



Storage options for cloud computing

WestDRI Webinar

Nov 9, 2022

Sarah Huber, University of Victoria

Outline



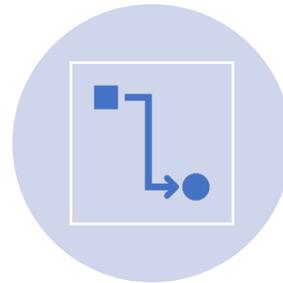
Brief cloud orientation



Options for storage in the cloud



Common use cases for each storage type



Basic usage demos

Getting started on the cloud



- the Alliance has multiple Infrastructure-as-a-Service (IaaS) clouds
 - You (or PI) can [apply](#) for a cloud project with a quota of resources
 - Networks, “flavors” and various other options are provided
 - You setup and manage your own virtual machines (VMs), including storage
 - Lots of terminology specific to cloud management software OpenStack
 - e.g. VMs vs. instances
- Our clouds:
 - Arbutus
 - Cluster sites, including Béluga, Graham, and Cedar

OpenStack Horizon - Dashboard

Project / Compute / Overview

Project / Compute / Overview

Project

API Access

Compute

Overview

Instances

Images

Key Pairs

Server Groups

Volumes

Network

Orchestration

Object Store

Share

Identity

Overview

Limit Summary

Compute

 Instances Used 2 of 10	 VCPUs Used 8 of 10	 RAM Used 21GB of 30GB
--	--	---

Volume

 Volumes Used 5 of 15	 Volume Snapshots Used 2 of 10	 Volume Storage Used 880GB of 1000GB
--	---	---

Network

 Floating IPs Allocated 1 of 2	 Security Groups Used 4 of 10	 Security Group Rules Used 12 of 100	 Networks Used 1 of 1	 Ports Used 8 of 110	 Routers Used 1 of 1
---	--	---	--	---	---

instance <-> running VM

Usage Summary

Select a period of time to query its usage:
The date should be in YYYY-MM-DD format.

Prerequisite cloud usage information



How to set up a virtual machine, including choosing

image (operating system)

security rules

access, including network access + ssh key

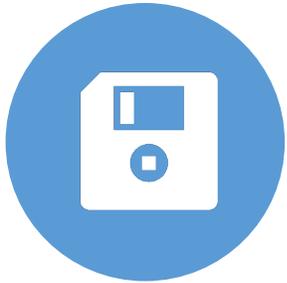
https://docs.alliancecan.ca/wiki/Cloud_Quick_Start



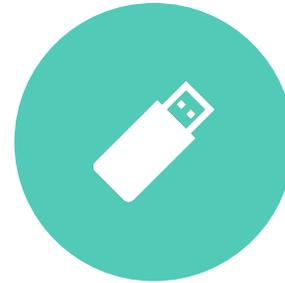
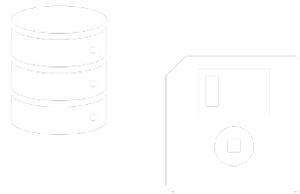
Security best practices:

https://docs.alliancecan.ca/wiki/Security_considerations_when_running_a_VM

Cloud storage types



Ephemeral disk storage



Volume storage



Shared file system
storage



Object storage



Ephemeral disk

- Linked to a single instance
- Destroyed when instance dies, any back up must be enabled by user
- Use cases:
 - Storage for operating system if launched from image
 - common with “compute” flavors
 - may outperform root stored in a volume



Volume storage

- Can be attached to different VMs, like a USB stick
- Mounted on a single VM at a time, like a USB stick
- Hardware redundancy, unlike a USB stick
- Not backed up by default
- Designed as “block” storage, must be formatted to use for a file system
- Use case:
 - Data that only needs to be used on a single VM at once
 - Storage for operating system/root of persistent VM

Creating a new volume

Create Volume

Volume Name

Description

Volume Source

No source, empty volume

Type

Default

Size (GiB) *

1

Availability Zone

nova

Group ?

No group

Description:
Volumes are block devices that can be attached to instances.

Volume Type Description:
Default
No description available.

Volume Limits

Total Gibibytes 840 of 1,000 GiB Used

Number of Volumes 4 of 15 Used

Project / Volumes

API Access

Compute >

Volumes >

Volumes

Backups

Snapshots

Groups

Group Snapshots

Network >

Orchestration >

Object Store >

Share >

Identity >

Displaying 4 items

Name

f248fb47-ea...

21a1cd8a-5...

root volume

5f2f7e7f-f35...

