VC3 Documentations Documentation

Release

Jeremy Van

Nov 28, 2017

Contents:

CHAPTER 1

Quickstart

CHAPTER 2

VC3 User Guide

All users of VC3 should read this portion of the documentation. This provides examples and documentation around VC3's assortment of options and how to specify them on the portal.

2.1 Getting Started with VC3

Last revised: Tuesday, November 26, 2017

questions/comments: lincolnb@uchicago.edu

|Virtual Clusters for Community Computation|, or **|VC3|**, is a platform for connecting clusters, grids, and clouds. VC3 can run overlay systems for a variety of cluster frameworks to make disparate resources appear as a single "virtual" resource for collaborative science.

Note: This portion of **|VC3|**'s documentation does not cover installation. See the *Installation* section for how to install **|VC3|**.

Prerequisites In order to use VC3, you'll need an allocation or account with with a supported target resource. These include, but are not limited to:

- University of Chicago - Research Computing Center
- University of Notre Dame - Center for Research Computing
- Brookhaven National Laboratory - Scientific Data & Computing Center
- Syracuse University - Research Computing
- Texas Advanced Computing Center
- NERSC
- Amazon Web Services
- Open Science Grid

- *and more!*

Institutions and resources are added frequently - be sure to subscribe to our newsletter and visit https://www.virtualclusters.org!

2.1.1 Building your first Virtual Cluster

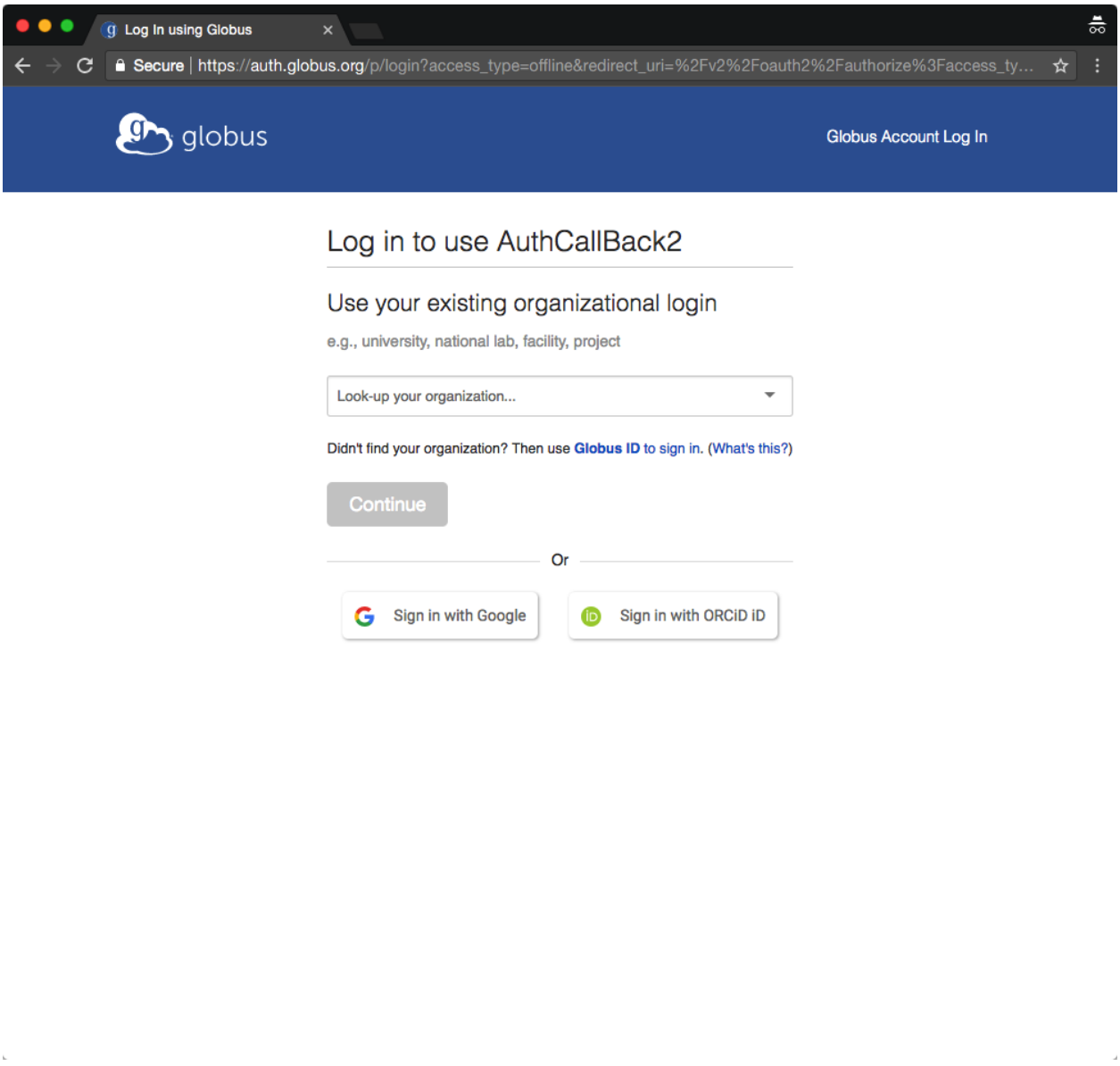
Login or Create Account

When you first visit https://www-dev.virtualclusters.org, you'll be presented with a Login link in the top right of the screen. Click "Login" - this will take you to a Globus (https://globus.org) sign-in site.

