Module 4
Sharing and Scaling a VM

George Mihaiescu

Bioinformatics on Big Data: Computing on the Human Genome

September 29 – September 30, 2016

bioinformatics.ca

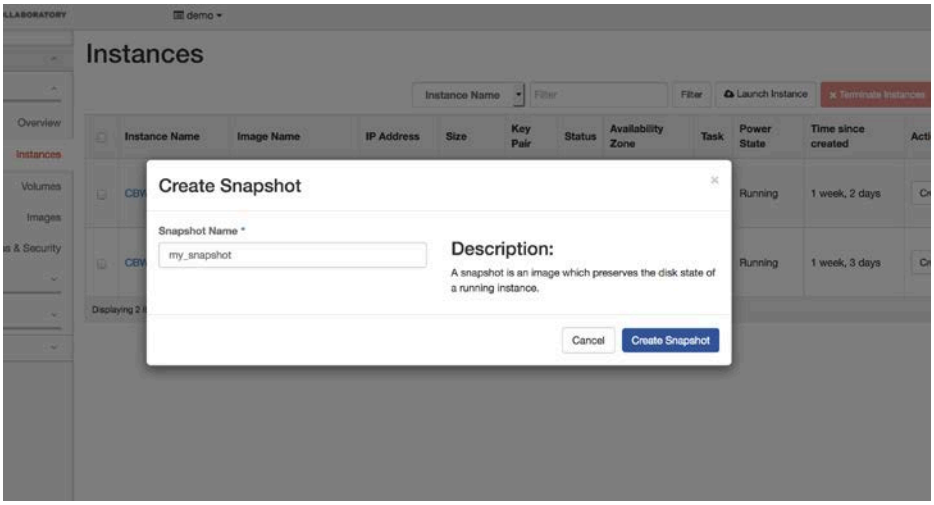
Cancer Genome COLLABORATORY

Cloud computing for collaborative research

Learning Objectives of Module

- Participants will gain practical experience and skills to be able to:
  - Freeze a VM
  - Share a VM with external partners
  - Launch a shared VM
  - Scale your VM to meet your analysis needs

What a snapshot is



Considerations when snapshotting a VM

Clean up confidential data – SSH key

rm ~/.ssh/authorized_keys

Clean up confidential data – saved credentials

Clean up confidential data – bash history

```
141  docker attach -it 76e33f3c3065f869e4642382 5f075423497e5644fefa5d4c357524bf5d457855 bash
142  docker exec -it 76e33f3c3065f869e4642382 5f075423497e5644fefa5d4c357524bf5d457855 bash
143  ll
144  cd /var/www/html/
145  ll
146  cd samples/
147  ll
148  mv 1e00b172-37aa-4061-801f-95dc013337d7 1e00b17237aa4061801f95dc013337d7
149  ll
150  mv 8a5018ed-47e2-46ce-b6e1-6df40eb77f26 906b7def6e49493bb7c58c5e67534dd7
151  ll
152  cd ..
153  ll
154  service nginx restart
155  cd samples/
156  docker run  -e ACCESSTOKEN=Secret icgc/icgc-storage-client bin/icgc-storage-client --profile collab download --object-id 6d89e978-34f6-5074-b30e-01b7203fcbb3
--output-dir /tmp
157  history
```