Displaying 4 items

Create Volume

Accept Transfer

Delete Volumes

Availability Zone	Bootable	Encrypted	Actions
Yes	No		Edit Volume
Yes	No		Edit Volume
Yes	No		Edit Volume
Yes	No		Edit Volume

Attaching a volume

Project / Volumes / Volumes

Volumes

Filter [+ Create Volume](#) [≡ Accept Transfer](#) [Delete Volumes](#)

Displaying 5 items

<input type="checkbox"/>	Name	Description	Size	Status	Group	Type	Attached To	Availability Zone	Bootable	Encrypted	Actions
<input type="checkbox"/>	newexample	-	200GiB	In-use	-	Default	/dev/vdd on storageexample	nova	No	No	Edit Volume
<input type="checkbox"/>	example	-	100GiB	In-use	-	Default	/dev/vdc on storageexample	nova	No	No	<ul style="list-style-type: none">Manage AttachmentsCreate SnapshotCreate BackupChange Volume TypeUpload to ImageUpdate Metadata
<input type="checkbox"/>	21a1cd8a-53d9-41e7-9812-647c120265d1	-	100GiB	Available	-	Default		nova	Yes	No	
<input type="checkbox"/>	root volume	-	200GiB	Available	-	Default		nova	Yes	No	
<input type="checkbox"/>	5f2f7e7f-f358-4cfd-b7e2-72b520cbb664	-	40GiB	In-use	-	Default	/dev/vda on testingvm	nova	Yes	No	Edit Volume

Displaying 5 items

Formatting + file system for volume

```
[ubuntu@storageexample:/$ sudo fdisk /dev/vdc ]
Welcome to fdisk (util-linux 2.37.2).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0xd84c8b2b.

[Command (m for help): n ]
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
[Select (default p): p ]
[Partition number (1-4, default 1): 1 ]
[First sector (2048-209715199, default 2048): ]
[Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-209715199, default 209715199): ]

Created a new partition 1 of type 'Linux' and of size 100 GiB.

[Command (m for help): w ]
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

[ubuntu@storageexample:/$ █ ]

[ubuntu@storageexample:/$ sudo mkfs -t ext4 /dev/vdc ]
mke2fs 1.46.5 (30-Dec-2021)
Found a dos partition table in /dev/vdc
[Proceed anyway? (y,N) y ]
Creating filesystem with 26214400 4k blocks and 6553600 inodes
Filesystem UUID: b37ab441-6116-4fc9-b993-9c26efbf5499
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424, 20480000, 23887872

Allocating group tables: done
Writing inode tables: done
Creating journal (131072 blocks): done
Writing superblocks and filesystem accounting information: done

[ubuntu@storageexample:/$ █ ]
```

Mounting a volume

- Make directory to mount on:

```
sudo mkdir /media/data
```

- Mount to that directory

```
sudo mount /location/of/volume /media/data
```

- *Optional:* manage ownership of mounted directory, e.g. change ownership to user

```
sudo chown -R ubuntu /media/data
```

An example use case

- Solve linear system of equations, $Ax = b$ for multiple large and sparse matrices A , $b = \vec{1}$
- Matrices obtained from SparseSuite
<https://people.engr.tamu.edu/davis/suitesparse.html>
- Want to store many (possibly large) matrix files
- Solve using MUMPS linear solver

An example use case

```
ubuntu@storageexample:/media/data$ cat runall.sh
```

```
#!/bin/bash
```

```
module load mumps-parmetis
```

```
runfile=c_example_read_mm
```

```
logfile=out.txt
```

```
rm $logfile
```

```
for foldername in *; do
```

```
  if [ -d "$foldername" ]; then
```

```
    echo $foldername
```

```
    cd $foldername
```

```
    cd ..
```

```
  } time { ./runfile ${foldername}/${foldername}.mtx >> $logfile;
```

```
  fi
```

```
done
```

```
=====
MUMPS compiled with option -Dparmetis
MUMPS compiled with option -Dptscotch
This MUMPS version includes code for SAVE_RESTORE
This MUMPS version includes code for DIST_RHS
=====
L D L^T Solver for symmetric positive definite matrices
Type of parallelism: Working host

***** ANALYSIS STEP *****

Average density of rows/columns = 3
Ordering based on AMF

Leaving analysis phase with ...
INFOG(1) = 0
INFOG(2) = 0
-- (20) Number of entries in factors (estim.) = 3342
-- (3) Real space for factors (estimated) = 3508
-- (4) Integer space for factors (estimated) = 25652
-- (5) Maximum frontal size (estimated) = 13
-- (6) Number of nodes in the tree = 1836
-- (32) Type of analysis effectively used = 1
-- (7) Ordering option effectively used = 2
ICNTL(6) Maximum transversal option = 0
ICNTL(7) Pivot order option = 7
ICNTL(14) Percentage of memory relaxation = 5
Number of level 2 nodes = 0
Number of split nodes = 0
RINFOG(1) Operations during elimination (estim)= 1.048D+04

MEMORY ESTIMATIONS ...
Estimations with standard Full-Rank (FR) factorization:
  Total space in MBytes, IC factorization (INFOG(17)): 1
  Total space in MBytes, OOC factorization (INFOG(27)): 1

Elapsed time in analysis driver= 0.0012

***** FACTORIZATION STEP *****

GLOBAL STATISTICS PRIOR NUMERICAL FACTORIZATION ...
Number of working processes = 1
ICNTL(22) Out-of-core option = 0
ICNTL(35) BLR activation (eff. choice) = 0
ICNTL(14) Memory relaxation = 5
INFOG(3) Real space for factors (estimated)= 3508
INFOG(4) Integer space for factors (estim.)= 25652
Maximum frontal size (estimated) = 13
Number of nodes in the tree = 1836
Memory allowed (MB -- 0: N/A) = 0
Memory provided by user, sum of LMK_USER = 0
Relative threshold for pivoting, CNTL(1) = 0.0000D+00
Effective size of S (based on INFO(39))= 5788

Elapsed time to reformat/distribute matrix = 0.0001

** Memory allocated, total in Mbytes (INFOG(19)): 1
** Memory effectively used, total in Mbytes (INFOG(22)): 1

Elapsed time for factorization = 0.0172

Leaving factorization with ...
RINFOG(2) Operations in node assembly = 2.998D+03
----- (3) Operations in node elimination = 1.048D+04
TOUT (A) Scaling effectively used = a
```



