Hands-on with CUmulus: a free Research Cloud platform for RMACC

Getting Started on CUmulus

Andrew Wilson (RC User Support)

www.rc.colorado.edu

rc-help@colorado.edu

Learning Objectives

- Logging into CUmulus with Horizon (the CUmulus web portal)
- Creating your instance (i.e. virtual machine)
- Logging into your instance via ssh

Account Setup for Demo

- What's needed for today:
  - *Access-ci Login information*
    - In the format: awilson1@access-ci.org
  - *CU Research Computing account*
    - Only needed to login to Instance
    - curc.readthedocs.io

Outline

- CU Research Computing Overview
- What is CUmulus?
- CUmulus Access
  - *Access to CUmulus and the allocation process*
  - *Logging into Horizon (CUmulus web portal)*
  - *Creation of an instance*
  - *Logging into your instance*
- Demo workflow one might use on CUmulus:
  - *Web App & Database*

CU Research Computing Overview

- Provides Computing and Data Storage Beyond standard desktop resources:
  - *High Performance Computing (HCP), clusters include:*
    - Alpine
    - Blanca
  - *Storage of Research Data*
    - PetaLibrary
  - *High-Speed Data Transfer*
    - Globus Endpoint
  - ***Cloud Resources***
    - **CUmulus (On-premise Cloud)**
    - Cloud Foundations (Commercial Cloud)
  - *Secure Cloud Resources*
  - *Consulting in Computational Science and Data Management*

CU Research Computing Overview

What is CUmulus?

- **CUmulus** is CU Research Computing's free-to-use on-premise cloud service
- Supports cases not well-suited for HPC such as:
  - *Research-Oriented Web Servers*
  - *Databases*
  - *Long-Running Services*
  - *Research Hubs (Jupyterhub, RStudio Server, etc)*
- Provides users with persistent or ongoing availability by allocating logically isolated section of the cloud

What is CUmulus?

- You get your own virtual environment for experimentation - an environment that can be easily created/tested/removed
  - *Install Software*
  - *Administer your instance (you're in control!)*
  - *Run applications and jobs*
  - *Interface w/ other CURC services: Blanca, Alpine, PetaLibrary*
- You can request specific resources (CPU, storage, memory) and can set up persistent storage

Example Use Case: Persistent Data Mining

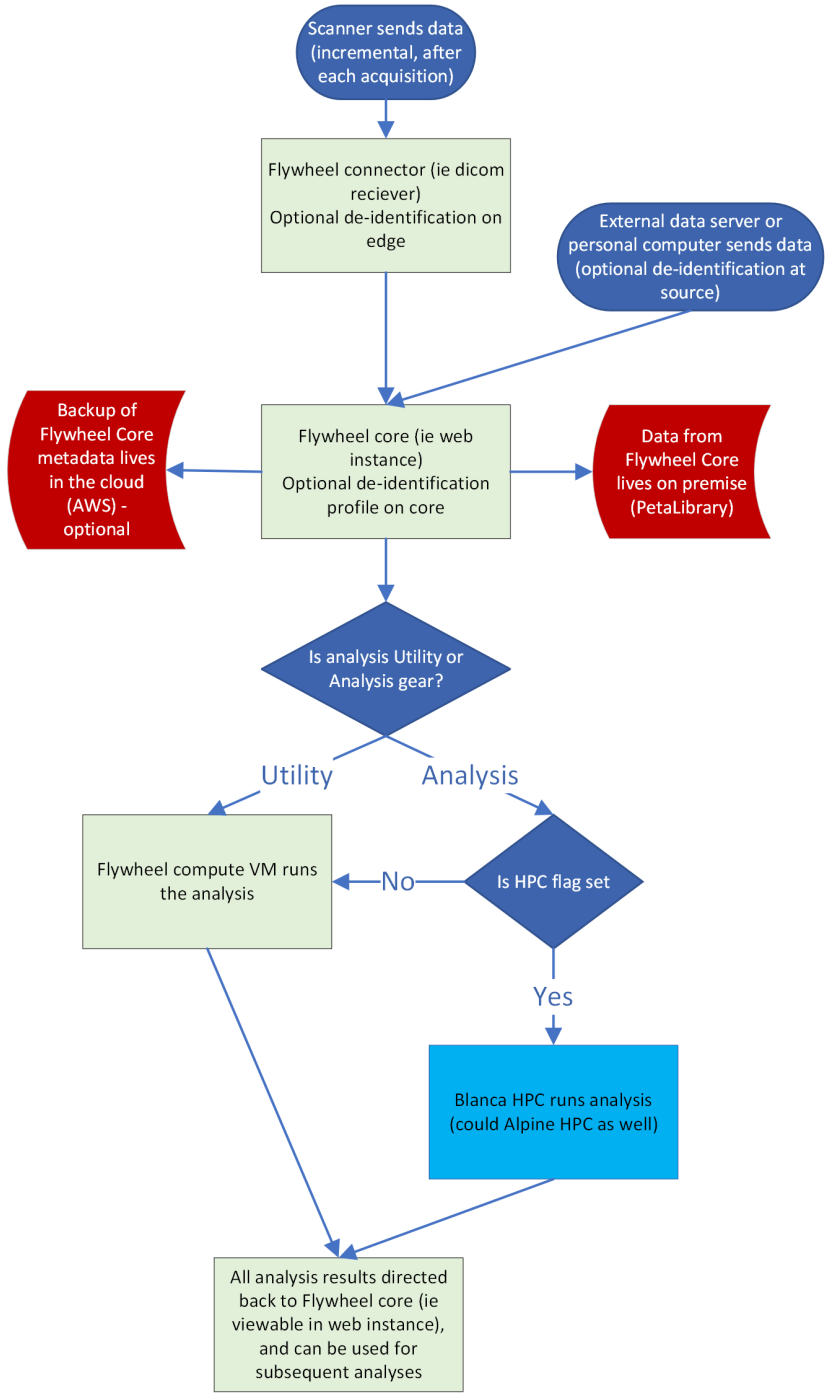
- Researcher wanted to use online text database to train novel search algorithm
- Needs:
  - *Persistent Querying of an online text database*
  - *Persistent storage of data*
  - *Compute ability to continuously mine the data and advance their algorithm*
- They were able to:
  - *Setup a data stream that queries data from Reddit at specified times of the day*
  - *Have the resources for persistent storage of large datasets and compute resources*
  - *Test and develop their novel search mechanism*