2. Sign in to Globus

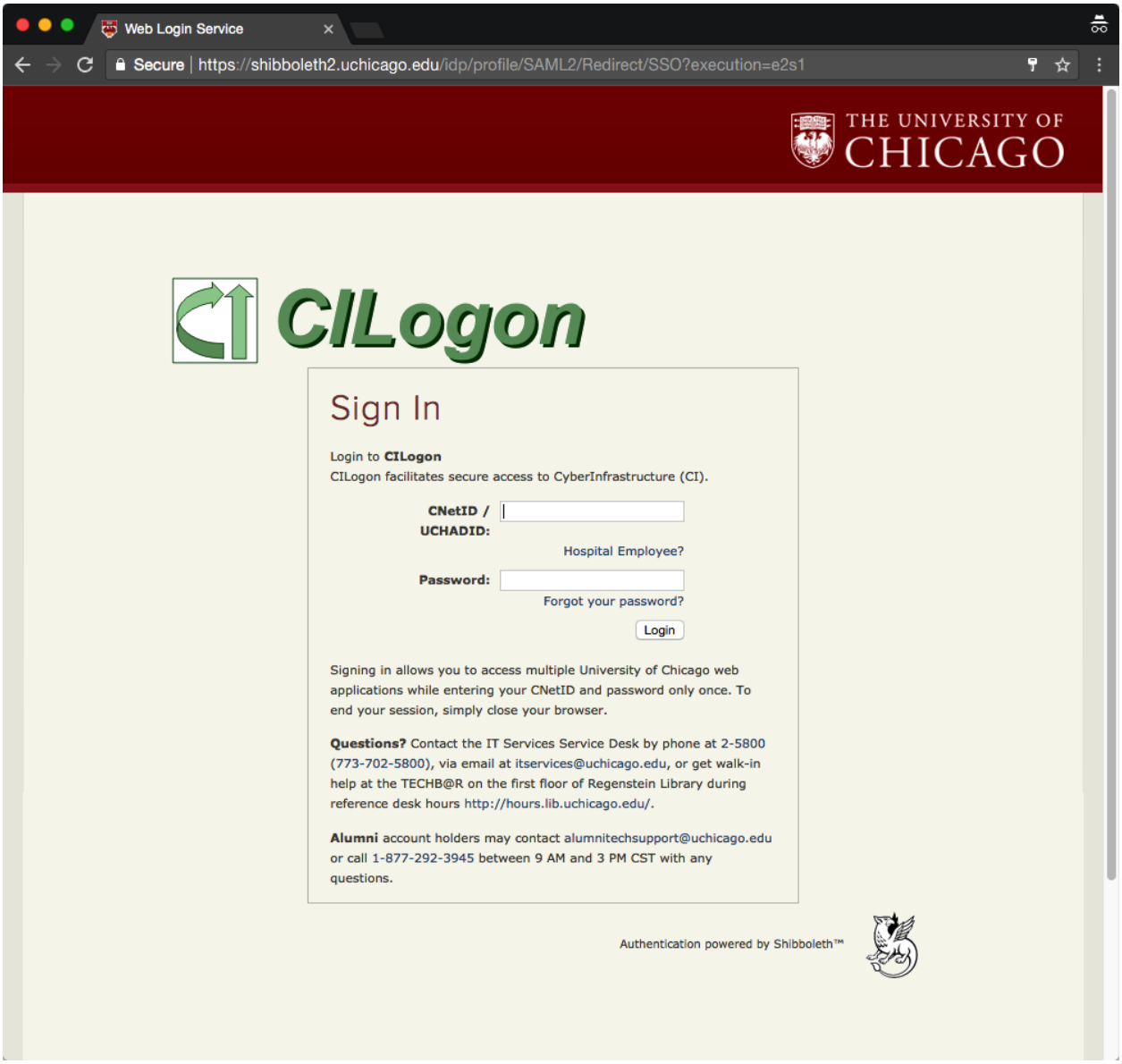
You will then be asked to sign in with your institutional identity, or your Globus ID. If you are using the former, simply type in the name of your institution and click "Continue". Proceed to Step 3a.

Otherwise, click "Sign in with Globus ID" and proceed to the alternate Step 3b.



3a. Login with your institutional ID

You should be presented with a login page for your institutional ID, with your institution's branding. Go ahead and sign-in now. Note that your password is *not* sent to the VC3 or Globus web portals. Continue to step 4.



