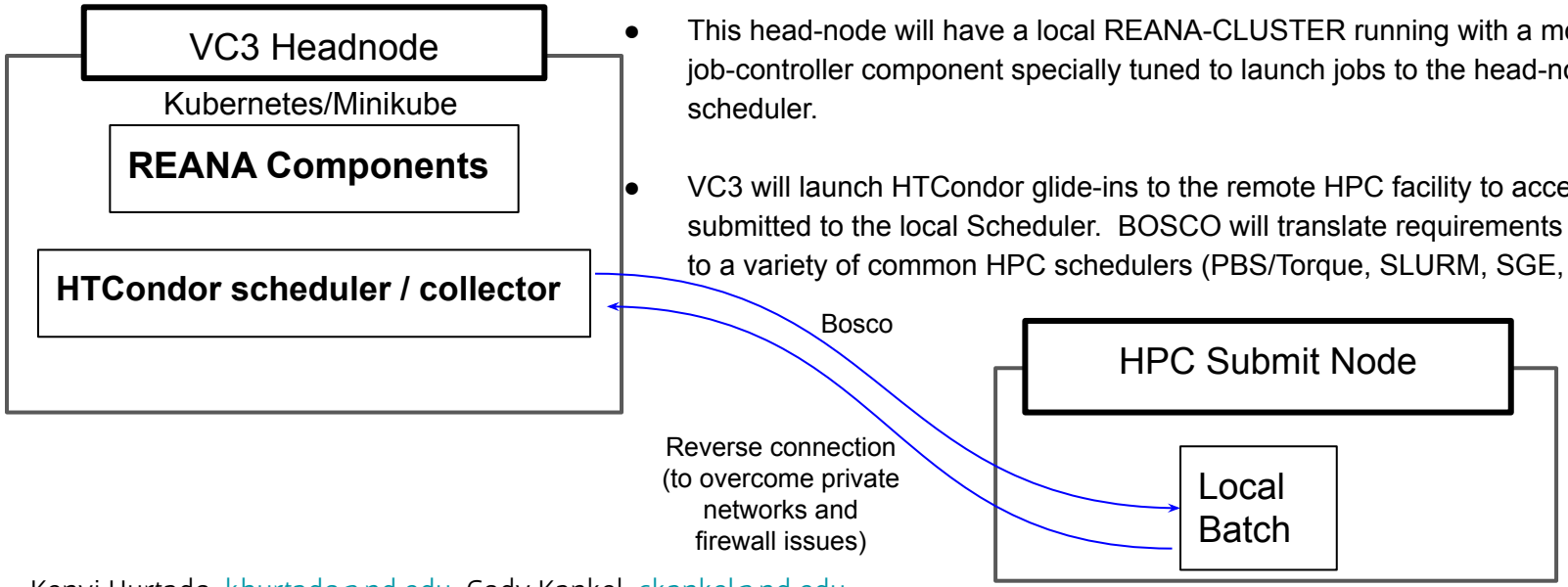
Job-wrapper File management

- REANA expects files to be where the workflow controller placed them
  - We must simulate this directory structure on the HPC infrastructure
- We can easily transfer entire directory structures with HTCondor's chirp protocol and Parrot



SCAILFIN and VC3

VC3 is utilized for remote connections to clusters.