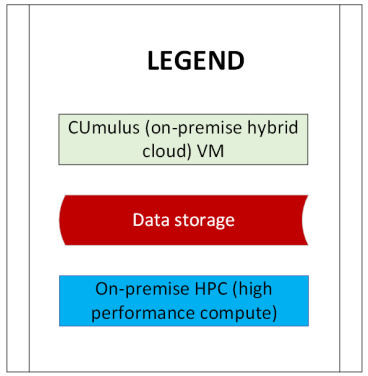
Example Use Case: Galaxy Server

- Researcher wanted to create a custom Galaxy Server (web-based GUI for popular bioinformatics software)
- Needs:
  - *A way to host a website that the researchers can use to load and analyze data, create reports and schedule compute jobs*
  - *Persistent storage of data*
  - *The ability to create compute jobs on the virtual machine or send jobs to the Alpine Supercomputer*
  - *Run persistent jobs (multiple days long)*
- They were able to:
  - *Host the website, and control access of users who can access it*
  - *Have the resources for persistent storage of large datasets and compute resources*
  - *Create an interface to send jobs to the Alpine Supercomputer, all controlled via GUI*
  - *Test custom tools and data pipelines that they created*

Example Use Case: Open Access Research Tool

- Researcher wanted to publish and host trainings for a custom language processing tool that helps to build language databases and tools (think auto-correct) for endangered languages
- Needs:
  - *A web-based dashboard where any user can load and use the tools the research team developed*
  - *Persistent storage of data*
  - *Compute ability to continuously mine the data and advance algorithm*
- They were able to:
  - *Setup a Jupyterhub instance where they control access of users*
  - *Have an environment where researchers around the world can easily use their research tools*
  - *Collect the incoming test data and use that for future training*
  - *Have the resources for persistent storage of large datasets and compute resources*





Example Use Case: Flywheel Project

- Researcher wanted to create a tool that interfaces with MRI machines in the area and pulls data to a persistent web application where they can create processing pipelines and send these to the Alpine Supercomputer
- Needs:
  - *A web-based dashboard where users can develop processing pipelines, move data, and create compute jobs*
  - *Persistent storage of data (very large datasets)*
  - *Compute ability to continuously process the data*
- They were able to:
  - *Setup a Flywheel Dashboard, a web-based application that researchers can access from anywhere.*
  - *Store data on the VM or in PetaLibrary or AWS*
  - *Send jobs to the Alpine Supercomputer for faster processing*

CUmulus Access

CUmulus Access and Allocation

Submit a proposal for your use case (email rc-help@colorado.edu)

- Describe your CUmulus workflow
- Describe why your workflow is appropriate for CUmulus
- Estimate the resources you require:
  - *Operating System, CPU cores, Disk Space, Memory*

This is an *iterative process* where we work with you to make sure the request for resources fits your (and our) needs

- Learn more about the allocation request process at https://curc.readthedocs.io/en/latest/cloud/cumulus.html

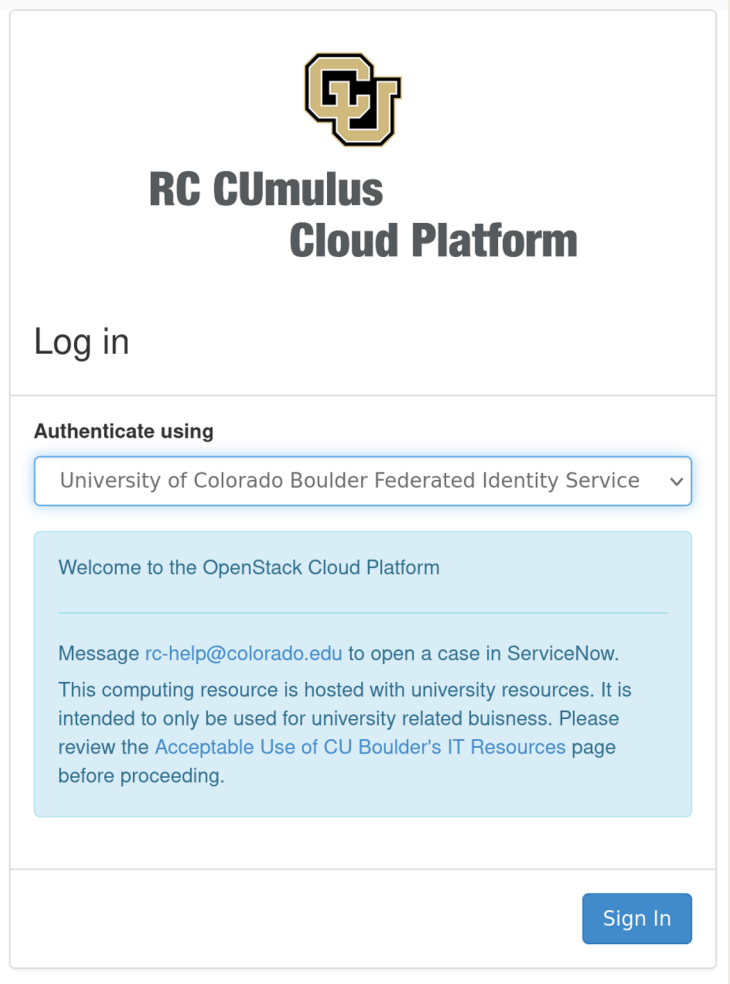
Log in to Horizon

There are 3 current authentication/login methods at cumulus.rc.colorado.edu/:

1. CU Boulder
2. CSU Fort Collins
3. ACCESS-CI (all other RMACC institutions)
    a. *Create account: https://identity.access-ci.org/new-user*
    b. *Configure 2FA (Duo)*

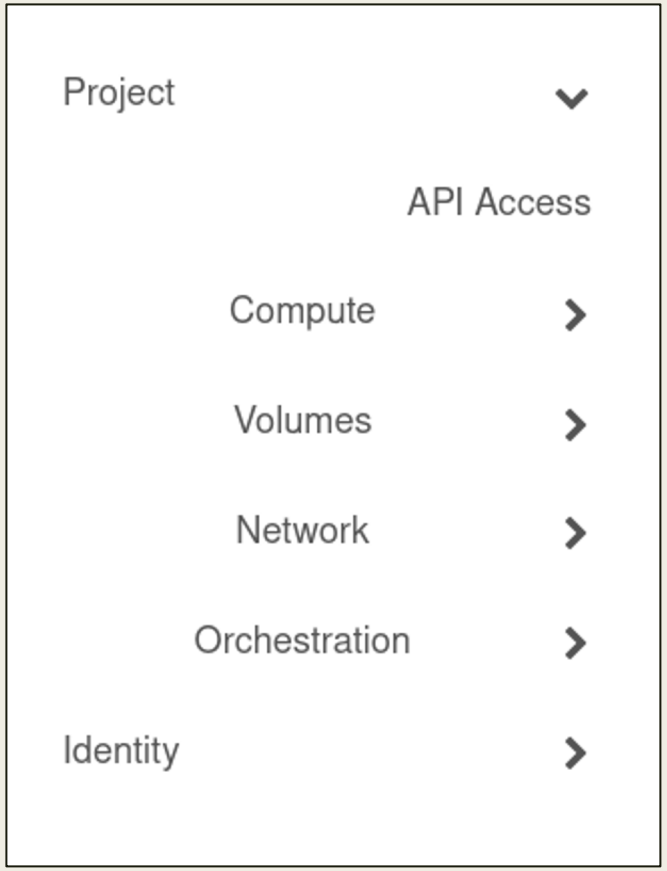
Log in to Horizon

- Horizon is the CUmulus web portal
  - *cumulus.rc.colorado.edu/*
- Let's take a brief tour of Horizon
- Log in with your institution's credentials:
- If you're using your ACCESS-CI Account
  - *Choose: 'Access-CI Identity Provider'*



