CloudBioLinux: History, current status and DebianMed integration

Brad Chapman
Bioinformatics Core,
Harvard School of Public Health
https://github.com/chapmanb

1 February 2014

Outline

- Overview of CloudBioLinux
- Cloud Adoption in Biology
- DebianMed integration
- Goals for Hackathon
- Walk through of CloudBioLinux

What is CloudBioLinux?

Infrastructure for installing biological software

- deb/rpm packages
- Bio-Linux
- Linuxbrew with homebrew-science
- Python, Ruby, R package management
- Conda + Binstar `https://conda.binstar.org/`
- Custom installation scripts

History

Integration of multiple efforts

- JCVI Cloud Bio-Linux
- Bioperl Max
- Infochimps machetEC2
- Bio-Linux
- DebianMed

Original goal

Overcome bare-metal problem with AWS images

- Ubuntu
- Single AMI with biological tools
- Automated build infrastructure
- Bring in developer community
- Ready to use for researchers

Biological data

- Genomes, organized and indexed
- Associated data files: dbSNP, reference transcripts
- S3 bucket
- Tools with organized data
- GEMINI: https://github.com/arq5x/gemini

Local installation

- Multiple platforms: RedHat/CentOS, Debian, ScientificLinux
- Isolated installations: no sudo, non-VM environments
- Rapid turnaround for fixes

Flavors: customized installations

- Target specific use case
- Sub-collection of packages from full distribution
- Example: cloudbiolinux/contrib/flavor/biopython

Pjotr Prins

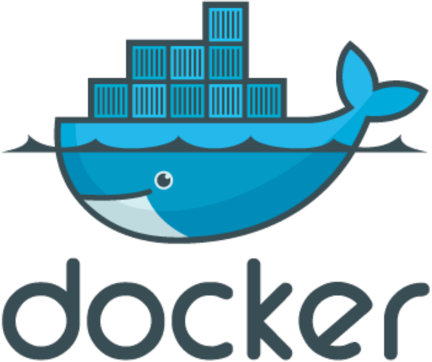
Hidden infrastructure

bcbio-nextgen

- CloudBioLinux drives fully automated installation
- Reproducible build scripts for docker migration

https://github.com/chapmanb/bcbio-nextgen

Containers



http://docker.io/

Practical docker

- Wrapper running bcbio-nextgen inside docker
- Create docker container with CloudBioLinux
- External code handles cluster integration

`https://github.com/chapmanb/bcbio-nextgen-vm`

Outline

- Overview of CloudBioLinux
- Cloud Adoption in Biology
- DebianMed integration
- Goals for Hackathon
- Walk through of CloudBioLinux

Hurdles

- Cost
- Investment in local hardware
- Non-automated culture
- Clash with HPC design

Current use cases

- New users
- One off jobs
- Hard to install software
- Training

http://compbio.sph.harvard.edu/chb/training

Key to success

http://deviq.com/dogfooding

http://en.wikipedia.org/wiki/Eating_your_own_dog_food

What is changing?

- Data sizes
- Access to local compute
- Local infrastructure mimic cloud
  - Docker: fully automated
  - Local VMs: vagrant

Outline

- Overview of CloudBioLinux
- Cloud Adoption in Biology
- DebianMed integration
- Goals for Hackathon
- Walk through of CloudBioLinux

Platform support

- Docker solves multi-platform support issues
- Allows use of single base image, local and cloud
- More Ubuntu + DebianMed + Bio-Linux packages

Real time updates

- Homebrew + CloudBioLinux scripts
- Allow immediate pushes for new version or fixes
- Critical for pipeline support
- Also want to contribute upstream as packages stabilize

Outline

- Overview of CloudBioLinux
- Cloud Adoption in Biology
- DebianMed integration
- Goals for Hackathon
- Walk through of CloudBioLinux

CBL Debian repository

- How can we do fast repo + contribute upstream?
- Quick packaging: FPM `https://github.com/jordansissel/fpm`
- Quick repository: apotiki `https://github.com/pyr/apotiki`
- Other approaches?

Manifest

- Full manifest of installed software
- Automated runs
- Prioritize biological software
- Work in progress script

`https://github.com/chapmanb/cloudbiolinux/blob/master/utils/cbl_installed_software.py`

Automated CloudBioLinux packaging

- Flavor to full image
  - Docker
  - Amazon AMI
  - Virtualbox
- build-debian-cloud https://github.com/camptocamp/build-debian-cloud
- packer http://www.packer.io/

Outline

- Overview of CloudBioLinux
- Cloud Adoption in Biology
- DebianMed integration
- Goals for Hackathon
- Walk through of CloudBioLinux

CloudBioLinux architecture

- Fabric scripts
- YAML configuration
- Flavors
- Documentation