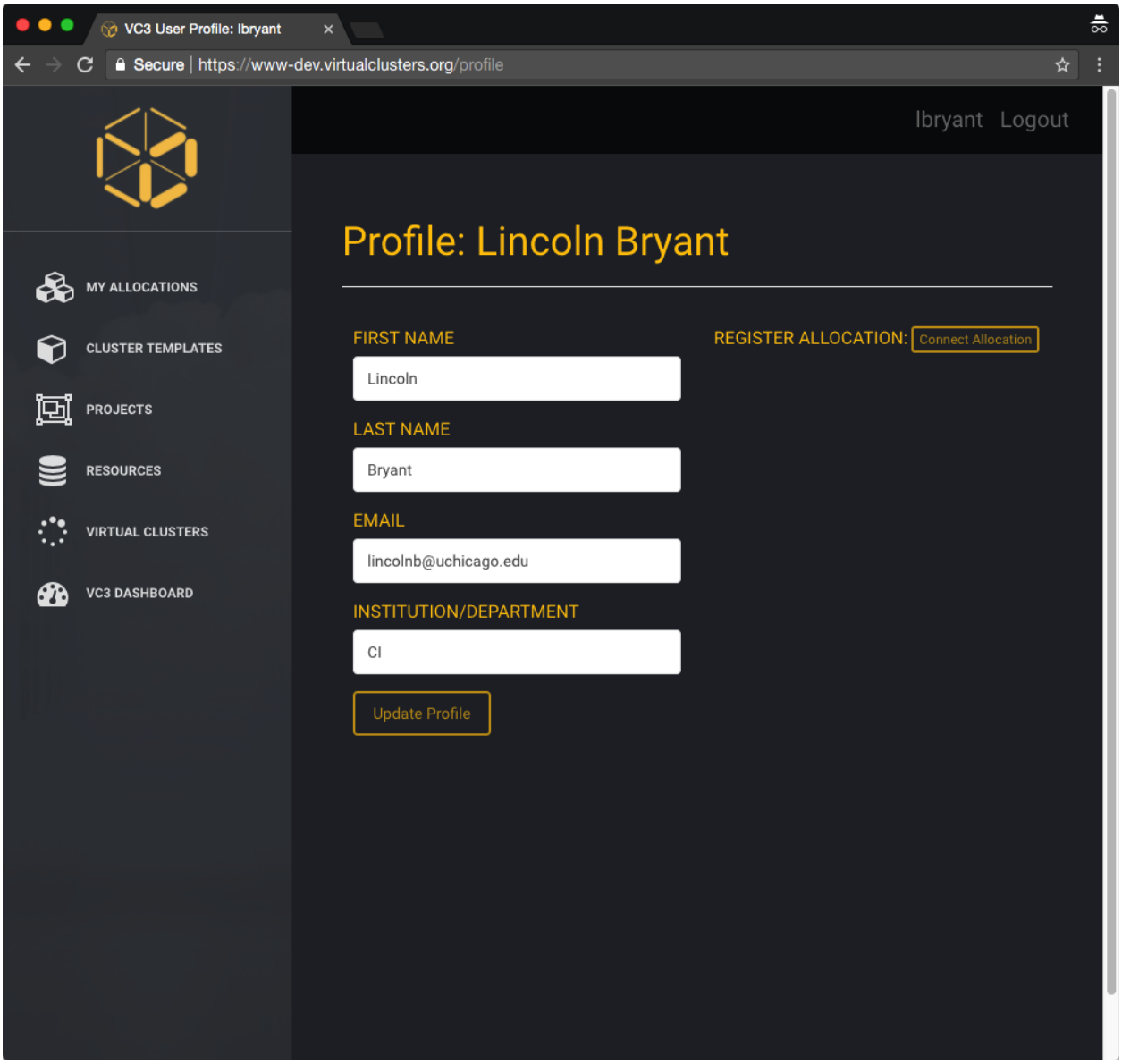
3a. Login with your Globus ID

(_Alternative step - if you do not have an institutional ID supported by **Globus**_)

<– Globus ID page –>

4. Complete or update your VC3 profile

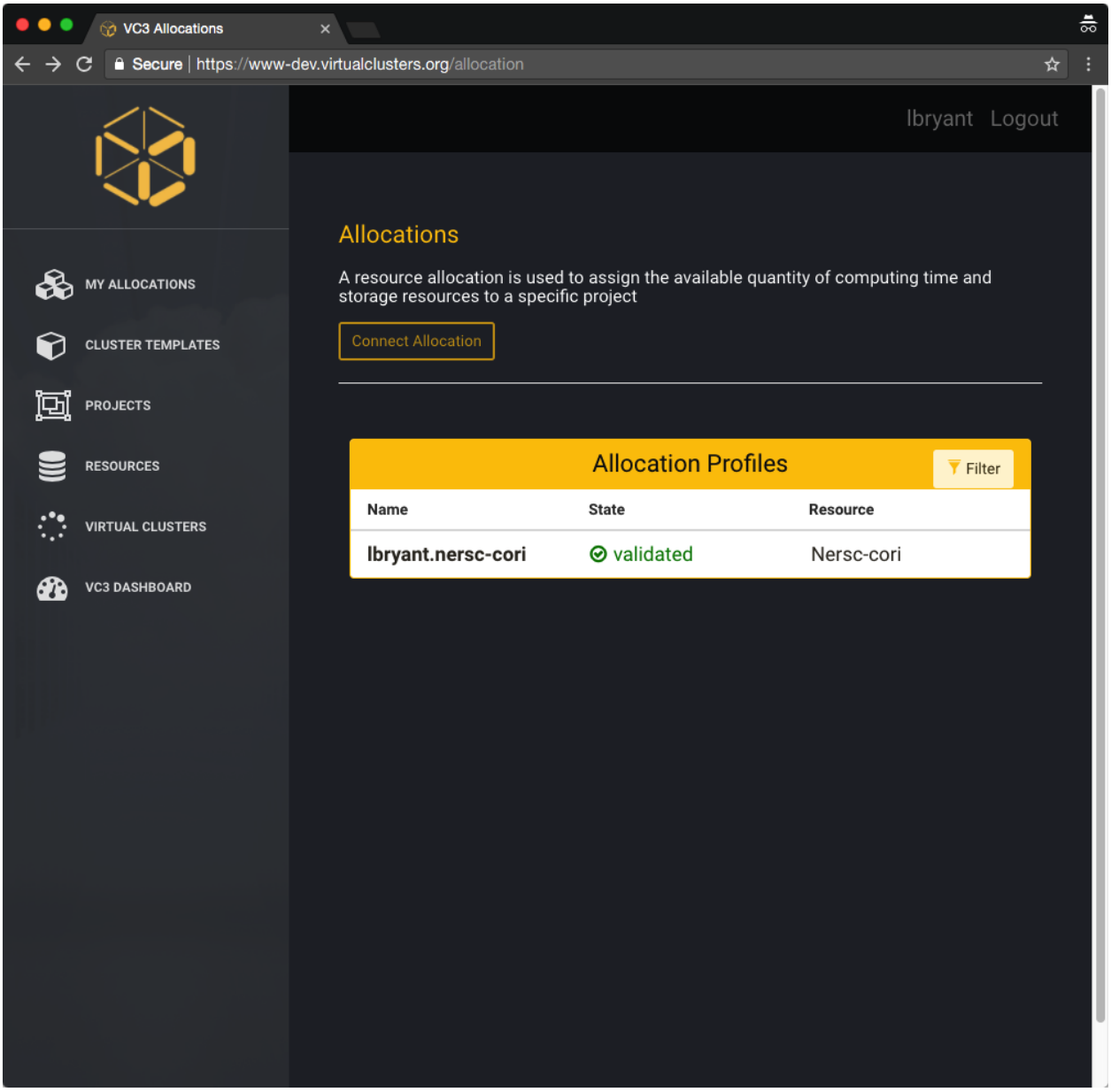
Once you have signed in, you'll be asked to update or complete your VC3 profile with information such as your Institution and any other information we cannot directly extract from your Globus account. Click "Update Profile" once done.



5. Connect an Allocation

After updating your profile, you can connect an allocation to the VC3 service. An allocation, in VC3, is defined as combination of a username and resource target that consumes some type of compute unit - regardless of whether it is billed as Service Units (many HPC centers), dollars (AWS, GCE), or priority (HTCondor and other opportunistic systems).