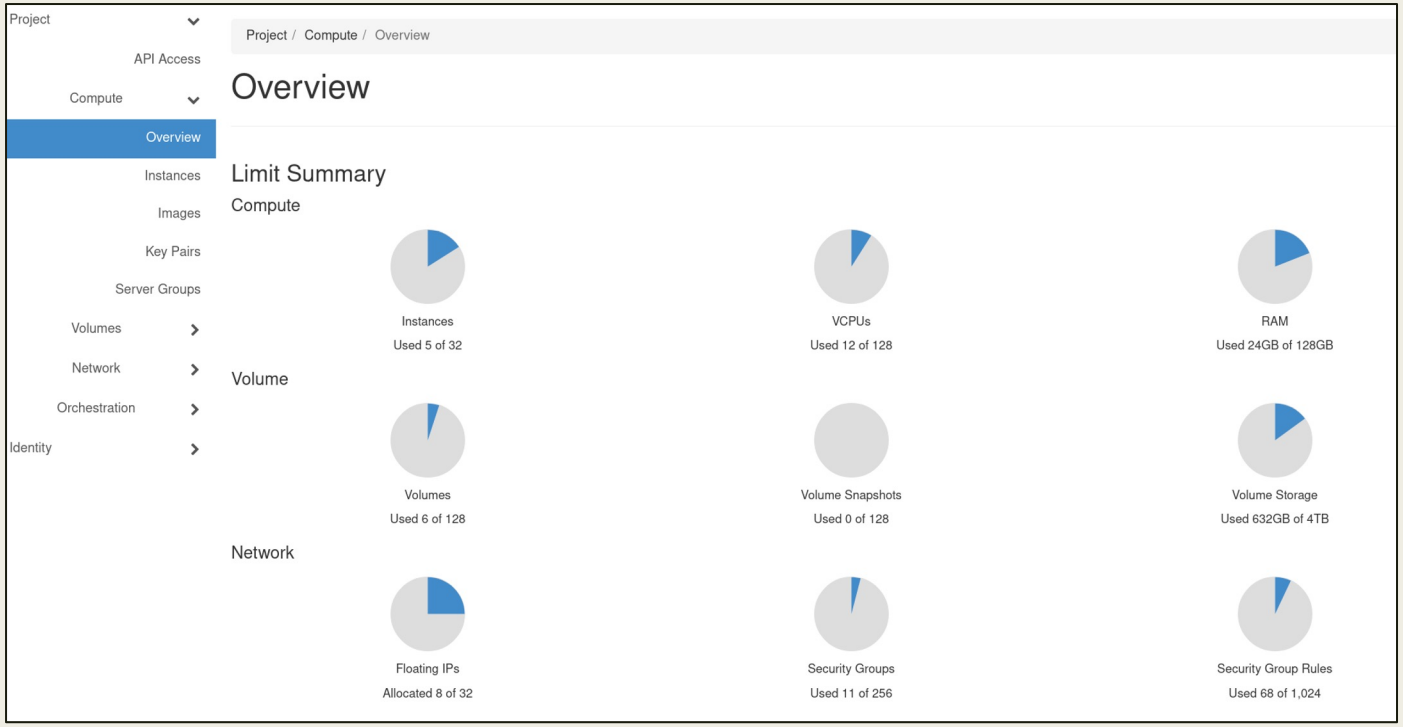
Navigate Horizon

- Choose your project (top left)
  - *Generally users only have 1 project*
- 4 main sections
  - *Compute*
  - *Volumes*
  - *Networks*
  - *Orchestration*