Shared file system storage

- Also called CephFS
- Backed up externally
- Can be accessed by multiple instances simultaneously
- Is most like a HPC cluster experience
- Setup may be tricky
- Currently only available on Arbutus

Using shared file system

- Create share (if needed)
- Configure system and mount
 - https://docs.alliancecan.ca/wiki/Arbutus_CephFS
- Manage access for mounted directory

```
sudo chown -R ubuntu /mnt/def-shuber-example
```

Project / Share / Shares

Shares

Filter

Displaying 1 item

<input type="checkbox"/>	Name	Description	Metadata	Size	Status	Protocol	Visibility ⓘ	Share Network	Share Group	Actions
<input type="checkbox"/>	def-shuber-example	test		1000GiB	Available	CEPHFS	private	-	-	<input type="button" value="Edit Share"/> <input type="button" value="v"/>

Displaying 1 item

Shares

- Share Snapshots
- Share Networks
- Security Services
- Share Groups
- Share Group Snapshots

Identity >

Object storage

- Flat file system
- Files stored together in buckets
- Public/private access switchable
- Files not editable in the store
- Customizable metadata
- Access via a client, e.g. s3cmd
https://docs.alliancecan.ca/wiki/Arbutus_Object_Storage_Clients
- Use case:
 - Reading millions of small files
 - May integrate with software, e.g. Pytorch



Be kind, don't store millions of
small files in a shared file
system

Dashboard setup and management

The screenshot displays the OpenStack dashboard interface for managing containers. On the left, a navigation sidebar includes sections for API Access, Compute, Volumes, Network, Orchestration, Object Store, Containers (highlighted), Share, and Identity. The main content area is titled "Containers <- OpenStack bucket" and shows a sub-section for "def-shuber-example".

Container Summary:

- Object Count: 11
- Size: 149.37 MB
- Date Created: Oct 21, 2022
- Storage Policy: default-placement
- Public Access: Disabled

Object List:

Name	Size	Actions
1138_bus.mtx	44.46 KB	Download
494_bus.mtx	18.34 KB	Download
662_bus.mtx	26.65 KB	Download
685_bus.mtx	33.33 KB	Download
bcsstk02.mtx	45.23 KB	Download
bcsstk17.mtx	5.35 MB	Download
bcsstk18.mtx	1.97 MB	Download
bcsstk25.mtx	3.34 MB	Download
bcsstm25.mtx	384.33 KB	Download
cfd2.mtx	40.01 MB	Download
stomach.mtx	98.16 MB	Download

Accessing your objects with s3cmd

```
#!/bin/bash
module load mumps-parmetis

runfile=c_example_read_mm
logfile=out.txt
filelist=$(s3cmd ls s3://def-shuber-example/ | awk '{ print $4 }')
for matrixname in $filelist; do
    echo $matrixname
    time {
        s3cmd get ${matrixname} matrix.mtx;
        ./$runfile matrix.mtx >> $logfile;
        rm matrix.mtx;
    }
done
```

Thanks for listening!

Further details and help

- <https://docs.alliancecan.ca/wiki/Cloud>
- cloud@tech.alliancecan.ca

Questions?