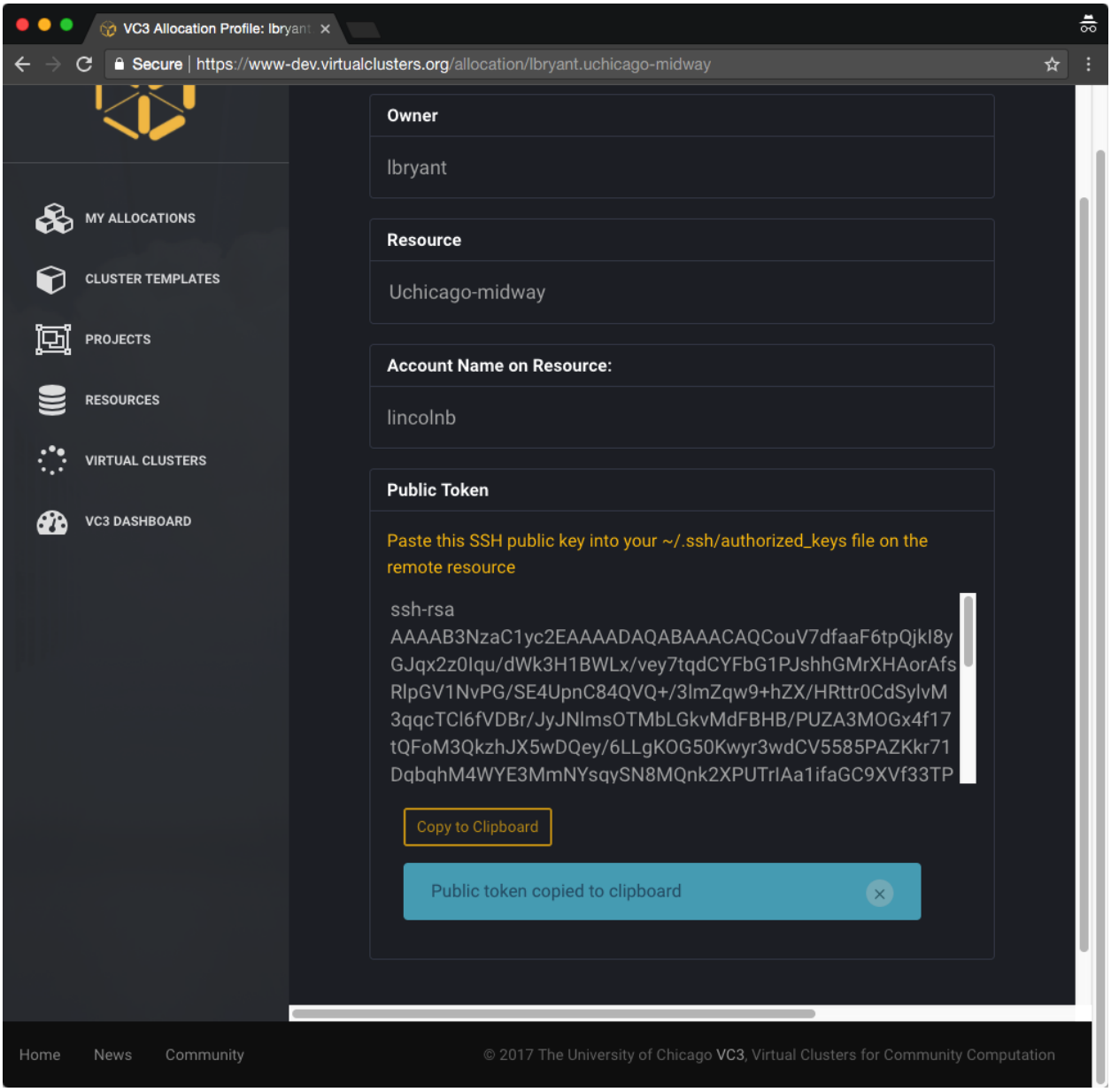
Clicking *My Allocations* on the left shows all allocations currently associated with your account. You may select a new one by clicking *Connect Allocation*.



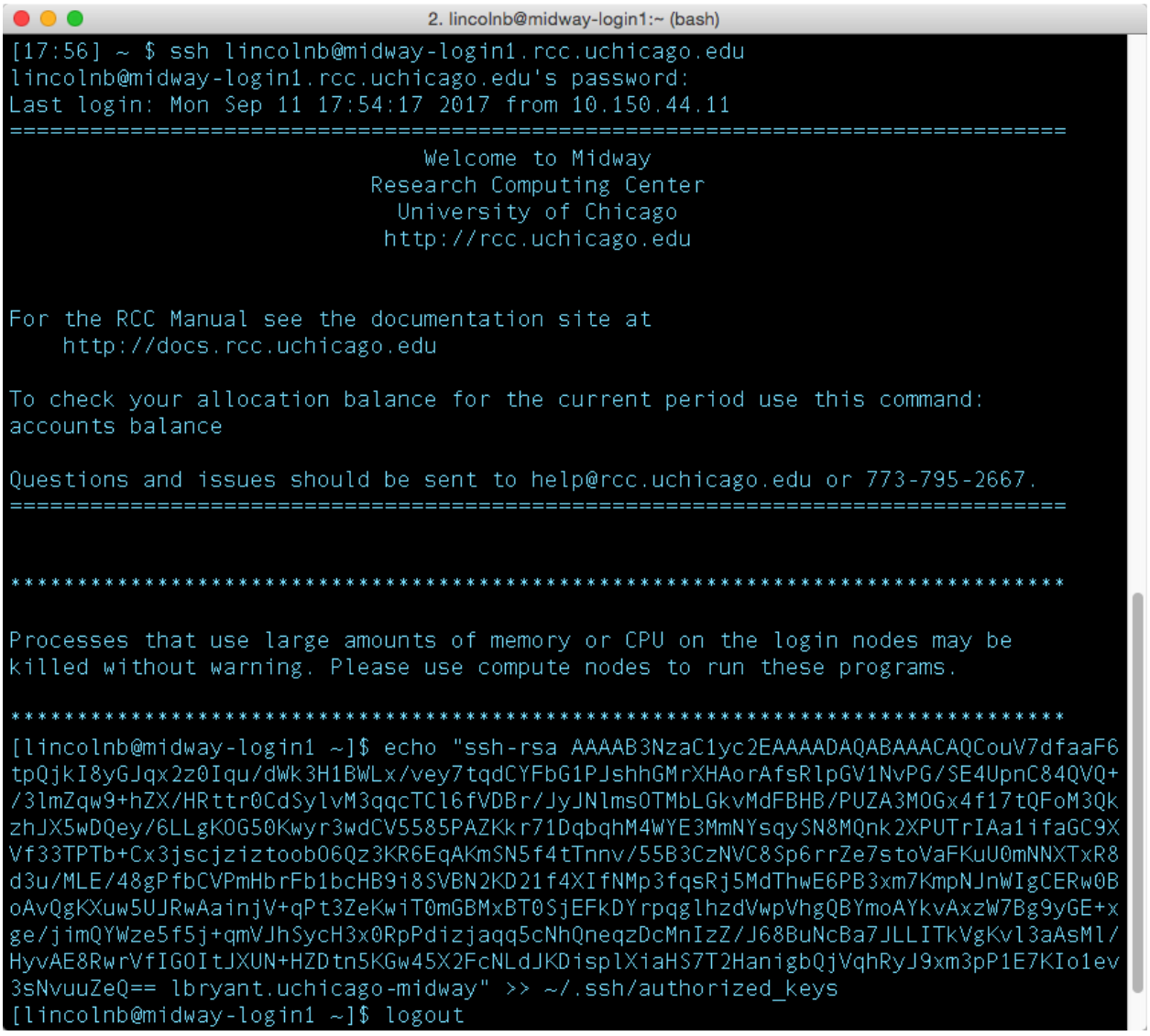
You will be able to select a resource target from the drop down menu, and enter an account name for the resource. This is the same account name that you use to SSH to the remote system. .. image:: /image/screenshot_278.png