Navigate Horizon: Overview

- Land on the Overview page under “Compute”
  - *quick summary of your project*



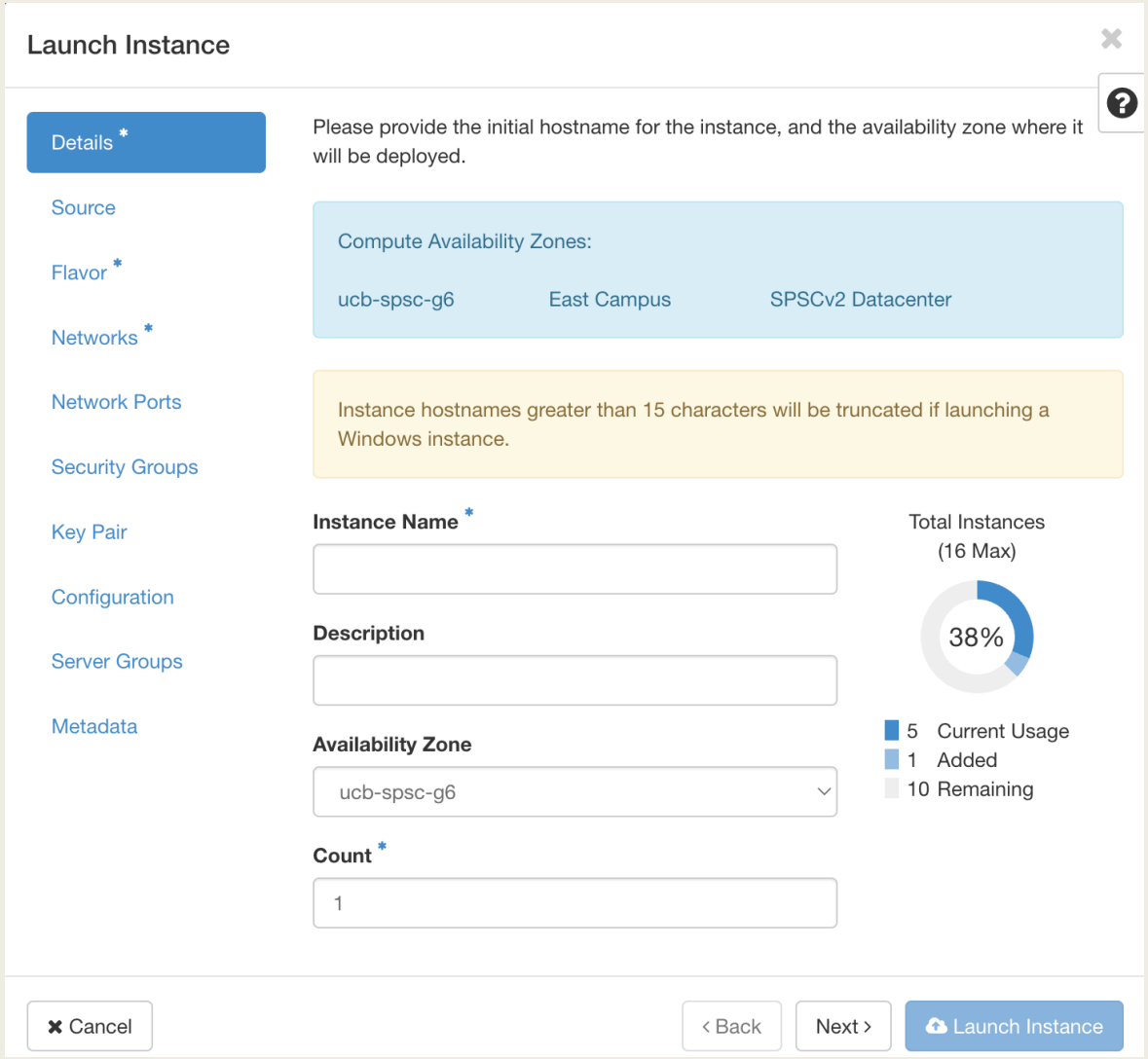
Navigate Horizon: Instances

- Navigate to:
  - *Project->Compute->Instances*
- Instances are **virtual machines** that run inside the cloud, more simply: an instance is just a digital version of a physical computer
  - *Instances can perform almost all the same functions as a computer, including running applications and operating systems*

Instance Creation

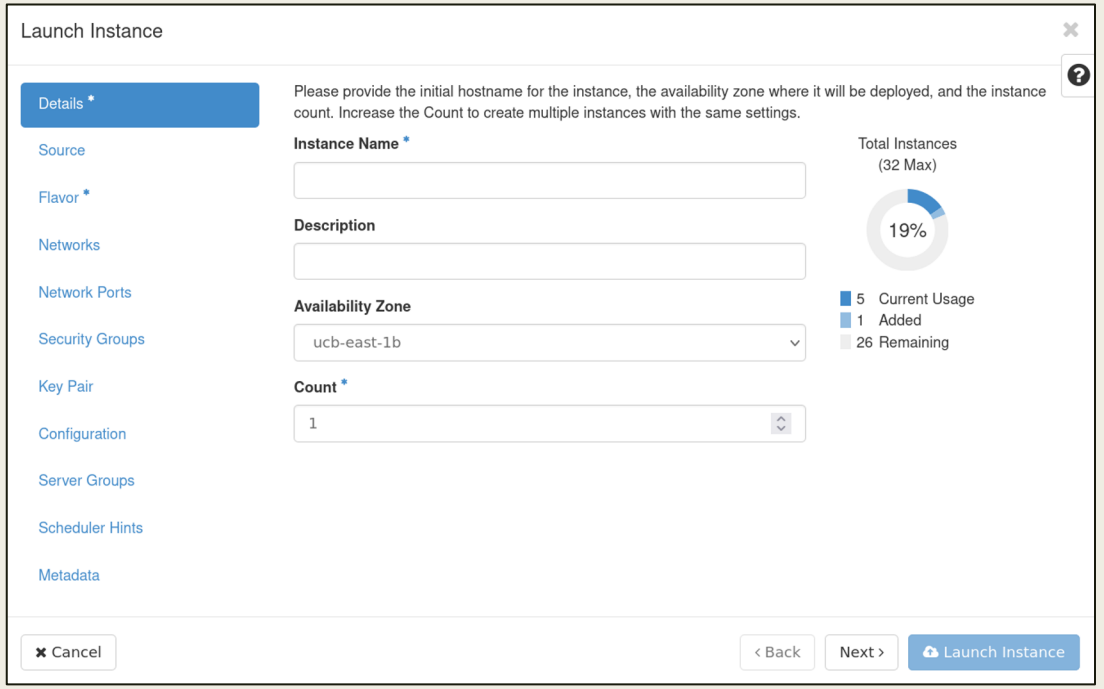
Let's create a simple instance together

- From the instances page click on "Launch Instance"
- The Instance Creation Launcher will pop up giving us options to create our virtual machine



