Galaxy CloudMan
Enis Afgan & Dannon Baker
GCC 2012, Chicago
The big picture
When to use the cloud?

• Don’t have informatics expertise or the infrastructure to run and maintain
• Have variable or particular resource needs
• Cannot upload data to a shared resource
• Need for customization
• Have oscillating data volume
• Want to test or share a tool, quickly & safely
• Want to make your analysis readily available to others
• Want fast access to AWS public datasets
Deploying an instance

1. Get an account on the supported cloud
2. Start a master instance via the cloud web console or CloudLauch
3. Use CloudMan’s web interface on the master instance to manage the platform
4. Use or modify Galaxy
Available functionality

• Start/launch through a web browser or the command line
• Choose between four cluster types
• Terminate/restart
• Scale
• Auto-scale
• Spot support
• Persist changes
• Share-an-instance (customized one too)
• Expand the file system
• Customize (via CLI or the Tool Shed): tools, data, references
• Access through ssh
• Mount an S3 bucket -> data library
• Control the Galaxy process
• API (via Blend)
• Build your own image (via mi-deployment)
Workshop plan

- Launch an instance
- Go through (most of) the ‘Galaxy 101’ tutorial while using the following CloudMan features:
  - Auto-scaling
  - Using an S3 bucket as a data source
  - Accessing an instance over ssh
  - Customizing an instance
  - Controlling Galaxy
  - Persisting changes to an instance (maybe)
  - Sharing-an-instance
  - API (time permitting)
GENTLEMAN, START YOUR BROWSERS

AND LADIES

GENTLEMAN, START YOUR BROWSERS
Start an Instance

1. Go to the URL on the sheet handed out at the beginning of the workshop and login

2. Create a security group (*use your username as the name of the security group*)

3. Create a key pair (*use your username as the key name and save it locally*)

4. Start an instance (*ami-78a00411*), composing the user data in the process and *using your username as the cluster name*

5. Wait for the instance to start

For exact details, see Step 2 on wiki.g2.bx.psu.edu/CloudMan
Configure Your Cluster

Initial Cluster Configuration

Welcome to CloudMan. This application will allow you to manage this cluster and the services provided within. To get started, choose the type of cluster you’d like to work with and provide the associated value, if any.

- **Galaxy Cluster**: Galaxy application, available tools, reference datasets, SGE job manager, and a data volume. Specify the initial storage size (in Gigabytes):
  - [ ] GB

- **Share-an-Instance Cluster**: Derive your cluster form someone else’s cluster. Specify the provided cluster share-string (for example, cm-011923649e9271f17c4f3ba6846db0/shared/2011-08-19--21-00):
  - [ ] Cluster share-string

- **Data Cluster**: A persistent data volume and SGE. Specify the initial storage size (in Gigabytes):
  - [ ] GB

- **Test Cluster**: SGE only. No persistent storage is created.
  - [ ]

Start CloudMan Cluster
Manage Your Cluster

CloudMan Console

Welcome to CloudMan. This application allows you to manage this instance cloud cluster and the services provided within. Your previous data store has been reconnected. Once the cluster has initialized, use the controls below to manage services provided by the application.

- Terminate cluster
- Add nodes
- Remove nodes
- Access Galaxy

Status

- Cluster name: ghem
- Disk status: 0 / 0 (0%)
- Worker status: Idle: 4 Available: 2 Requested: 5
- Service status: Applications  Data

Autoscaling is off. Turn on?
Register as a new user (use gcc2012 as the password because you’ll share it with your neighbor later)
Workshop plan

• Launch an instance ✓
• Go through (most of) the ‘Galaxy 101’ tutorial while using the following CloudMan features :
  • Auto-scaling
  • Using an S3 bucket as a data source
  • Accessing an instance over ssh
  • Customizing an instance
  • Controlling Galaxy
  • Persisting changes to an instance (maybe)
  • Sharing-an-instance
  • API (time permitting)
Auto-scaling

Autoscaling Configuration

Autoscaling attempts to automate the elasticity offered by cloud computing for this particular cluster. Once turned on, autoscaling takes over the control over the size of your cluster.

Autoscaling is simple, just specify the cluster size limits you want to want to work within and use your cluster as you normally do. The cluster will not automatically shrink to less than the minimum number of worker nodes you specify and it will never grow larger than the maximum number of worker nodes you specify.