Once you've connected your allocation, the system will validate it. .. image:: /image/screenshot_279.png

In order to create a virtual cluster, the VC3 software needs to be able to SSH to the remote resource. If you click your allocation, you should see a section titled *Public Token*.



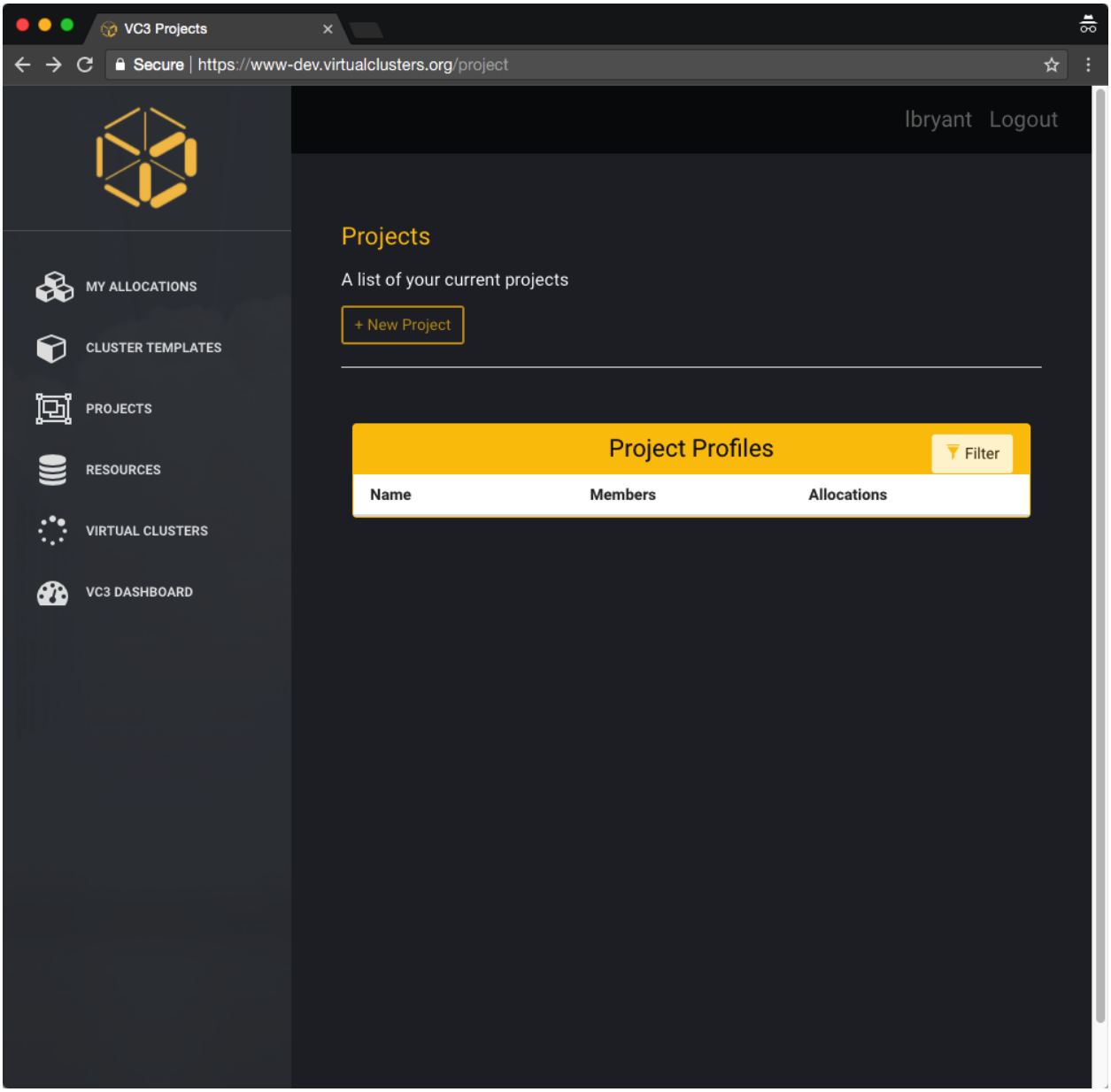
You will need to add this token to your Unix account, in the file *~/.ssh/authorized_keys*. You can either edit this file with your favorite editor (such as *nano*, *vim,* or *emacs*), or use the *echo* command to append it to the authorized keys file.



