Working with CloudMan

Enis Afgan

@ GCC 2014, Baltimore
How to use the cloud?

1. Get an account on the supported cloud
2. Start a master instance via the cloud web console or CloudLaunch
3. Use CloudMan’s web interface on the master instance to manage the platform
4. Use or customize Galaxy
YOUR TURN
Launch an instance

1. Visit biocloudcentral.org
2. Enter the access key and secret key provided by Dave Clements on 6/25
3. Provide your email address
4. Use your initials as the cluster name
5. Set any password (and remember it)
6. Keep Large instance type
7. Start your instance

Wait for the instance to start (~2-3 minutes)

For more details, see wiki.galaxyproject.org/CloudMan
Launch an instance

1. Visit usegalaxy.org/cloudlaunch
2. Enter your **access key** and **secret key**
   (credits provided by Dave Clements on 6/25)
3. Choose **New cluster**
4. Set **any name** as the cluster name
5. Set **any password**
6. Keep **cloudman_keypair**
7. Keep **Large** instance type
8. Launch your instance

Wait for the instance to start (~2-3 minutes)

For more details, see wiki.galaxyproject.org/CloudMan
Customize your instance - install a new tool

cd /mnt/galaxy/export
unzip dnaclust_linux_release3.zip
cd dnaclust_linux_release3
chmod +x *
Use the new tool in the cluster mode

1. Create a new sample shell file to run the tool; call it `job_script.sh` with the following content:
   ```bash
   #$ -cwd
   ./dnaclust -l -s 0.9 /mnt/workshop-data/mtDNA.fasta
   ```
2. Submit single job to SGE queue
   ```bash
   qsub job_script.sh
   ```
3. Check the queue: `qstat -f`
4. Job output will be in the local directory in file `job_script.sh.o#`
5. Start a number of instances of the script:
   ```bash
   qsub job_script.sh (*10)
   watch qstat –f
   ```
   1. See all jobs lined up
6. See auto-scaling in action (using `/cloud`) [1.5-2 mins]
7. See jobs being distributed
Sharing-an-Instance

- Share the entire CloudMan platform
- Includes all of user data and even the customizations
- Publish a self-contained analysis
- Make a note of the share-string and send it to your neighbor
UNDERPINNING CONCEPTS
Launcher / Management Console

Application(s) (eg, Galaxy)

Contextualize image

Start CloudMan

Setup services

Application(s) (eg, Galaxy)

CloudMan machine image

CloudMan master instance

Persistent data repository

S3/Swift

FS archive

Managed FS

Inst. storage

Snap1

Snap...

FS 1

FS ...

Managed FS

Block storage

Client

1°

2°

3°

4°

5°

6°, 8°

9°

10°

11°
What do you get?

• Cluster-in-the-cloud: SGE
• Galaxy on the Cloud + control
• Customizable: tools, configs, data
• Sharable
• Extensible
CloudMan core

- well-defined interface / API
- web interface

Tool as a service

- service dependencies
- automatic dependency management
- dynamic loading

- cloud resource management
- multi-cloud interface
- service management and monitoring
- user interaction
- system state

- self-contained service definition
- implements the default service interface
- implements optional service hooks
Supported cloud middleware

1. Amazon Web Services
2. OpenStack
3. Eucalyptus
4. OpenNebula
Building

- Leverage **CloudBioLinux** build framework
- Via GVL flavor
  - Base CloudMan machine image
  - Tools and data
- There are also more specific CBL flavors available
  - cloudman
  - [github.com/afgane/gvl_flavor](https://github.com/afgane/gvl_flavor)
Deploying

- Integrated with BioCloudCentral.org app
- Use the public one, deploy your own or run locally
- BCC supports multiple clouds
Troubleshooting

1° Launcher / Management Console

2° /mnt/cm/paster.log

3° /mnt/galaxy[Indices]

4° Persistent data repository

5° CloudMan machine image

6° CloudMan MI

7° Application(s) (eg, Galaxy)

8° /mnt/cm/paster.log

9° /usr/bin/ec2autorun.log

10° /tmp/cm/cm_boot.og

11° S3/Swift

CM-w

CM-w

CM-w

CM-w

CM-w

...