While respecting the set limits, if there are more jobs than the cluster can comfortably process at a given time autoscaling will automatically add compute nodes; if there are cluster nodes sitting idle at the end of an hour autoscaling will terminate those nodes reducing the size of the cluster and your cost.

Once turned on, the cluster size limits respected by autoscaling can be adjusted or autoscaling can be turned off.

Minimum number of nodes to maintain:

0 OK

Maximum number of nodes to maintain:

3 OK

Type of Nodes(s):

Same as Master

Turn autoscaling on
Using an S3 bucket as a data source
Accessing an instance over ssh

$ chmod 400 GCC2012.pem

$ ssh -i GCC2012.pem ubuntu@ec2-107-21-161-225.compute-1.amazonaws.com

• Once logged in, you have full system access to your instance, including sudo; use it as any other system

• *galaxy* user exists on the system and should be used when manipulating Galaxy (sudo su galaxy)
Customizing an instance

• **Edit Galaxy’s configuration**

  $ sudo su galaxy
  $ cd /mnt/galaxyTools/galaxy-central
  $ vi universe_wsgi.ini

  `allow_library_path_path_paste = True`

• Add a new tool via the command line

• Add a new tool via the Tool Shed
Controlling Galaxy

- Restart Galaxy so the config change takes effect
- Add an admin user
  - Use the email you registered with
- Within Galaxy, create a Data Library, using S3 bucket path as the data source (/mnt/gcc2012)
  - This will import SNPs.bed dataset to the Data Library
- Import that dataset into a history
Galaxy 101

- usegalaxy.org/galaxy101/
- Follow **through the end of Step 2**
- Instead of downloading SNPs from UCSC, import them from the created Data Library
Sharing-an-Instance

- Share the entire Galaxy 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
Use CloudLaunch to start a new, derived, instance


- CloudLaunch alleviates all instance startup/setup steps (and is fully interchangeable with the web console!)

- Choose *Share-an-instance* type and enter your neighbor’s *share-string*

- After Galaxy starts up, login using your neighbor’s email address & see all the data
API (via Blend)

blend

- A python library for interacting with Galaxy’s API
- And CloudMan

- Documentation available at http://blend.readthedocs.org/
Request compute infrastructure
Manipulate compute infrastructure
Upload data and run analyses
Docs and examples included
Test
Distribute
Requesting an Instance

$ [sudo] pip install blend-lib

$ python

```python
from blend.cloudman.launch import CloudManLaunch

cml = CloudManLaunch(access_key, secret_key)

c = cml.launch('GCC2012_CloudMan_Workshop', img_id, 'm1.large', 'gcc2012')

cml.get_status()
```

Just an example, we’ll skip this step
from blend.cloudman import CloudMan

cm = CloudMan("instance IP", "gcc2012")

# cm.initialize(type="Galaxy")

cm.get_status()

cm.add_nodes(1, 't1.micro')
Working with Galaxy’s API

from blend.galaxy import GalaxyInstance

gi = GalaxyInstance('Galaxy IP', key='API key')
libs = gi.libraries.get_libraries()
gi.workflows.show_workflow('workflow ID')
gi.workflows.run_workflow('workflow ID', input_dataset_map)

Come to the Galaxy API workshop 😊
WS6 @ 11am and 3:30pm (in this room)
Workshop plan

• Launch an instance
• Go through (most of) the ‘Galaxy 101’ tutorial while using the following CloudMan features:
  • Auto-scaling
  • Using an S3 bucket as a data source
  • Accessing an instance over ssh
  • Customizing an instance
  • Controlling Galaxy
  • Persisting changes to an instance (maybe)
  • Sharing-an-instance
  • API (time permitting)
Troubleshooting

/mnt/galaxy[Tools|Indices|Data]

Instance block storage

1°
2°
3°

Management Console

/start command

Contextualize image

Start CloudMan

Setup services

Application(s) (eg, Galaxy)

Persistent data repository

S3/Swift

/opt galaxy/pkg/ec2autorun.log
/tmp/cm/cm_boot.log

CloudMan machine image
CloudMan instance

1°
2°
3°
4°
5°
6°
7°
8°
9°
10°
11°