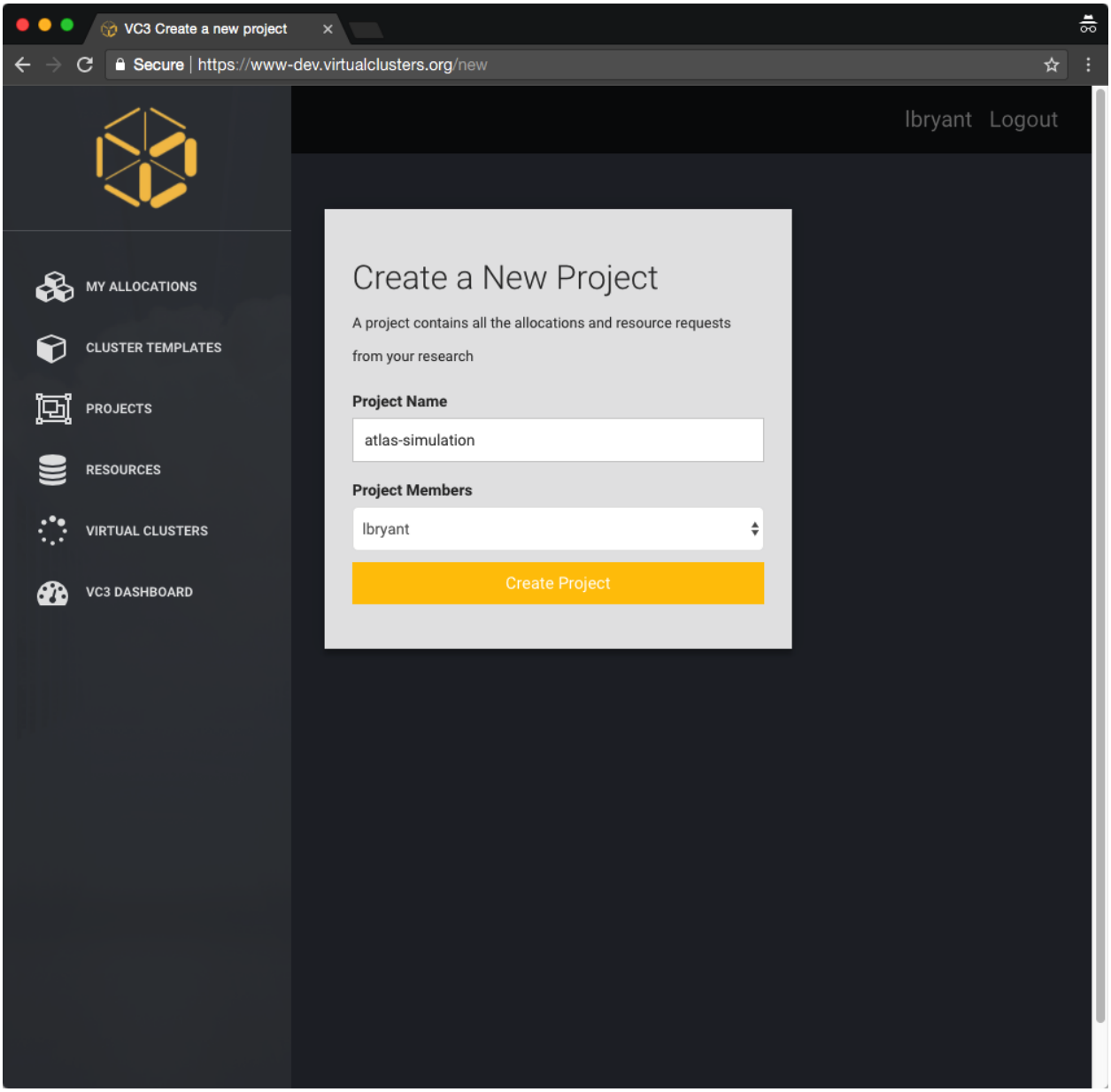
This token allows the VC3 system to SSH into a cluster _as **you**_ and submit jobs on your, or your project's, behalf.

6. Defining a Project

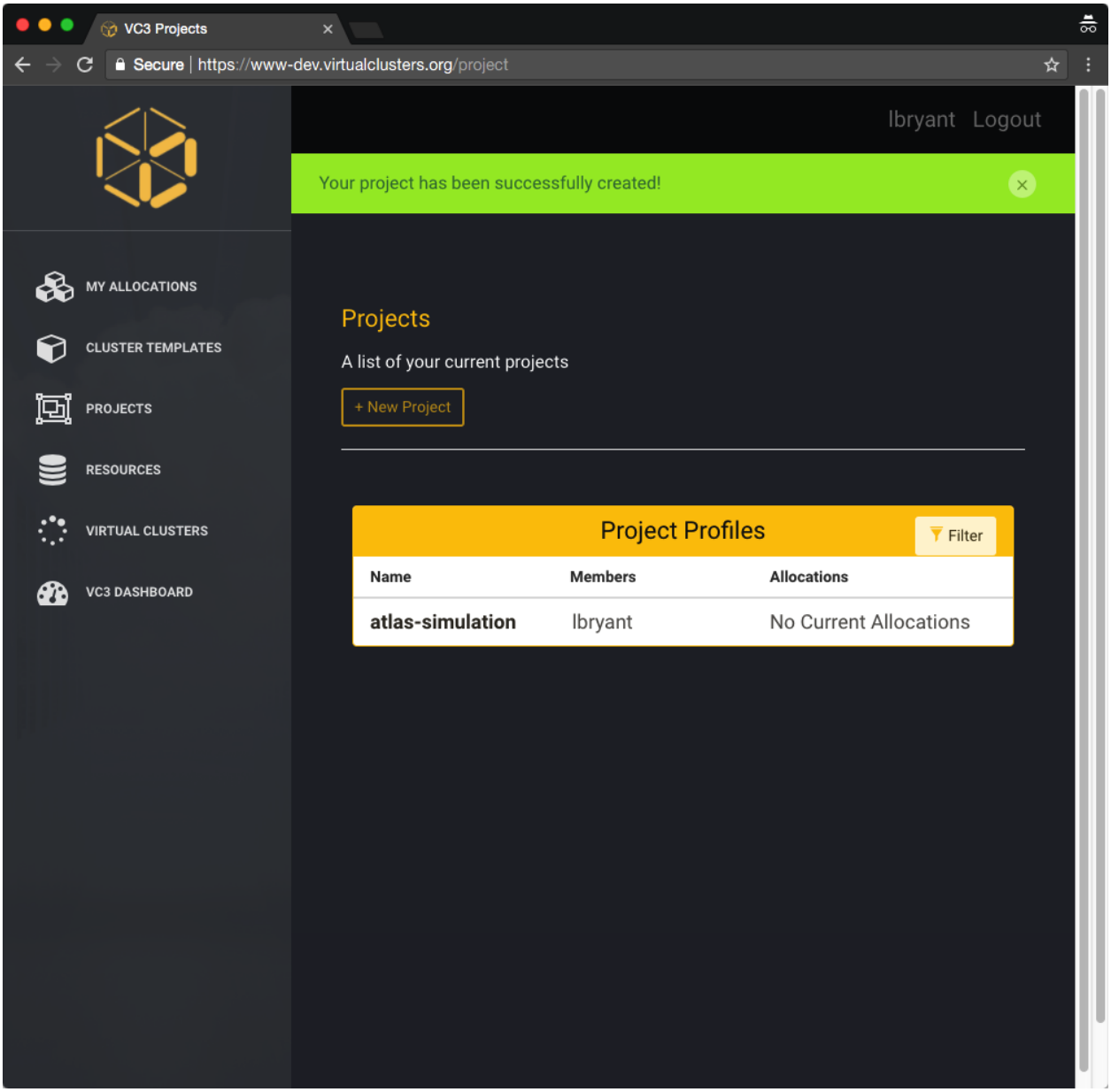
VC3, as a platform for cooperative scientific computing, allows you create projects to share your allocations and virtual clusters with trusted members of your group, laboratory, or collaboration. To start a new project, click "Projects" on the sidebar, then click "+ New Project".