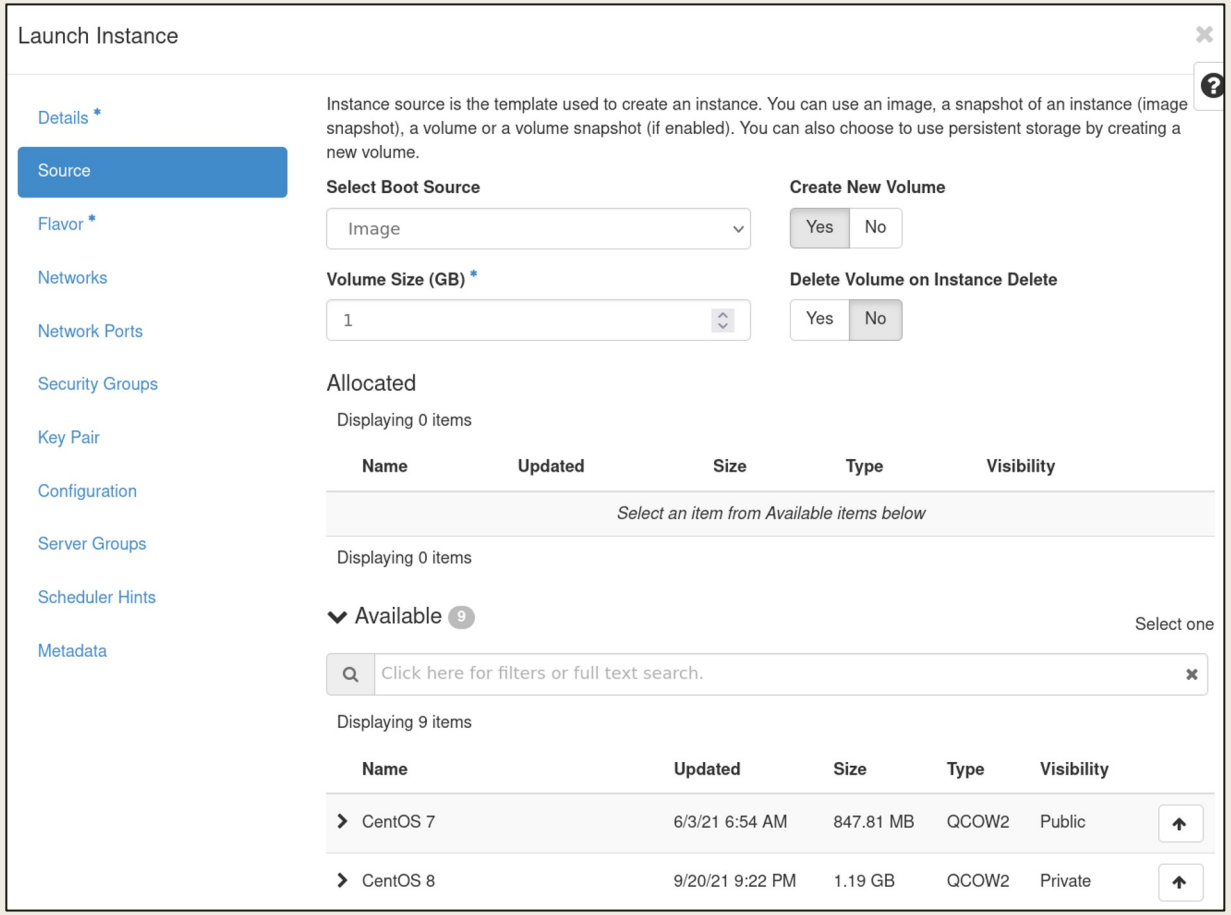
Details

- Fill out Instance details, including a name and description
  - *availability zone and count can be left as defaults*



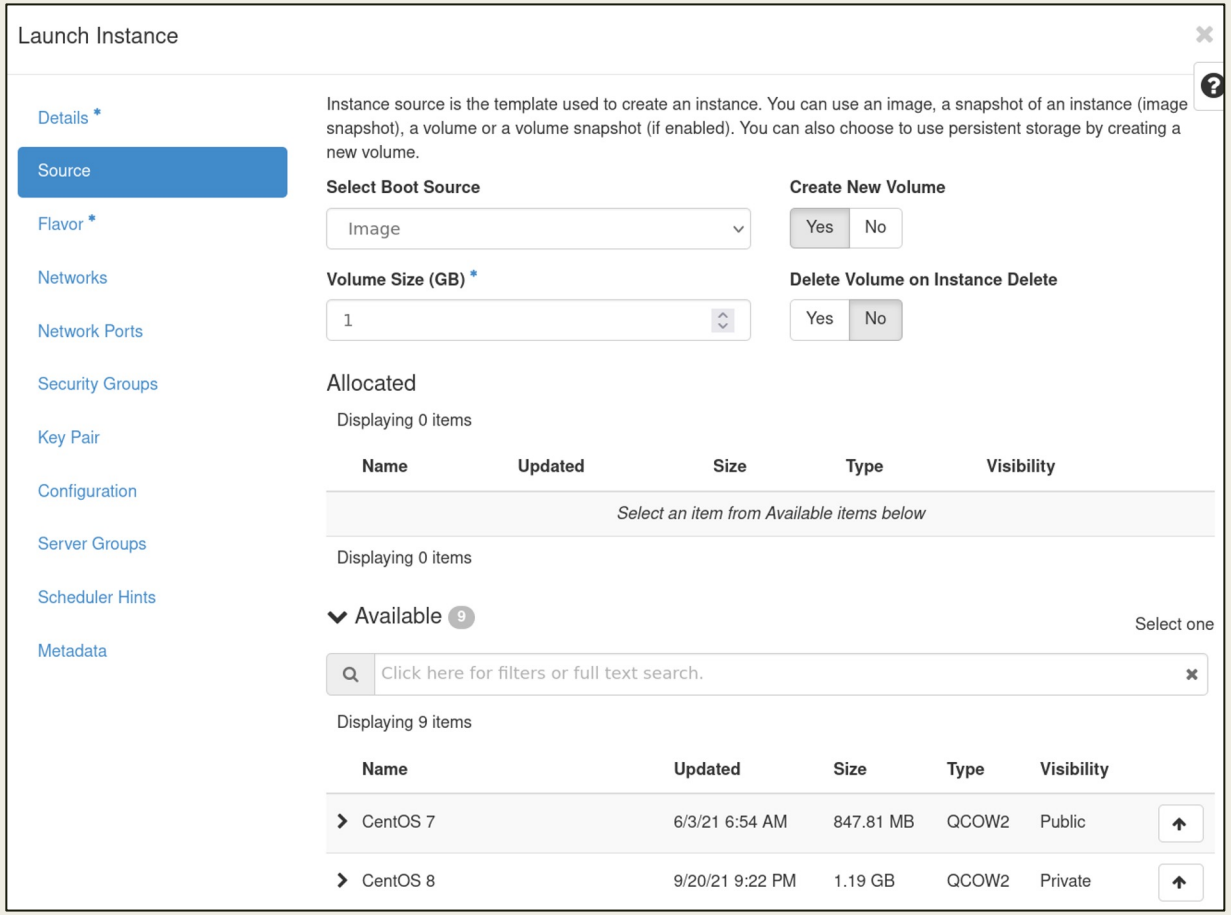
Source

- Choose an image from the available list
- A virtual machine image (“image”) is **a single file that contains a virtual disk that contains a bootable operating system**
- Choose to have your storage volume deleted on instance deletion
  - *If you select “no” be aware of “zombie” volumes*



