IIT CS 451
Assignment 4
Due: Novemeber 8th 2024 11:59pm cst/cdt
100 pts

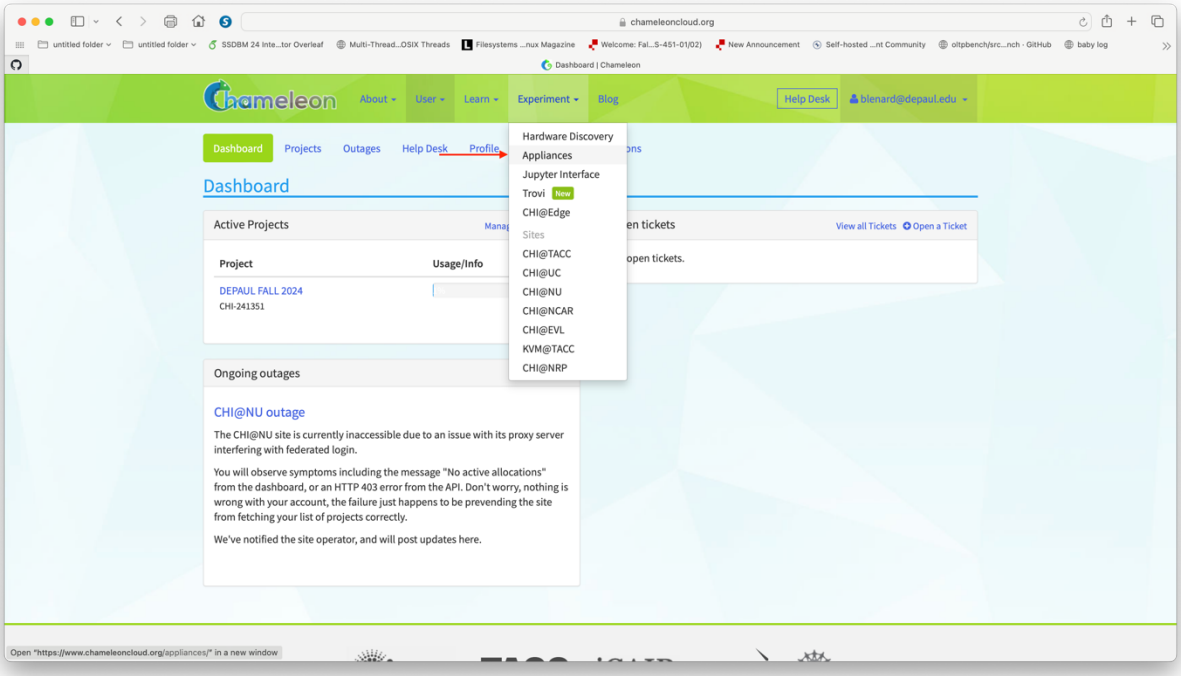
Objective: Build a virtual machine cluster to run mpich programs against.
You will deploy the following vm's in our cloud:

- 3 Compute, one with a public IP
- 1 NFS with 1gb of storage

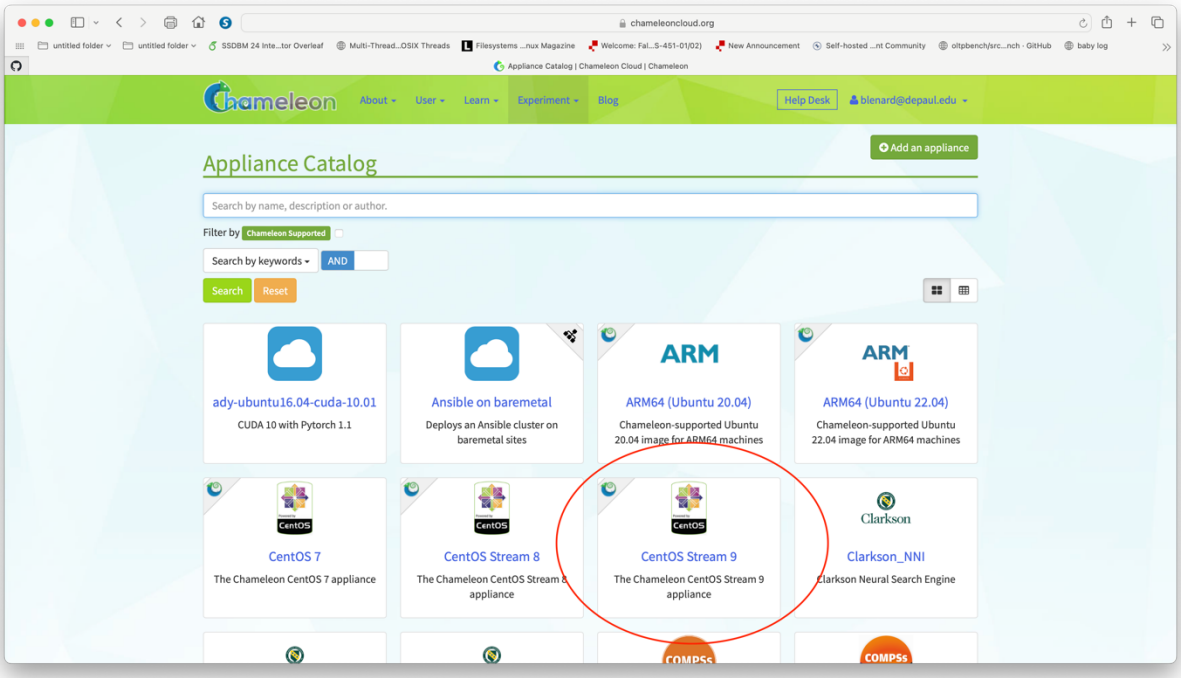
You will successfully setup MPICH, shared storage, as well as SSH keys.
Please shutdown the NFS server last and start it first.
Please turn down the VM's when not used.