You may give your project an aribtrary name and choose initial project members. Once finished, click “Create Project”.



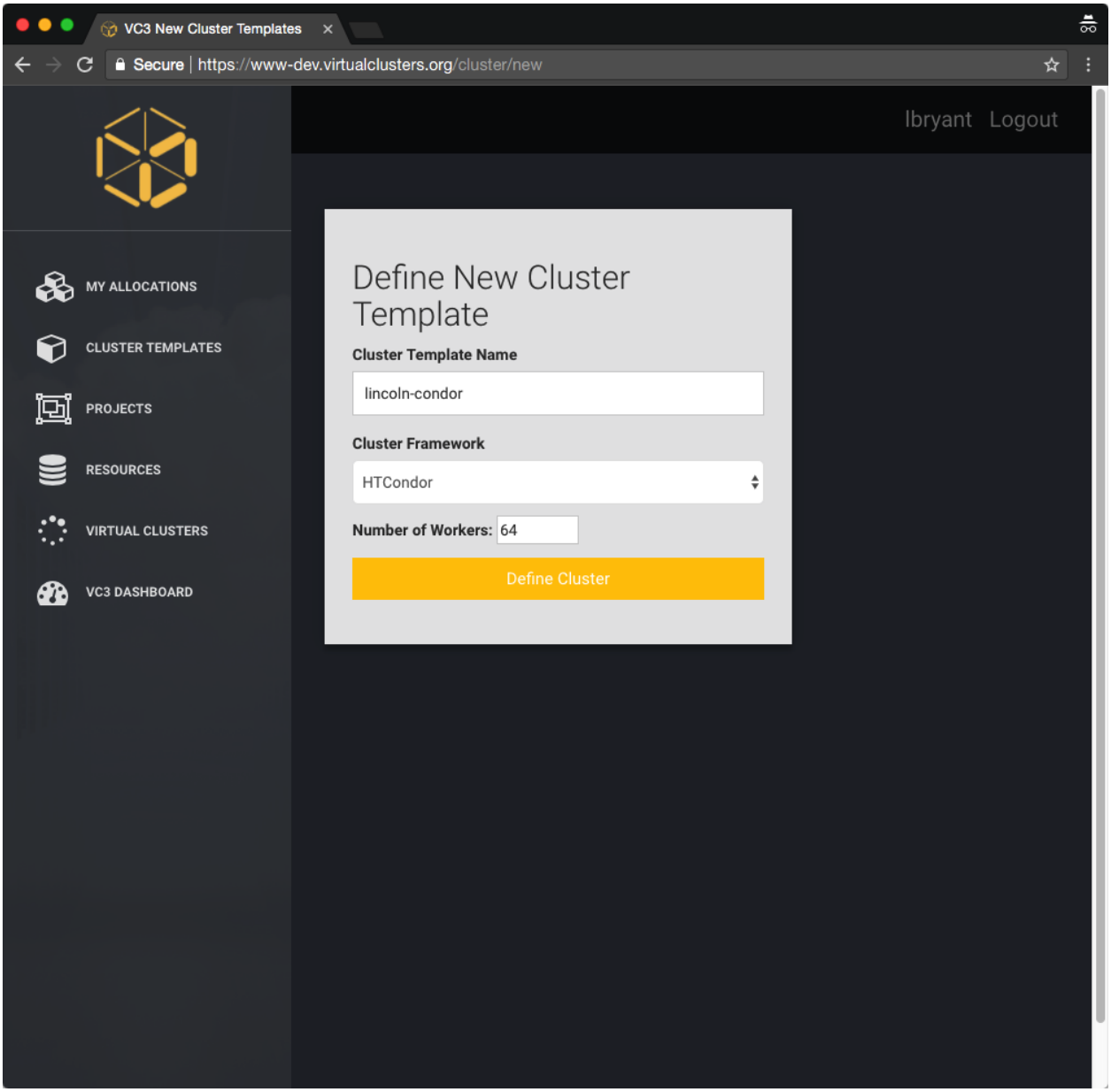
You should be returned to the Projects page, where you will be able to see all of your projects and memberships.