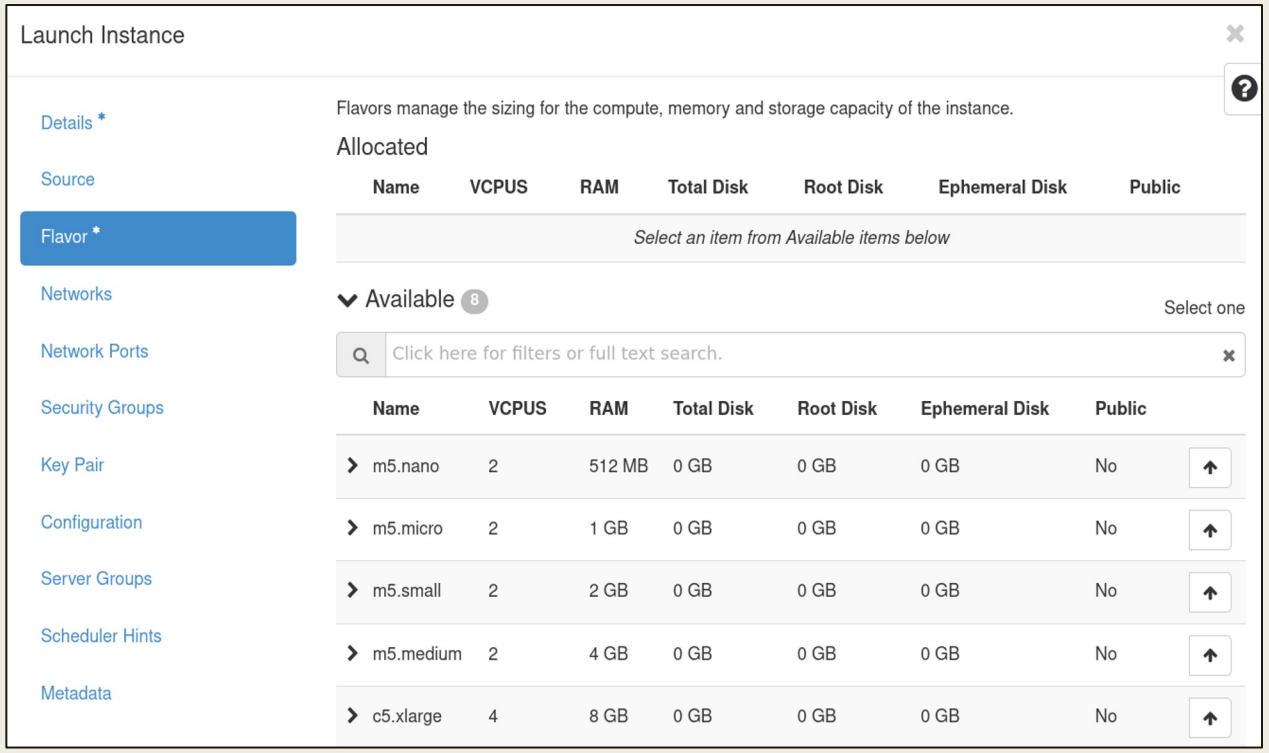
Source

- A brief note about Source Images
  - *We have custom images for things like The Littlest JupyterHub, Rstudio Server, Galaxy Server, and LAMP servers.*
  - *Makes setup simple for these tools, with only the need to setup SSL.*
- **Anything else you would find useful?**



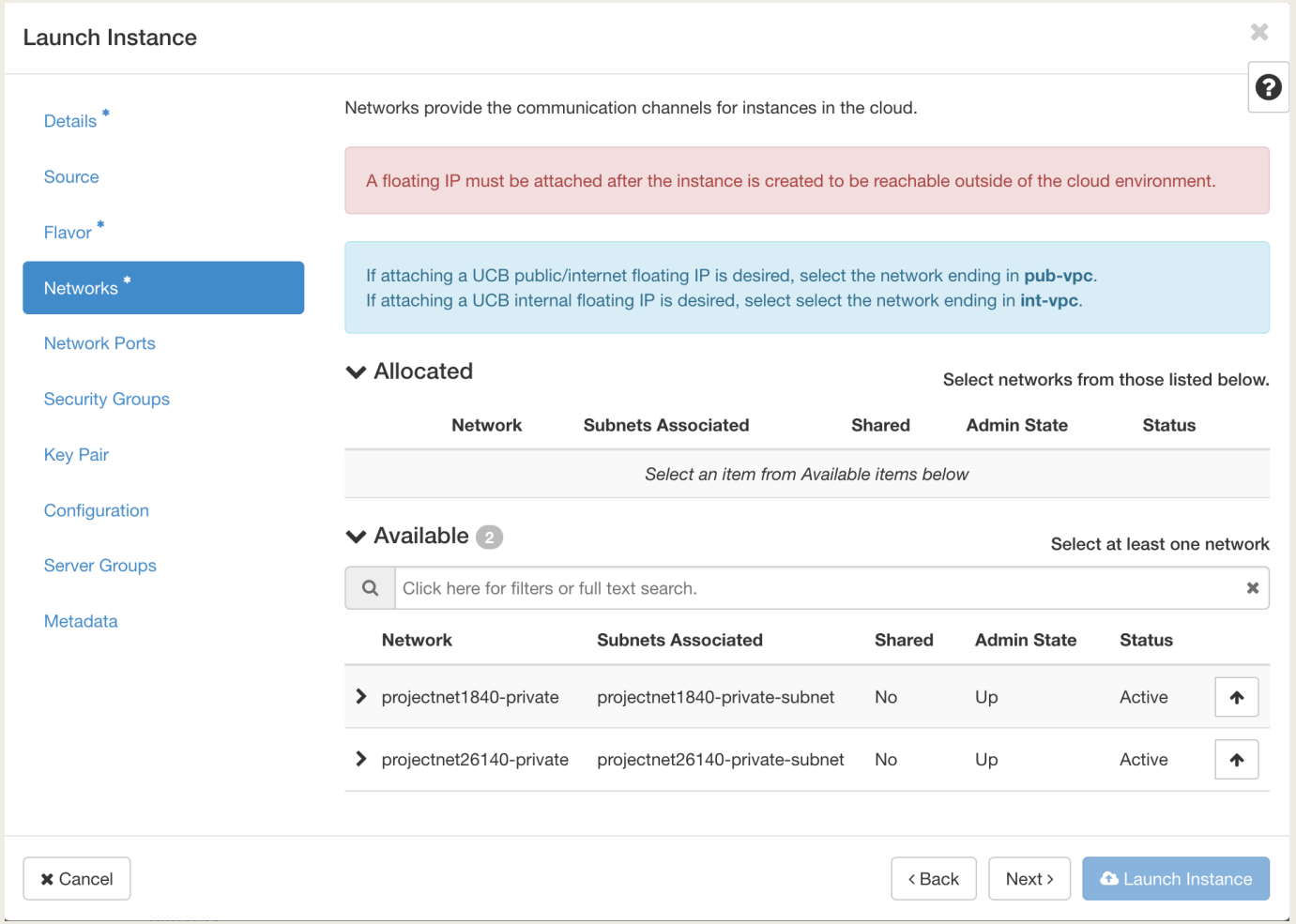
Flavor

- Choose from a list of pre-selected resources:
  - *A flavor defines the compute and memory capacity of our instance.*
- One CUmulus, we have a maximum allowed flavor of:
  - *24 vCPUs*
  - *96GB RAM*



Networks & Network Ports

- Select a project network, which determines **routability of either a public/internet or campus/internal floating IP.**
  - *We'll choose an external network:*
    - **Projectnet26140-private**
- Ports provide extra communication channels to your instances
- You can select ports instead of networks or a mix of both