Part 1 (50 pts)

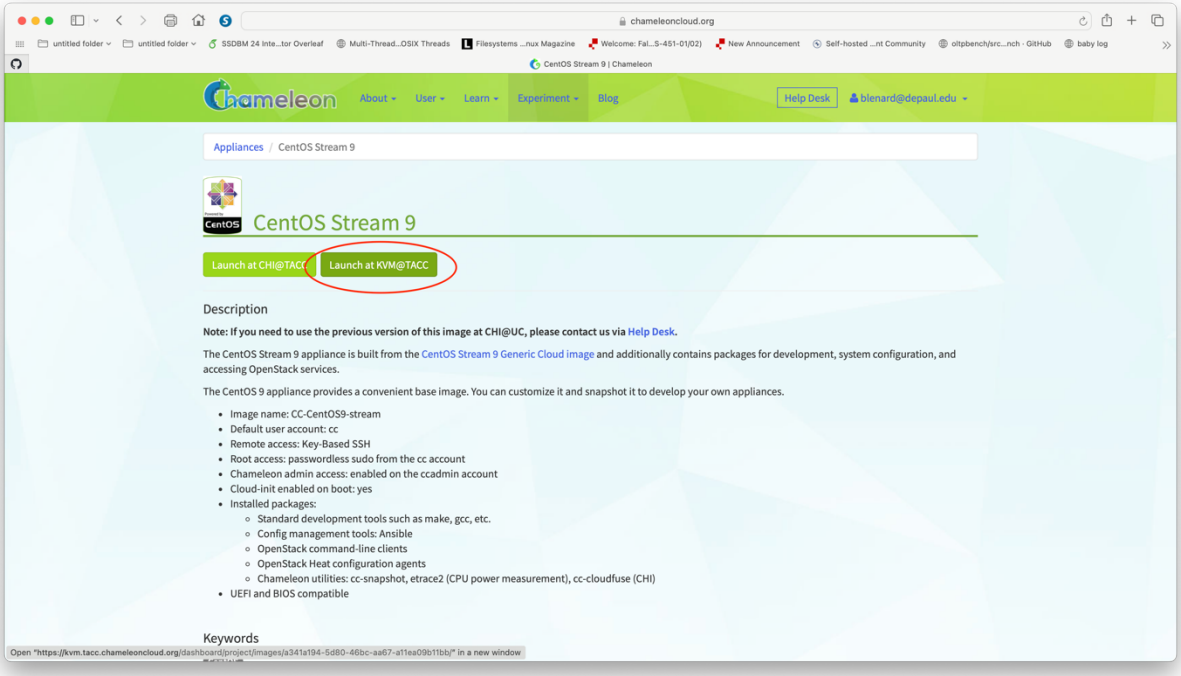
1) Log into https://www.chameleoncloud.org/
2) Click Appliance:



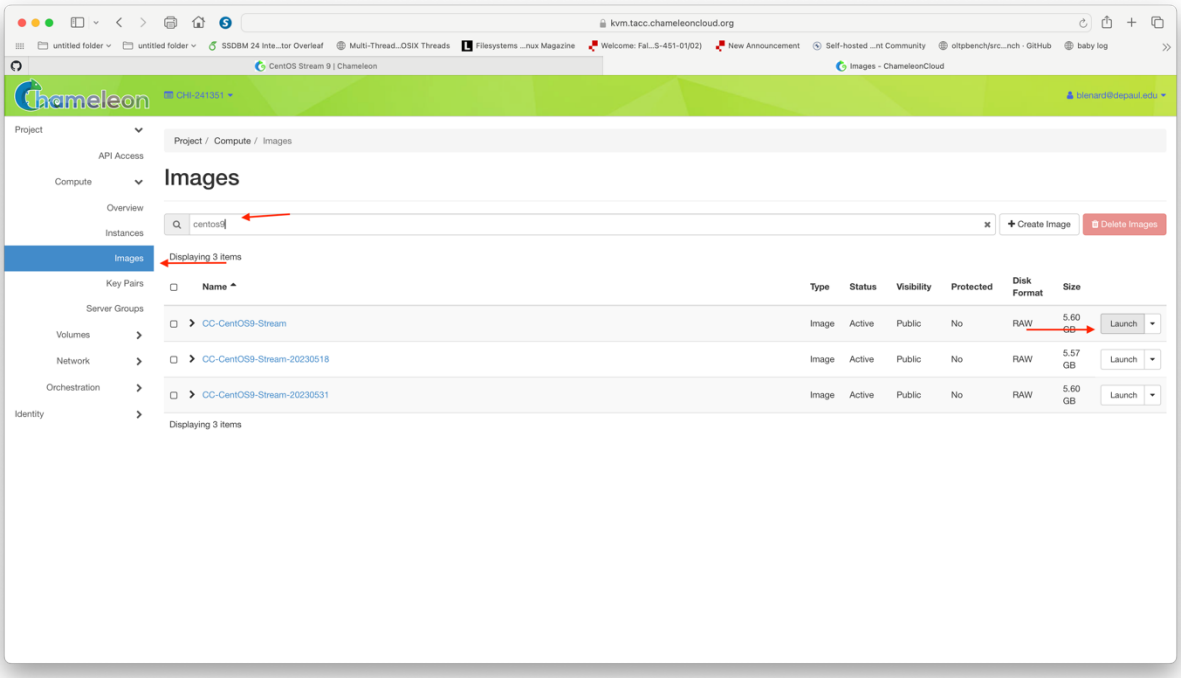
3) Look for CentOS 9 and click:



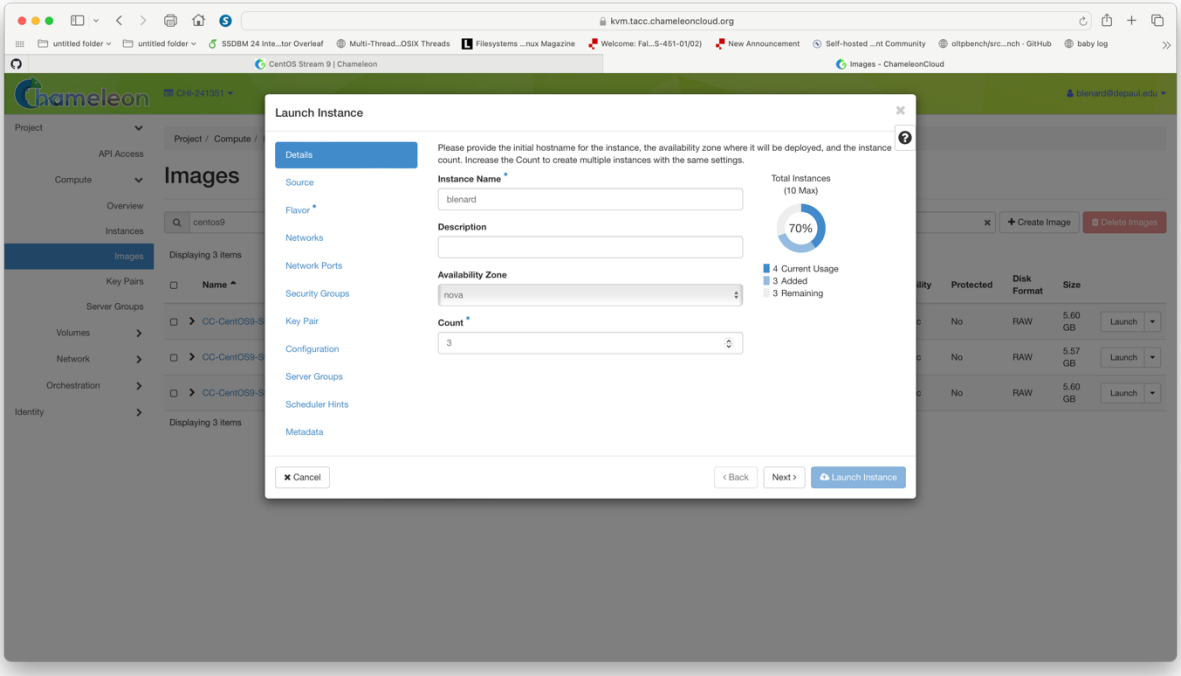
4) Launch KVM@TACC:



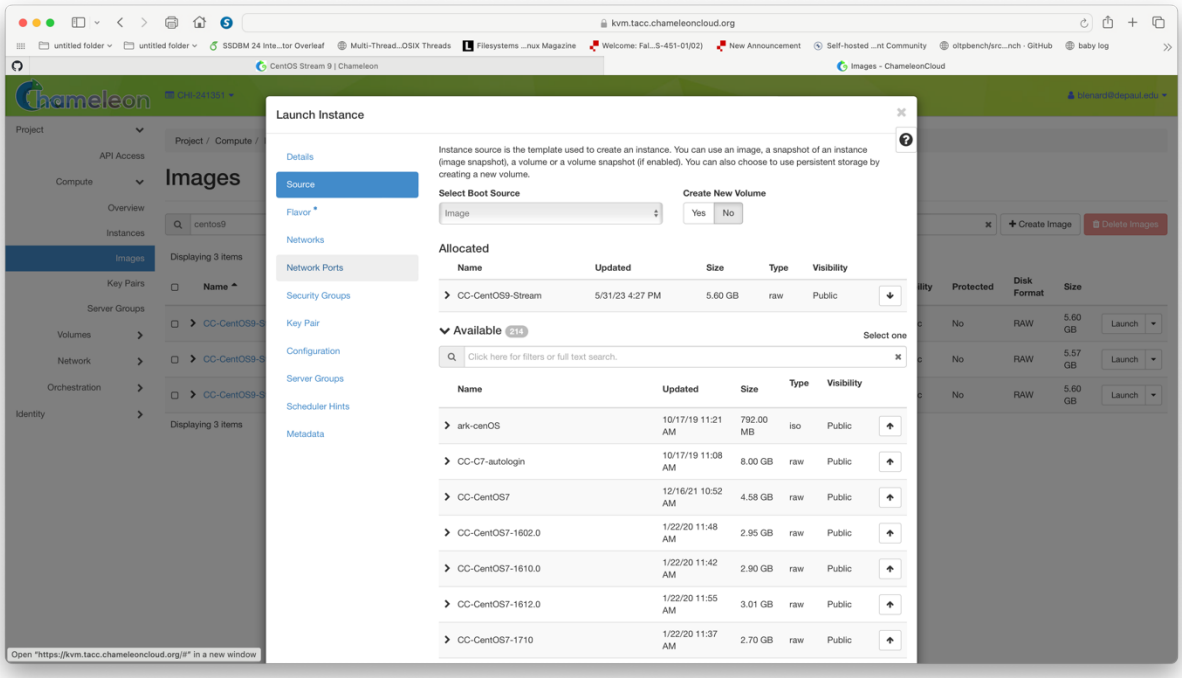
5) Now click Images, then, search for centos9, then launch:



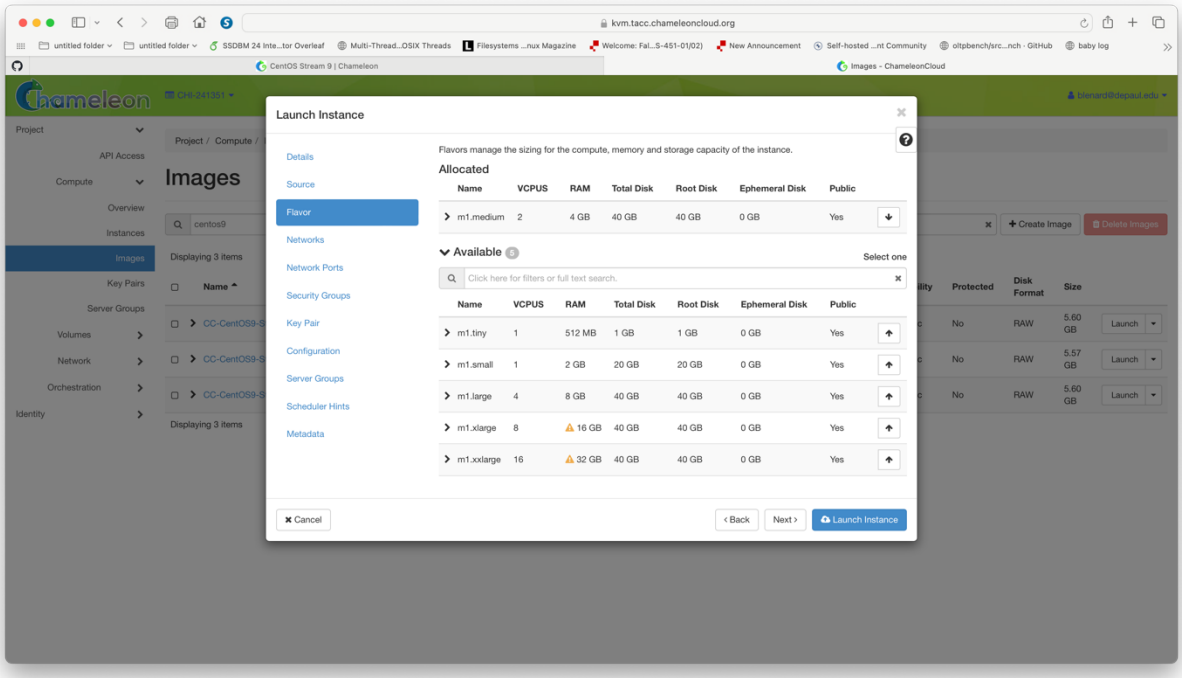
6) Enter your IIT ID as your instance name as seen below:



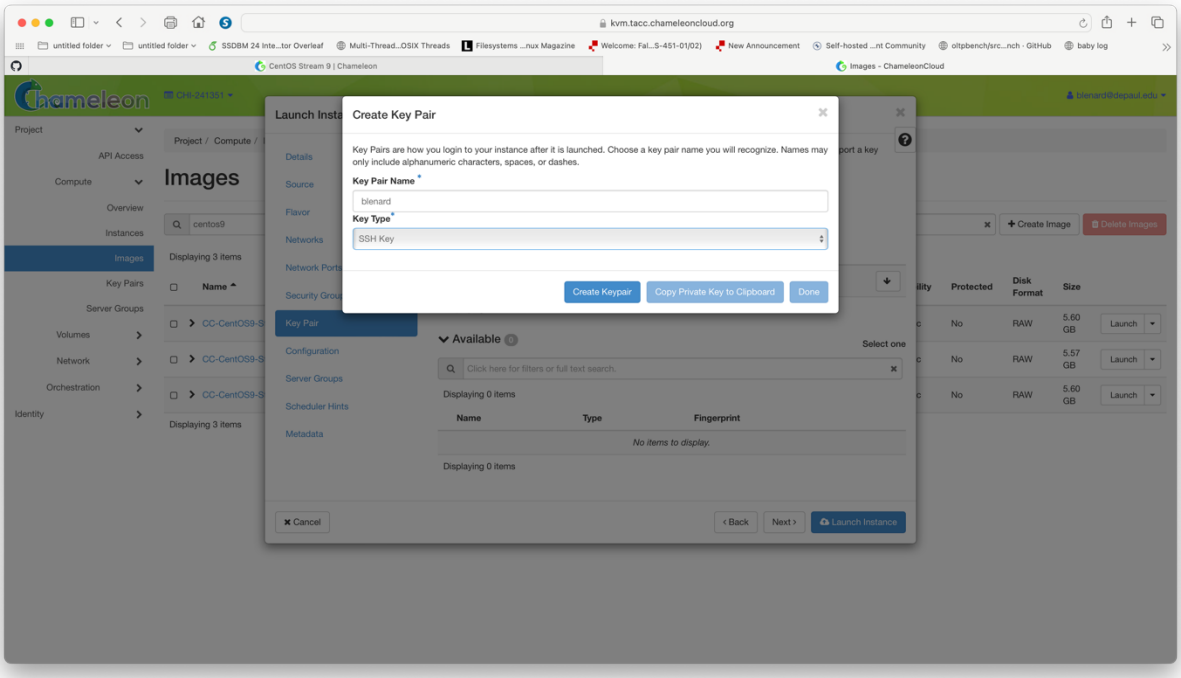
Ensure the source is Centos9



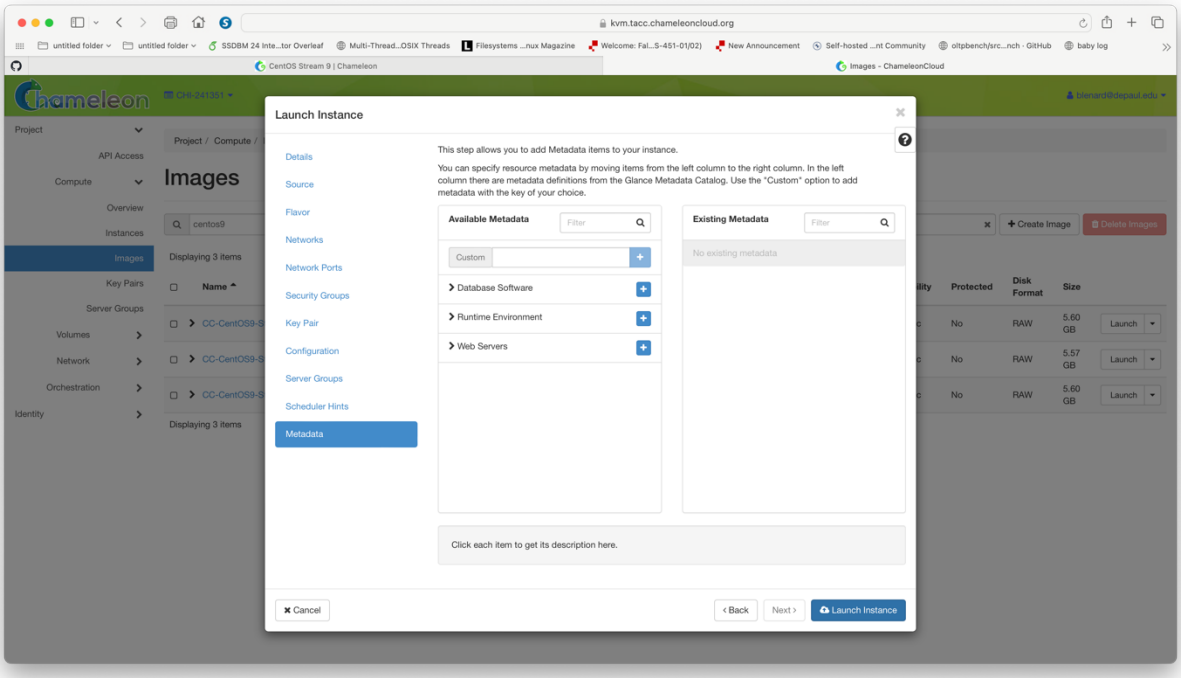
Then Select m1.medium for the size.



Now generate your key pair with your IIT id and download your key:

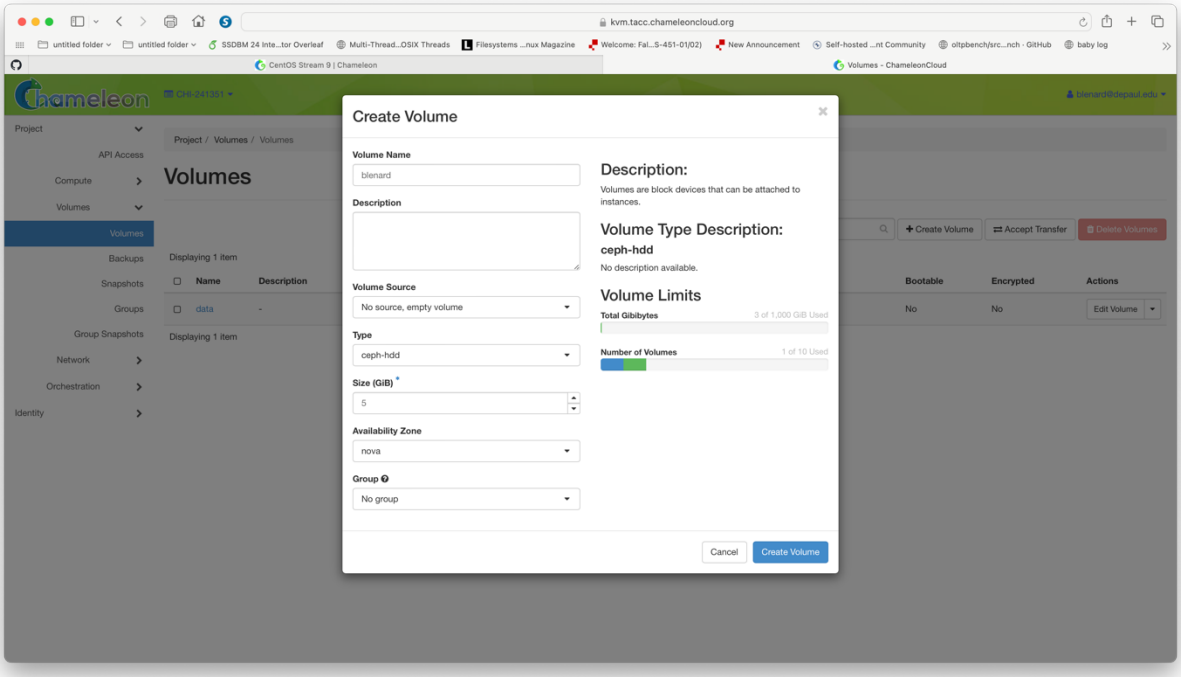


Now hit deploy

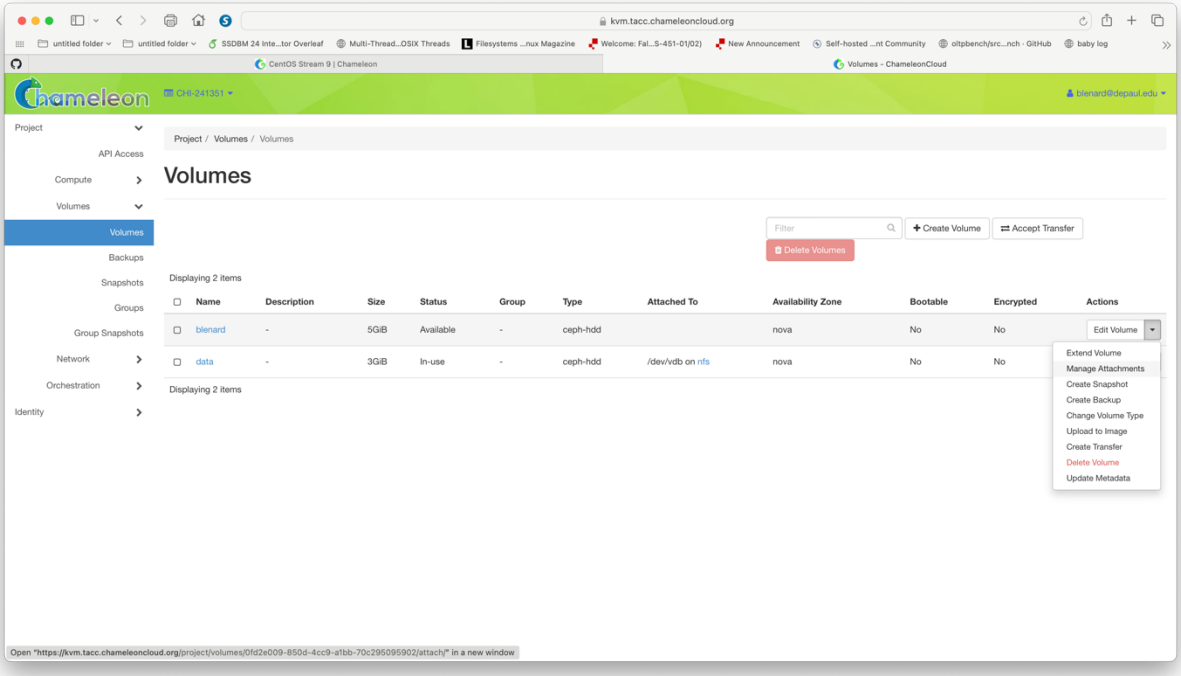


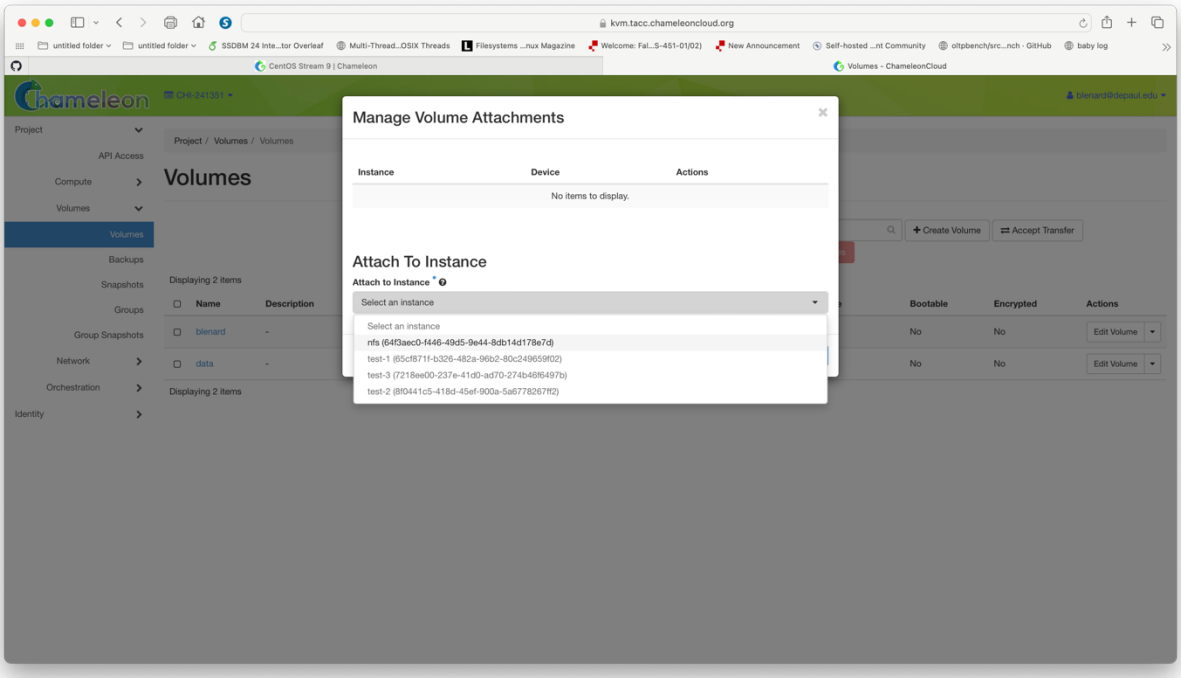
7) Now do the same again, and call it nfs-IITID

8) Now create a volume of 1gb for the nfs server:

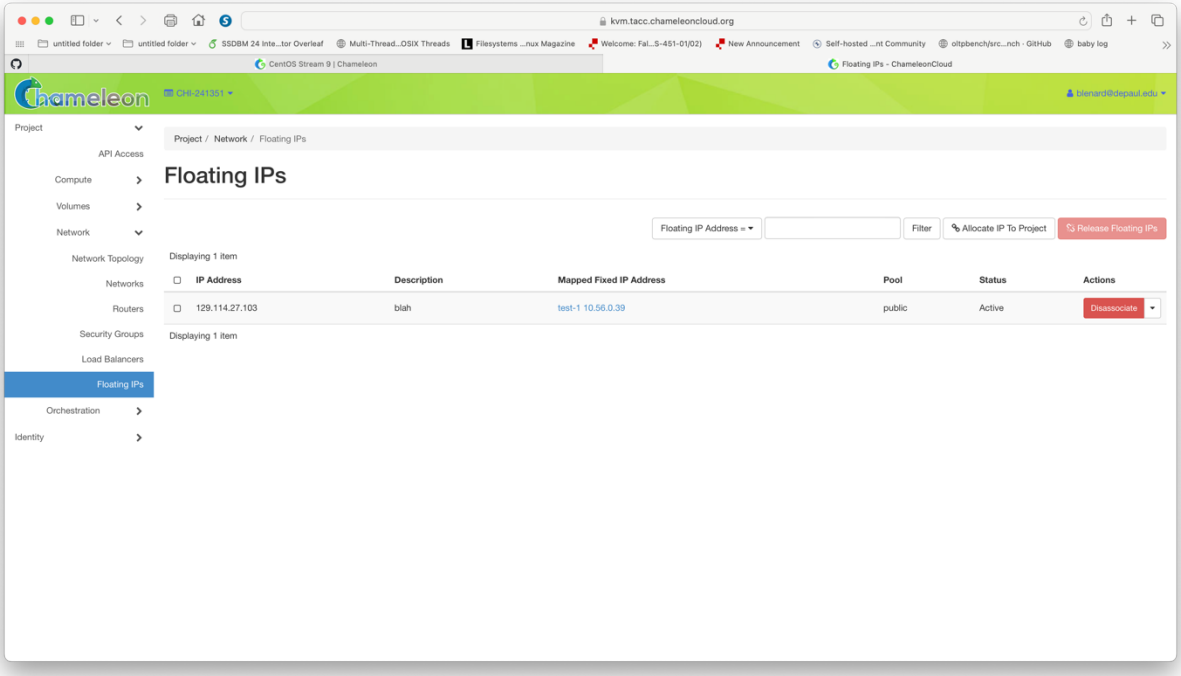


And attach it to your NFS server by clicking manage attachment then find your new NFS server:

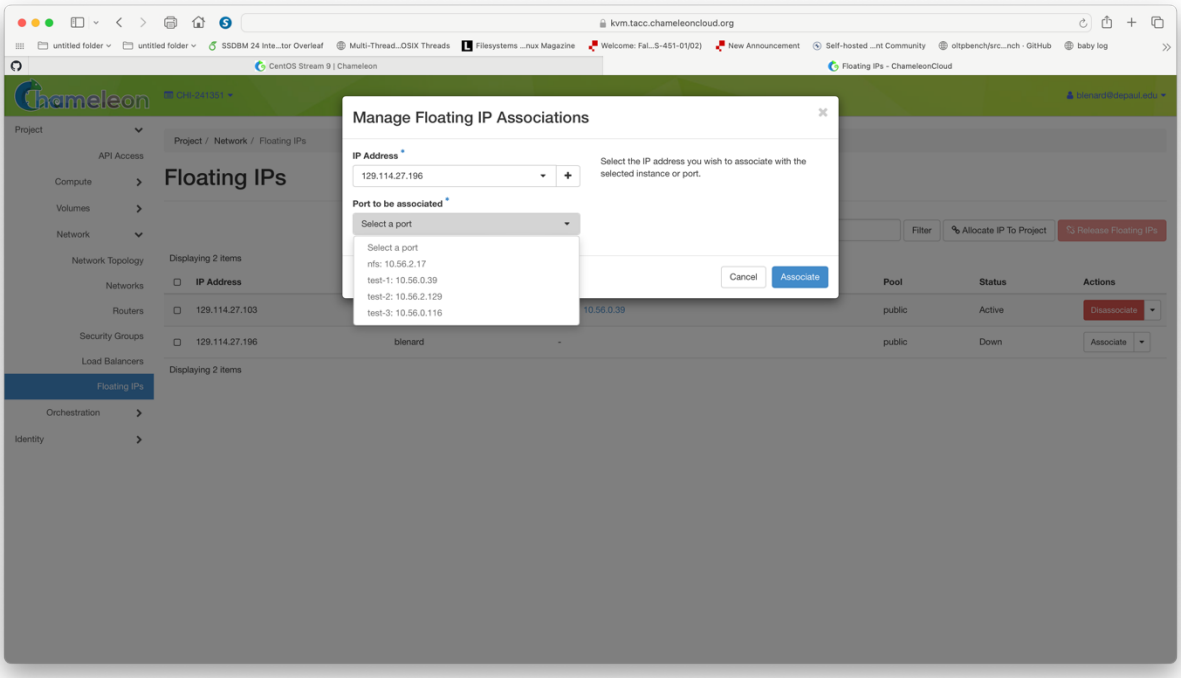




9) Now assign a floating routable IP to your cluster:



Then click assign and pick one of your VM's



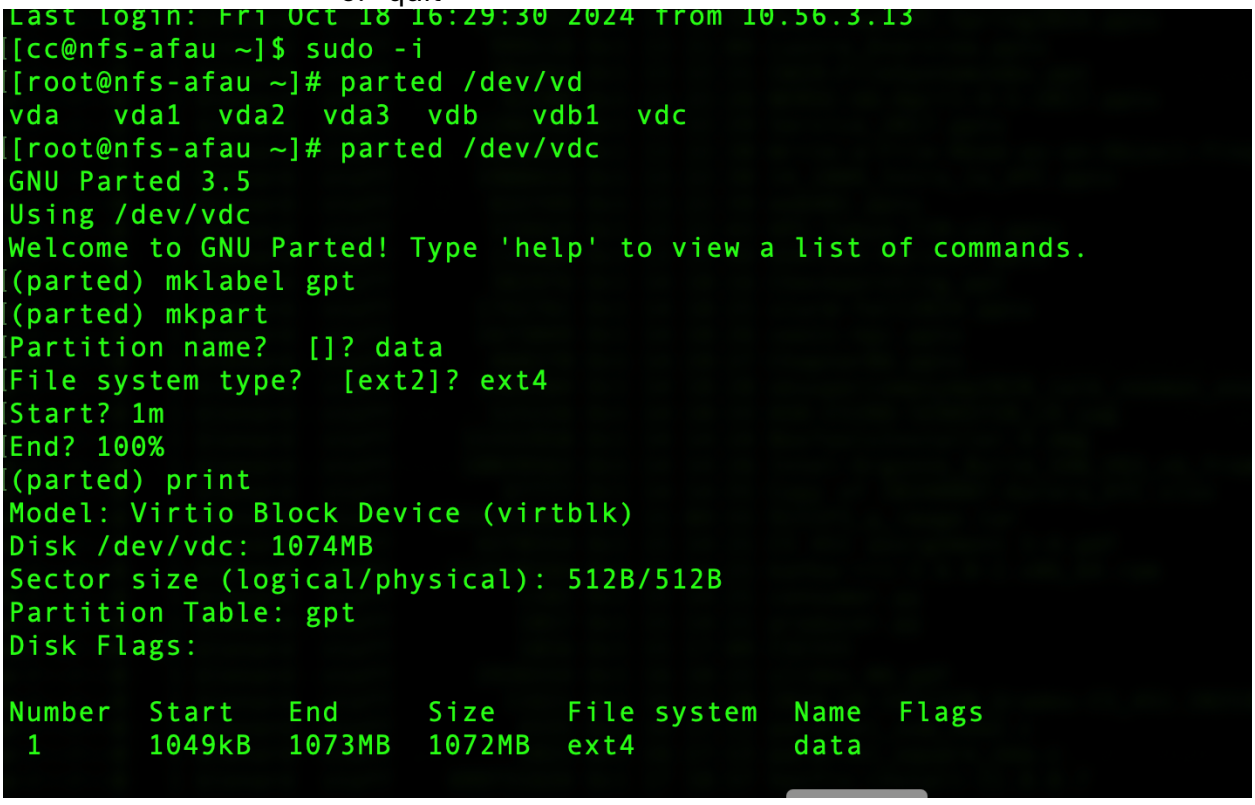
10) Now ssh in a set it up – your key might be else where but you should get the idea
    a. ssh -i Downloads/kvm-test.pem cc@129.114.27.103
    b. 129…. Is the floating ip I was assigned
    c. Put your key on this VM under .ssh as id_rsa and chmod 600 to it
        i. To do that, ssh to the vm with floating IP
        ii. vi (or emacs or nano,etc) .ssh/id_rsa
        iii. Paste the contents of the private key you downloaded and save it
        iv. Now, enter chmod 600 .ssh/id_rsa
11) Fixing /etc/hosts
    a. **My cluster is called test; use your IIT username**
    b. On the Instance screen grab the private IP's of your 4 vms, and write them down
    c. On the vm, type sudo -i
    d. Now edit /etc/hosts and add the 4 vm's you created, ie like mine:

```
[root@test-1 ~]# cat  /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
10.56.0.39  test-1.novalocal
10.56.2.129 test-2.novalocal
10.56.0.116 test-3.novalocal
10.56.2.17  nfs-test.novalocal
[root@test-1 ~]#
```