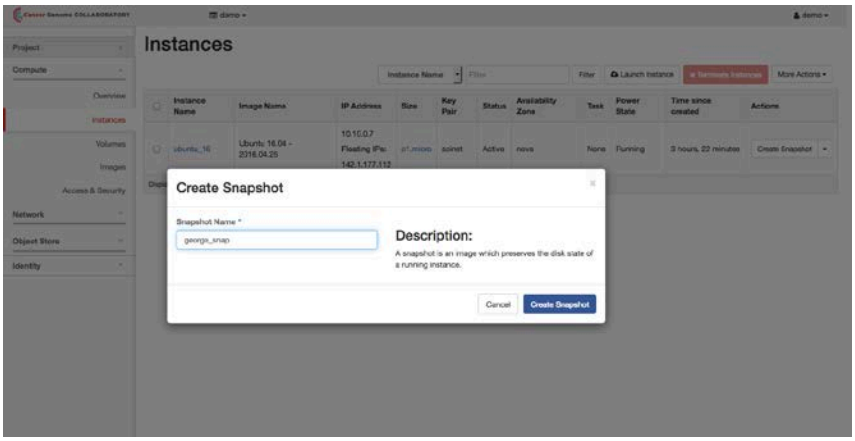
history –c
rm ~/.bash_history

Keep the image size small

Plan your image

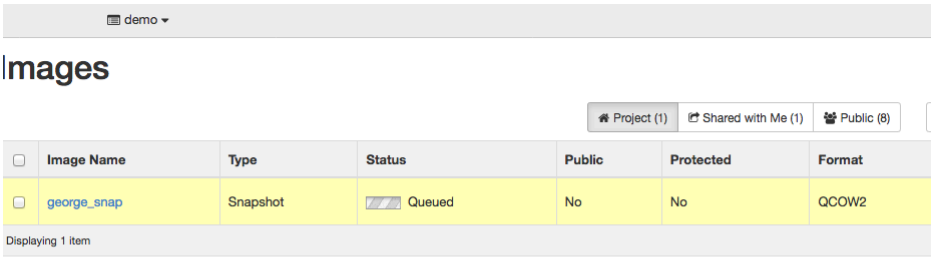
- Try to envision how you are going to use the image you are creating, and prepare accordingly
- If the workflow needs a large reference dataset, it might be better to store this dataset on a web or FTP server instead of having it inside the image
- You could add a script that executes the first time the virtual machine starts and installs all security updates available
- Proper planning will save you time and disk space($) later on

Taking a snapshot

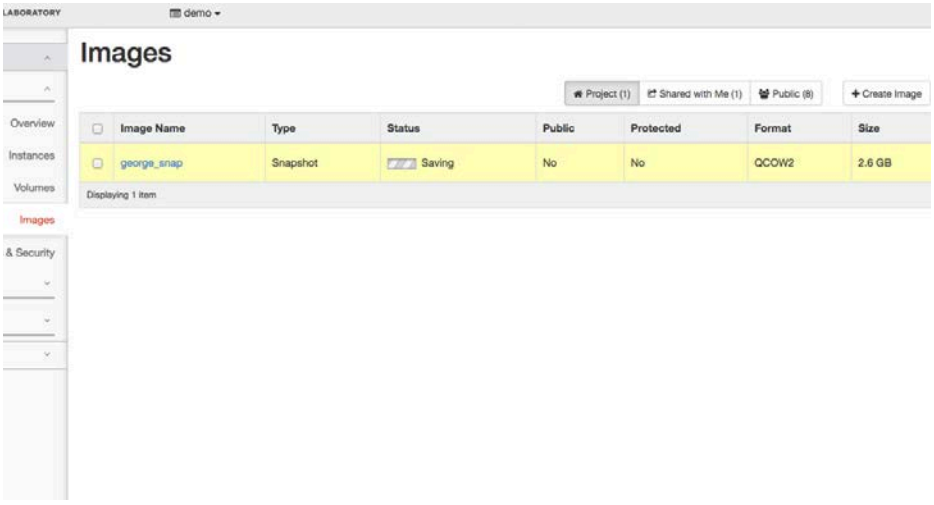


Taking a snapshot

Taking a snapshot -stages



Taking a snapshot -stages



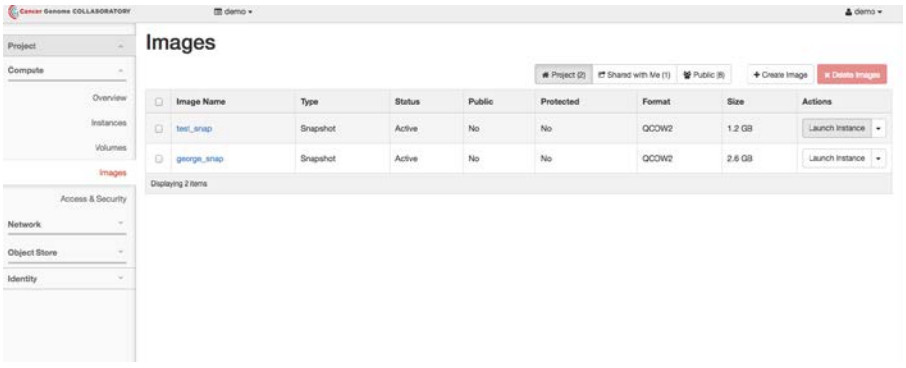
If you removed the SSH key…

Share the new image with another tenant/project

- A common use case in a shared cloud environment is to share a customized image with other projects and users.
- Tenant A could have two users called John and Mary. John works on a new algorithm and Mary can see his virtual machine listed in Dashboard, terminate it, restart it, etc.
- At the completion of his work, John can take a snapshot of his customized instance and share this new image/snapshot with Paul who is a user from tenant B.
- Other users from tenants C and D that have accounts in the same cloud provider will not have access to this image/snapshot that was shared between A and B.

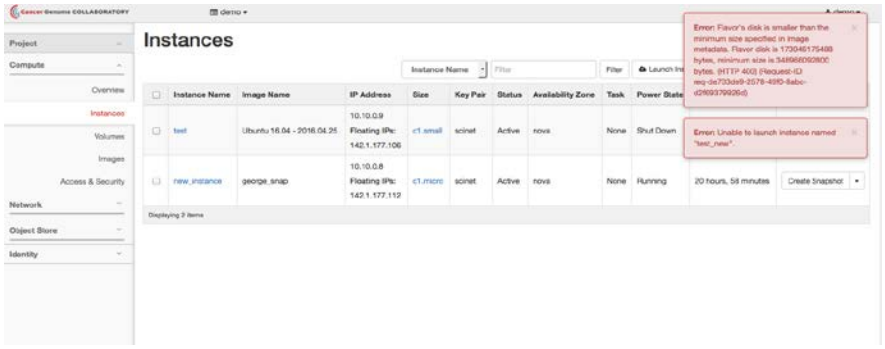
Launching a new instance from a snapshot

Once you have a customized image that you snapshotted, this can be used to repeatedly recreate the exact environment, as well as scale out your analysis.