Networks: An aside

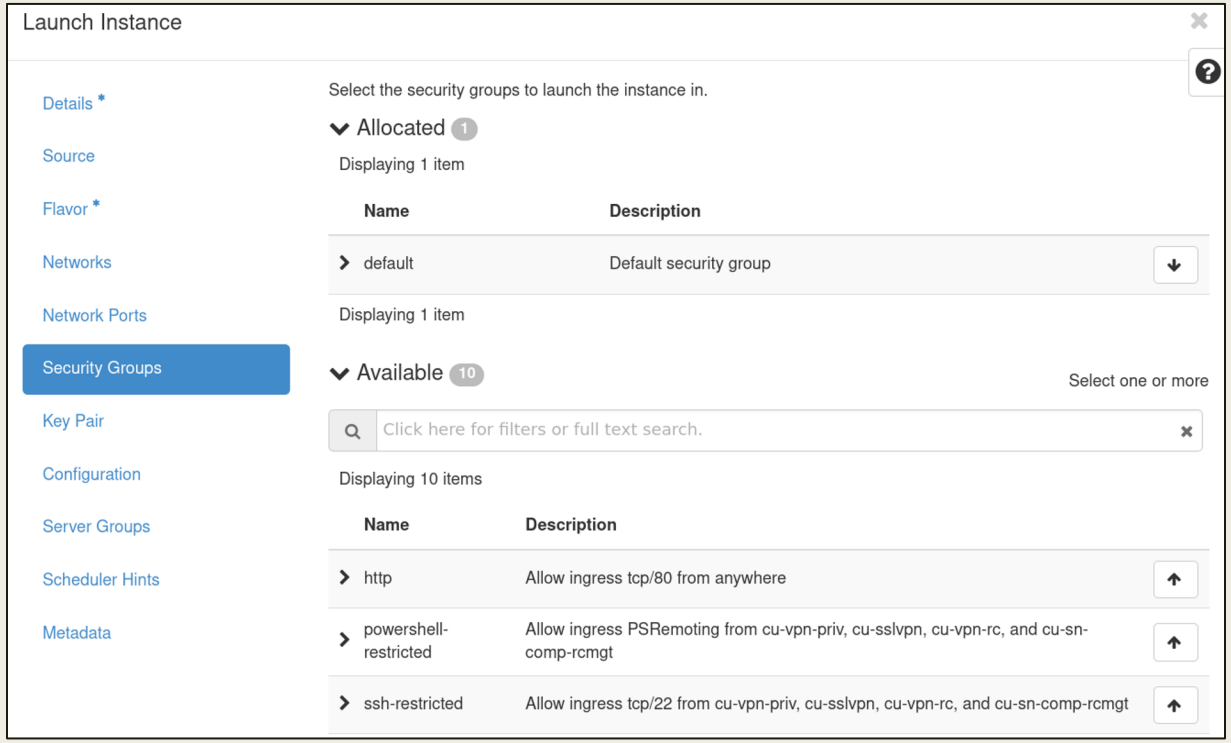
Two network types are provided:

1. **Private Network**- restricts your instance to CU **internal** network
2. **Public Network**- allows certain ingress/egress from the **public** internet (for certain access, e.g http, https)

- Direct shell access (e.g. *ssh*, *powershell*) is restricted to the CU internal network due to multi-factor-authentication requirements
- This means that you will need to be on the CU network to access your instance (either VPN or through RC).

Security Groups

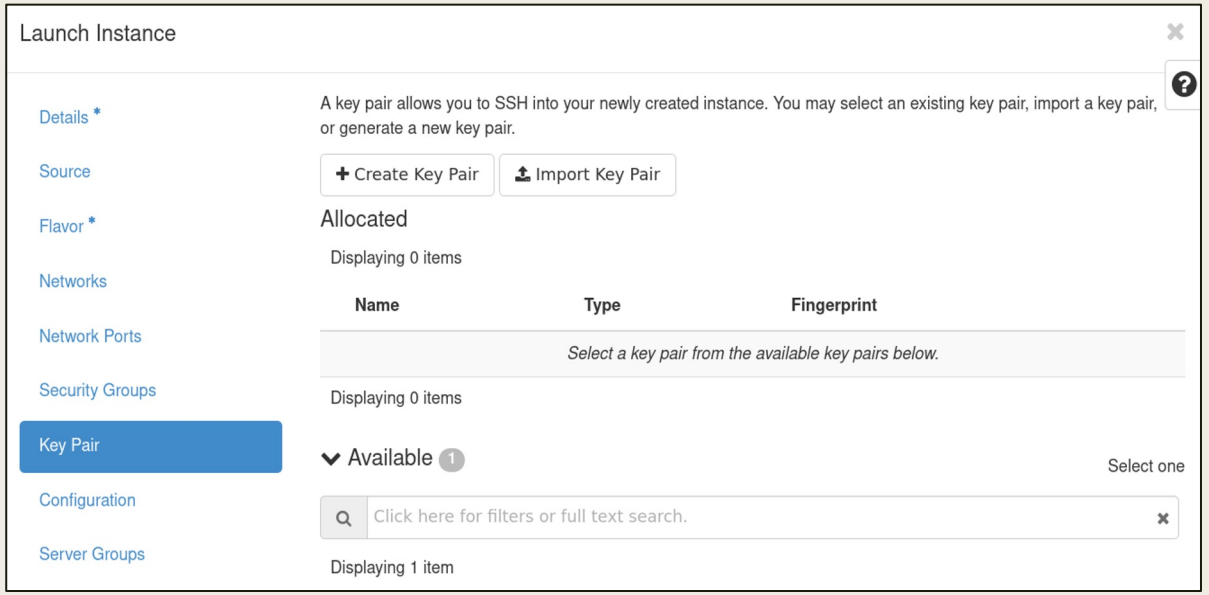
- Security Groups act as a **virtual firewall** for your instance to control inbound and outbound traffic.
- Note: We can create custom security groups if there are specific ports you need opened
- We'll choose ssh-curc, ssh-restricted, http, and https for our demo



Key Pairs

- A key pair allows you to SSH into your new instance

- You may select an existing key pair, or import a key pairI find it easiest to create a keypair on my machine and import it
  - https://www.ssh.com/academy/ssh/public-key-authentication



Key Pairs: An aside

- You will need to create an SSH key *from* or copy it *to* a computer you have CU internal access to:
  - *If you have an RC account already, login as follows from a terminal:*

```
$ ssh <username>@login.rc.colorado.edu
# Where <username> is your identikey
```