12) Also as root, type:

a. yum -y install mpich mpich-devel
b. mkdir /data
c. systemctl disable –now firewalld

13) do steps 11 and 12 on the other 3 hosts, by going from the one you logged in to with the public address
14) Go to the nfs server via ssh
    a. You should see /dev/vdb when you do a ls; ie. **ls /dev/vdb** – stop if you don't
    b. Now type
        i. parted /dev/vdb
        ii. mklabel gpt
        iii. mkpart
            1. data
            2. ext4
            3. 1m
            4. 100%
            5. print
            6. quit



IGNORE THAT THE PICTURE SAYS VDC

        iv. now you should see /dev/vdb1 when you do a ls – if you don't , stop
        v. mkdir /data
        vi. mkfs.ext4 /dev/vdb1
        vii. edit /etc/fstab
            1. add this to the file

/dev/vdb1 /data ext4 defaults 0 1

2. Save, exit
3. mount -a

c. edit /etc/exports, but insert your **VM ips** like this:

```
[root@nfs ~]# cat /etc/exports
/data 10.56.0.116(rw,no_root_squash)
/data 10.56.2.129(rw,no_root_squash)
/data 10.56.0.39(rw,no_root_squash)
```

d. systemctl enable --now rpcbind nfs-server
e. chown cc:cc /data
f. mkdir /data/code; chown cc:cc /data/code

15) Edit the compute vm's, non-nfs to mount the NFS mount – do these steps on all 3
    a. mkdir /data
    b. edit /etc/fstab
        i. add where nfs is your nfs server name
            nfs.novalocal:/data /data nfs defaults 0 0
    c. mount -a
16) Lets run our first MPICH program
    a. As cc, cd /data/code on a non-nfs vm
    b. Put the sample c file there.
    c. Edit your hostlist for mpich
        i. Edit the file hosts, add your 3 compute hosts
        ii. Like mine

```
[cc@test-1 code]$ cat host_file
test-1.novalocal
test-2.novalocal
test-3.novalocal
```

17) Compile the c program:
    a. /usr/lib64/mpich/bin/mpicc -o mpi_hello_world mpi_hello_world.c
18) Execute it:
    a. /usr/lib64/mpich/bin/mpirun -n 3 -f host_file ./hello_world
19) You should get:

```
[cc@test-1 code]$ /usr/lib64/mpich/bin/mpirun -n 3 -f host_file ./hello_world
Hello world from processor test-1.novalocal, rank 0 out of 3 processors
Hello world from processor test-3.novalocal, rank 2 out of 3 processors
Hello world from processor test-2.novalocal, rank 1 out of 3 processors
```

Part B (50 pts – 12.5 each)

1. Adjust hello world so that after you type in your name once, when prompted by the manager node, every node salutes you, using the name you typed in.
2. We measure the wall clock time using *time* mpirun in the broadcasting of an array of doubles. To avoid typing in the dimension *n*, either define *n* as a constant in the

program or redirect the input from a file that contains *n*. For increasing number of processes and *n*, investigate how the wall clock time grows. Run N as 2,4,8,16,32

3. Compile and run both static_loaddist.c and dynamic_loaddist.c and compare the runtimes. Run N as 2,4,8,16,32
4. Adjust parallel_sum.c to go to 200 instead of 100 and adjust it to work for *p* processors where the dimension *n* of the array is a multiple of *p*. In other words, so if you have N = 3 or 6 , it'll work correctly.

What to turn in:

1) Screen shot of the apps working for part b. Screenshot of the hello world working in part a.
2) Write up answer the following – a few sentences each:
    a. What was the purpose of the NFS server? What happens if you run hello world from a non-shared spot?
    b. How are the different instances of the same program communicating?
    c. How might the interconnect effect the speed of a MPI program?