Deploying CloudMan on the NeCTAR Research Cloud

usecloudman.org
CloudMan features

• A complete solution for instantiating, running and scaling cloud resources
  • Get a scalable compute cluster (SGE)
• Get an automatically configured Galaxy
  • Scope of tools and reference datasets exceed Galaxy Main
• Deployment on AWS, OpenStack, and OpenNebula clouds
  • Wizard-guided setup: requires no computational expertise, no infrastructure, no software
• Automated configuration for machine image, tools, and data
• Self-contained deployment
• Elastic resource scaling: manual or automatic
• Dynamic persistent storage
• Share your instance: including all customizations (data, tools & configurations)

• Deploy a (Galaxy) cluster in minutes!
CloudMan architecture

1° Management Console
2° Contextualize image
3° Start CloudMan
4° Setup services
5° Instance block storage
6°, 8° Persistent data repository
7° Application(s) (eg, Galaxy)
8° CloudMan machine image
9° S3/Swift
10° CloudMan instance
11° CM-w
CloudMan internals

- Written in Python as a web app
- Uses boto library for cloud communication
- Porting to OpenStack
  - Introduced a new cloud interface
  - Added new connection properties
  - 90% compatible & it works!
NeCTAR RC Experience
CloudMan Requirements

- A preconfigured machine image
- User data
- Persistent object store
- Resource metadata (i.e., tags)
- Data volumes and volume snapshots
CloudMan Requirements

- A preconfigured machine image
- User data
- Persistent object store
- Resource metadata (ie, tags)
- Data volumes and volume snapshots
A preconfigured MI

- Contains system-level requirements & contextualization hooks
- Machine image creation is automated via Fabric scripts
  - mi-deployment:
    ```
    fab -f mi_fabfile.py -i private_key_file -H instance_IP configure_MI
    ```
  - As part of CloudBioLinux
    ```
    fab -f fabfile.py -i private_key_file -H instance_IP -u username install_biolinux
    ```
- The CloudBioLinux version requires more than 10GB root image size
  - Had to reduce the number of installed tools
  - Offloaded some parts to a shared NFS disk
Resource metadata (ie, tags)

- Used to identify resources
  - Cluster restarts
  - User-identification via the Dashboard
- boto does not work
- Native OpenStack python-novaclient does not work
  - Some code twiddling helps but does not seem to resolve it
- REST API works (but…)
- Filed a bug on launchpad (should be fixed in essex?)

https://bugs.launchpad.net/nova/+bug/972102
Enable easy creation of user-specific cloud platforms

Couple the infrastructure, complex and functional application execution environments, applications, and data into a single unit that can easily be used and manipulated by a user.
Data volumes (and snapshots)

- Volumes provide
  - Persistence
  - Enable pre-configured data elements
  - Platform sharing
- Waiting…
The Research Cloud works quite smoothly!

and thank you for that.