Key Pairs: An aside

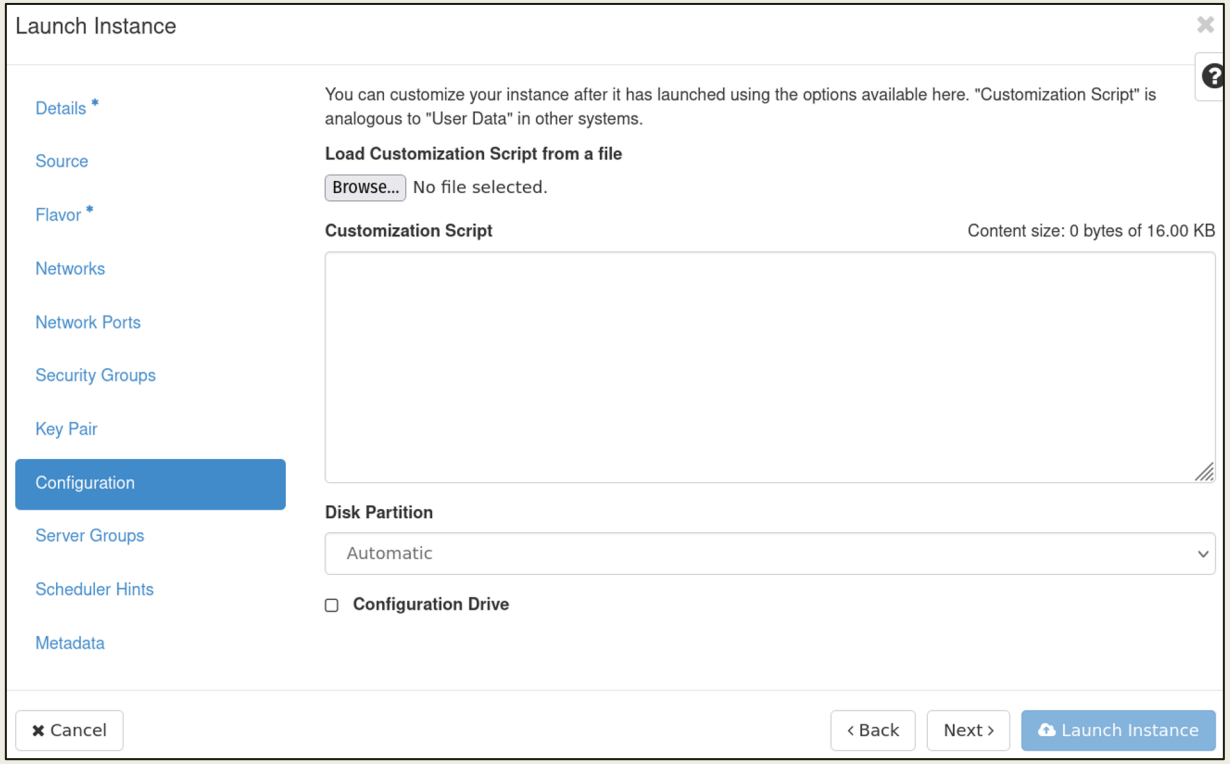
- SSH keys are an access credential that is used in the SSH protocol
- They can be tricky to set up however, so we'll go over a simple example here. From your terminal in a local machine use the ssh-keygen command to create a new ssh keypair:

```
$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/username/.ssh/id_ed25519):
```

- Our new keypairs have been created at `/home/username/.ssh/` and are called `id_ed25519` and `id_ed25519.pub`. The public key (.pub) can be transferred to other remote servers (*this is the key we will import to our CUmulus instance*) but the private key (no suffix) should *never* leave the host machine.

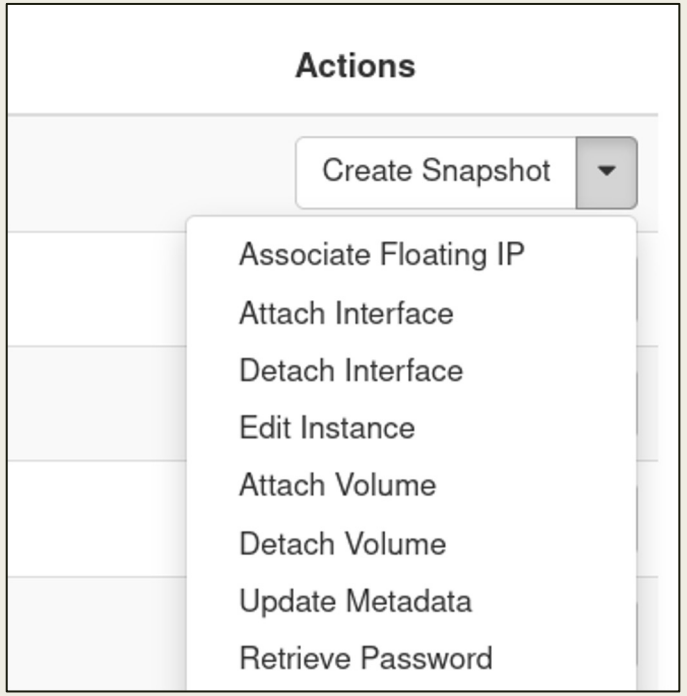
Config, Server Group, Scheduler Hints, and Metadata

- We'll leave these as defaults as they are extra configuration we can provide our instances, but not necessary



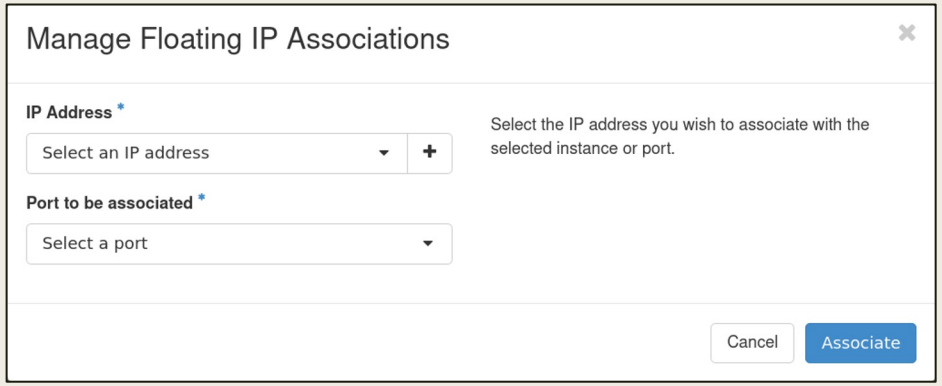
Launch Instance and Associate IP

- Launch instance and wait for it to be set up
- In the meantime, we can associate a Floating IP which will allow us to access the instance from outside of the CU network
  - *On the right-hand side of the newly created instance choose "Associate Floating IP" under the "actions" dropdown*



Associate IP

- Select from available IP addresses
  - *If needed, you can add a floating IP*
- Select port to be associated
  - *This should be pre-populated with the internal IP of your new instance*



Logging into your Instance

Logging In

- You must be on the CU internal to connect via ssh (CURC restriction)
- Open up an ssh connection providing the identity (key) file:

```
$ ssh -i ~/.ssh/<private key> <hostname>@<external floating IP>
```

- For an ubuntu instance this will look something like:

```
$ ssh -i ~/.ssh/testkey ubuntu@123.456.789.123
```

Logged In

- Congratulations! You are now logged into your instance

- You can now:
  - *Install Software*
  - *Administer your instance*
  - *Run applications and jobs*

Demo

Demos:

1. Setting up an Instance
2. CUmulus integration with CURC HPC
3. Mounting a remote filesystem from a CUmulus Virtual Machine

Users that are *not* using tutorial accounts may access CUmulus and your instances for the next week to test and run through tutorials

A few questions for you:

1. What types of applications could you see your users building on CUmulus?

2. What demand do you have for these types of persistent services?

Thank you!

- Contact: anwi7603@colorado.edu
- Help Desk: rc-help@colorado.edu