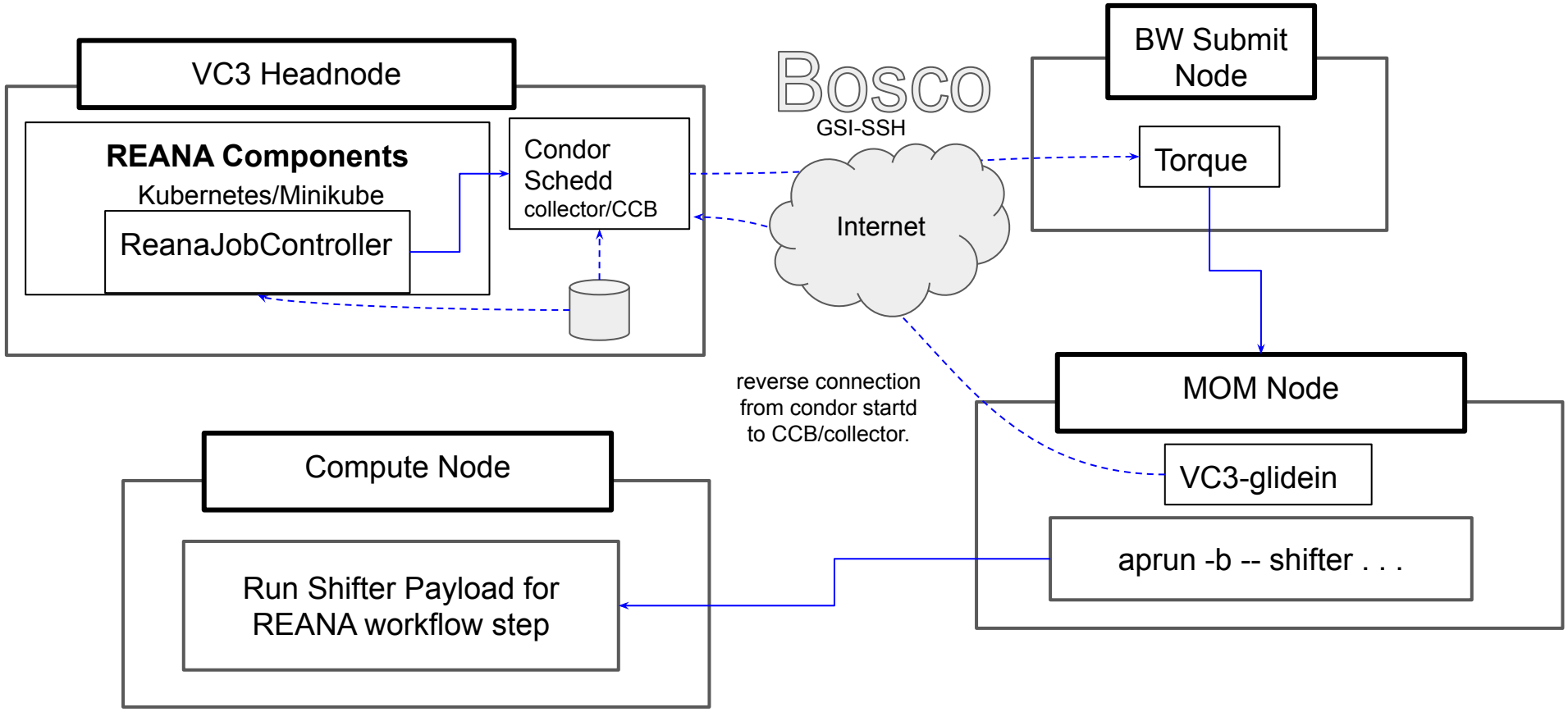
- This head-node will have a local REANA-CLUSTER running with a modified job-controller component specially tuned to launch jobs to the head-node's HTCondor scheduler.
- VC3 will launch HTCondor glide-ins to the remote HPC facility to accept jobs submitted to the local Scheduler. BOSCO will translate requirements from HTCondor to a variety of common HPC schedulers (PBS/Torque, SLURM, SGE, etc.)

Kenyi Hurtado: khurtado@nd.edu, Cody Kankel: ckankel@nd.edu

SCAILFIN's VC3 modifications

- Cluster template for REANA+HTCondor
  - Uses the standard HTCondor template as the base to create a condor pool that sends jobs to HPC resources, translating the job to the corresponding batch system submission syntax via bosco.
  - Deploys Kubernetes via minikube
  - Deploys the REANA cluster and client and set up the environment, so the user can interact with them as soon as the VC3 headnode is created.
- GSI-SSH support
  - The GSI-SSH authentication mechanism was added in the infrastructure, in order to support e.g.P XSEDE HPC centers like Blue Waters, Stampede, NERSC.
  - Proxies can be renewed through the VC3 website for the virtual cluster allocations to remain active.
- Additional patches to support Cray Linux environments (NERSC, Blue Waters)

SCAILFIN on Blue Waters



Kenyi Hurtado: khurtado@nd.edu, Cody Kankel: ckankel@nd.edu

Links

- SCAILFIN Source code:
  - SCAILFIN's modified RJC
    https://github.com/scailfin/reana-job-controller/tree/job_manager
  - REANA
    - https://github.com/reanahub
  - VC3
    - https://github.com/vc3-project
- Websites
  - https://www.virtualclusters.org
  - http://www.reanahub.io/

Contacts

- Kenyi Hurtado
  - khurtado@nd.edu
- Cody Kankel
  - ckankel@nd.edu