7. Creating a Cluster Template

VC3 allows users to create "Cluster Templates" that describe the components of their virtual cluster, including number of head nodes, worker nodes, etc. We currently support HTCondor and WorkQueue clusters with dynamic worker nodes, and fixed head nodes.

To define a new template, click the "Cluster Templates" link on the left panel. You'll be able to give your cluster a name, select framework, and number of workers. Click "Define Cluster" to finish creating the template.



8. Resources

The VC3 team curates an ever-expanding list of resources for end-users, with a focus on Campus Clusters, HPC centers, and Cloud resources. You can find these resources by clicking the "Resources" link on the left panel.

You can also click an individual resource and see expanded information, such as batch system type, links to documentation, etc.

8. Launching a Virtual Cluster

2.2 Installation

To write example code for VC3:

```
python<version> -m pip install vc3
```

Note: It is **very** important to install VC3 on the *correct* version of Python for your needs. Example Note here. `commands within notes`.

CHAPTER 3

FAQ and Glossary

3.1 Glossary of Terms Used in VC3 Documentation

3.1.1 Internal

cluster states list of each possible state of a cluster throughout its lifecycle:

> NEW - Request was just created. VALIDATED - Request has been validated for basic correctness. PENDING - Request is valid and is waiting to be instantiated. GROWING - Cluster is in the process of being instantiated but is not yet usable. RUNNING - Cluster is ready to use. SHRINKING - Cluster resources are being removed. TERMINATING - Cluster is about to be destroyed. TERMINATED - Cluster no longer exists.

credible Credible is a 3rd-party utility for programmatically generating, storing, and retrieving security tokens.

dynamic infrastructure Services that are instantiated upon a virtual cluster request, such as the factory.

factory The scheduler and resource manager for middleware.

formatter A plugin that augments the output of **|Flake8|** when passed to `flake8 --format`.

info service Long-running daemon that interacts with the information database on behalf of other services.

> The VC3 info service serves as both a persistence mechanism for the overall service, and a message bus between components. Information is stored and retrieved in the form of JSON-formatted documents, which thus form a single tree of information entities/nodes. The service optionally allows access security by enforcing ACLs at each node level.

PIN Personal Identification Number. One-time password for configuring a VC3 resource via vc3-resource-tool

plugin-manager The plugin manager is a 3rd-party small utility for quickly constructing plugin objects from configuration input.

request ID Unique identifier for a virtual cluster request.

static infrastructure A set of long-running services, such as the Info Service, Master, etc.

vc3-application One of the supported middleware applications to be deployed as an overlay defining a virtual cluster.

vc3-builder Pilot-like executable that prepares an environment for middleware and user applications. The vc3-builder is a pilot-like utility, submitted to resource targets, which programmatically satisfies all requested dependencies before handing off control to the middleware layer. Its special feature is the ability to satisfy dependencies in different ways on different targets, depending on what it finds, e.g. it can tell if a dep is already satisfied, can download a pre-built library, or dynamically compile a dep if needed. Several builders can simultaneously satisfy dependencies in parallel on a resource (provided a shared filesystem).

vc3-client Package containing the VC3-aware library for creating, listing, updating, and deleting entities within the infoservice. It also contains a command line interface to the library.

vc3-core The VC3 component that coordinates activity within the dynamic infrastructure. One vc3-core exists per virtual cluster Request during its lifecycle. A vc3-core will typically start a vc3-factory, along with any central components the cluster will need (e.g. an HTCondor collector/negotiator/schedd, a WorkQueue catalog, or a Squid server).

vc3-master Package containing the long-running daemon, running on the static infrastructure, that manages the lifecycle of all virtual cluster Requests. The vc3-master is a long-running daemon, running on the static infrastructure, that manages the lifecycle of all virtual cluster Requests. It polls the infoservice for new Requests, and spawns vc3-core instances on the dynamic infrastructure to service them. It also handles the generation and processing of all derived entities within the infoservice tree.

vc3-release This is a developer package that contains various setup and test utilities, and artifacts needed to create and use a YUM RPM repository.

vc3-resource-tool The vc3-resource-tool is a utility to be run by end users on resource targets in order to pair and enable them for usage by the VC3 system.

3.1.2 External

Allocation An Allocation refers to an User and a Resource Each Allocation must be owned by an User. Allocations are divisible/fractionable, and can be given to Projects. Allocations may not be oversubscribed. But unbounded Allocations may be parents of multiple unbounded SubAllocations. Bounded Allocations cannot spawn unbounded SubAllocations. If a Resource grants hard allocation and allows backfill mode, those are two distinct Allocations (one hard and one unbounded)

Authentication The current mechanism for users to sign-up and create accounts into the VC3 project is by authenticating themselves with their GlobusID account.

MFA multi-factor authentication

Project a collection of "Allocations". It has at least 1 "user owner", and 0 or more non-owner members. The owner is also a member.

Request Entity that encapsulates all information that defines a particular virtual cluster. Creating a new Request triggers creation of the cluster.

Request templates a list of pre-existing forms to be used as base for new cluster requests creation.

Resource Any target on which a vc3-builder will run to provide computing power to a virtual cluster.

Resource profiles a list of pre-existing forms to be used as base for new resource definition.

Service unit Service units are essentially just walltime hours, with minimum charges based on minimum cores or minimum nodes per job. Much like HEPSPEC, the SUs can be normalized/converted based on LINPACK benchmarks. Doc from XSEDE: https://portal.xsede.org/knowledge-base/-/kb/document/bazo

For storage, possibly with multiple allocations per user, examples are scratch disk vs long term storage.

Exotic devices like GPUs may or may not be accounted for, depending on the resource.

Sub-Allocation A SubAllocation can be defined in terms of fraction or units (cpuhours?, $dollars, HEPSPEC) or be unbounded. SubAllocations are children of an Allocation.

User Every User has 0 or more Allocations. Users are owners or members of one or more projects. A User in a project can make Request(s) utilizing project member's Allocations

CHAPTER 4

Contributor Guide

If you are reading VC3's source code for fun or looking to contribute, you should check out the github repository.

CHAPTER 5

General Indices

- genindex
- *Glossary of terms*

Index