Launching a new instance from snapshot considerations

- If you try to start an instance from a snapshot by choosing a flavor smaller than the original instance it will result in an error message.



Scale out planning exercise

Scenario:

- You just created a workflow that performs VCF, and you would like to run it on 100 samples.
- One analysis takes about 24 hours on a VM with 4 cores and 16 GB of RAM.
- You have enough budget to run 100 cores for 72 hours.
- You also have a CPU quota allowing you to use a maximum of 100 cores at any given time.
- 85 of the samples have a size around 180 GB (small), but the other 15 need around 310 GB of disk space (large)

Scale out planning exercise

Your cloud provider offers the following instance flavors:

Flavor name	Specs
c1.micro	1 core, 8 GB RAM, 100 GB disk
c1.small	2 cores, 16 GB RAM, 200 GB disk
c1.medium	4 cores, 24 GB RAM, 250 GB disk
c1.large	6 cores, 28 GB RAM, 280 GB disk
c1.jumbo	8 cores, 32 GB RAM, 320 GB disk
c1.xlarge	12 cores, 36 GB RAM, 400 GB disk

How would you organize your VMs to perform the analysis in three days, staying within the budget?

Scale out planning exercise – one possible solution

- On an external server, set up a simple Python web server that has two sets of samples: under 200 GB and over 300 GB.
- The web server accepts HTTP requests on four URL paths (/sample_small, /sample_large, /sample_small_done, /sample_large_done).
- Start 49 VMs using the c1.small flavor (2 cores, 16 GB RAM and 200 GB disk), and use Ansible to orchestrate their workload.

Scale out planning exercise

- At the beginning of the workflow, each VM would do a "GET / sample_small" request to the web server containing its IP address and receive back a JSON document containing the sample ID it should analyze.
- The web server will mark that sample as "in progress" along with the IP address of the VM working on it, and the timestamp.
- At the end of the workload, the VM should send a "POST / sample_small_done" request to the external web server with the IP of the instance, and the sample analyzed.

Scale out planning exercise

- After 24 hours, you should have 49 small samples analyzed and budget for 48 more hours.
- Repeat the same process using 34 VMs of c1.small flavor in order to continue analyzing the small samples, using 68 cores out of your quota.
- Start four c1.jumbo that will send POST requests to "/ sample_large" in order to receive large sample IDs, using 32 cores out of your quota.
- At the end of day two, you should have 83 average samples analyzed and four large ones.

Scale out planning exercise

- Repeat the same process using 2 VMs of c1.small flavor in order to finish the small samples, using 4 cores out of your quota.
- Start 11 c1.jumbo that will send POST requests to "/sample_large" in order to receive large sample IDs, using 88 cores out of your quota.
- At the end of day three, you should have all samples analyzed and still have some unused budget for re-running any samples that failed.

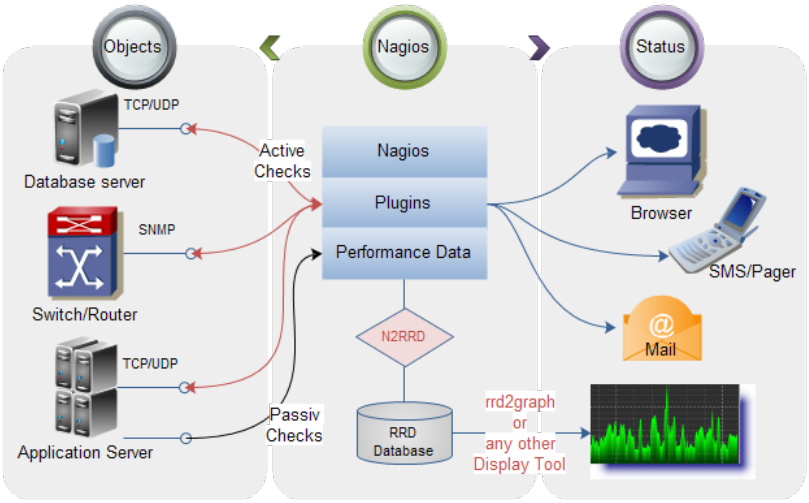
Considerations when running large analysis - size

Start small …

and increase your VM fleet slowly as you gain confidence in your process.

Mistakes can be costly

Monitoring



Dependencies

- Minimize external dependencies (e.g. provide a reference dataset on a server in the same cloud environment, for better performance).

Failure domains