

Adding Transparency and Automation into the Galaxy Tool Installation Process

Enis Afgan
Johns Hopkins University
Galaxy Team

Galaxy Admins Web Meetup
August 20, 2015.

Outline

- Installing tools in bulk (i.e., Automation)
- Keeping up with what's installed on Main & Test (i.e., Transparency)
- Retrieving the toolset from a Galaxy instance [coming]

A common question:

“How can I replicate [the toolset from] Galaxy Main?”

Answer:

By hand; perhaps via the API;
recently, inspect the `usegalaxy-playbook`

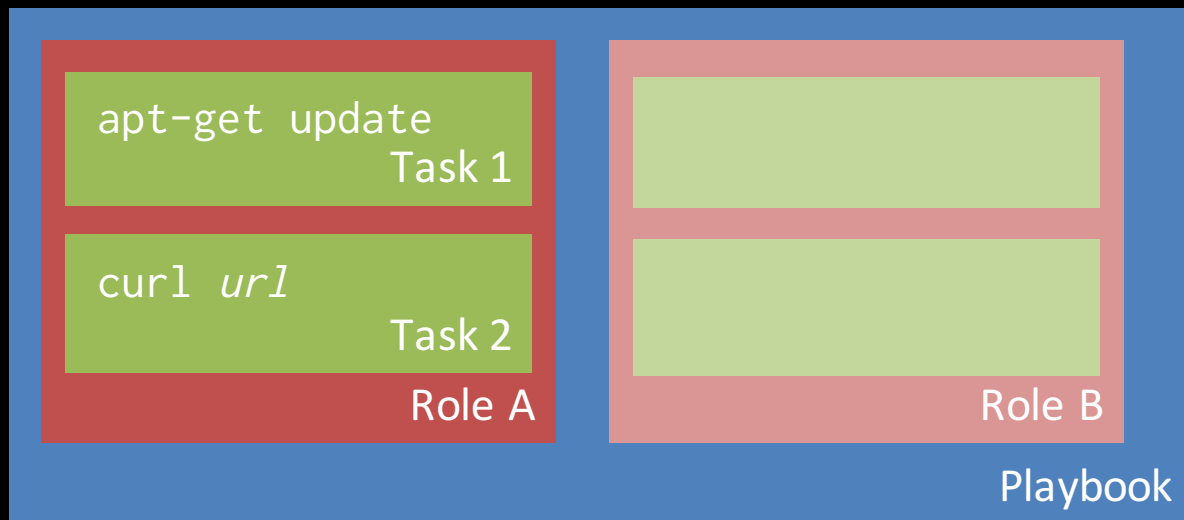
A more general question:

How do I install dozens or hundreds or tools?

Answer:

With a lot of clicks.

Ansible anatomy



An example task:

- name: Install required system packages
 - apt: pkg={{ item }} state=latest
 - with_items:
 - ansible
 - automake
 - ...

Ansible role: Galaxy Tools

Automate installation of tools from a Tool Shed into Galaxy

1. **A list of tools** in a dedicated YAML file
 - Readily editable, can specify required level of details
2. **A tool installation script** reads the tools file and performs the installation
 - Idempotent, command line options for installing tools individually

github.com/galaxyproject/ansible-galaxy-tools

The tools list file

tools:

- **name**: synapse_interface
owner: kellrott
tool_panel_section_id: peak_calling
- **name**: fastqc
owner: devteam
tool_panel_section_id: cshl_library_information
tool_shed_url: <https://testtoolshed.g2.bx.psu.edu>
revisions:
 - '8c650f7f76e9' # v0.62
 - 'd2cf2c0c8a11' # v0.63

The tool installation script

- An option to create (and delete) a bootstrap Galaxy user
galaxy_tools_create_bootstrap_user: yes
galaxy_tools_delete_bootstrap_user: yes
- Can be used standalone to install individual tools

```
$ cd files  
$ python install_tool_shed_tools.py -g g_url -a key --name X --owner Y --section Z
```

- Useful for dev and testing but should not be the default use (we'll see why in the coming slides)

Using the Galaxy Tools role

A ready-to-use playbook is available at github.com/afgane/galaxy-tools-playbook

- Clone and edit the tools file; run

```
$ git clone --recursive git@github.com:afgane/galaxy-tools-playbook.git
# Provide a list of tools in files/tool_list.yml
$ ansible-playbook tools.yml -i "localhost," --extra-vars galaxy_tools_api_key=<key>
```

Galaxy instance will be populated with the desired toolset – as long as the tools install properly, you're all set.

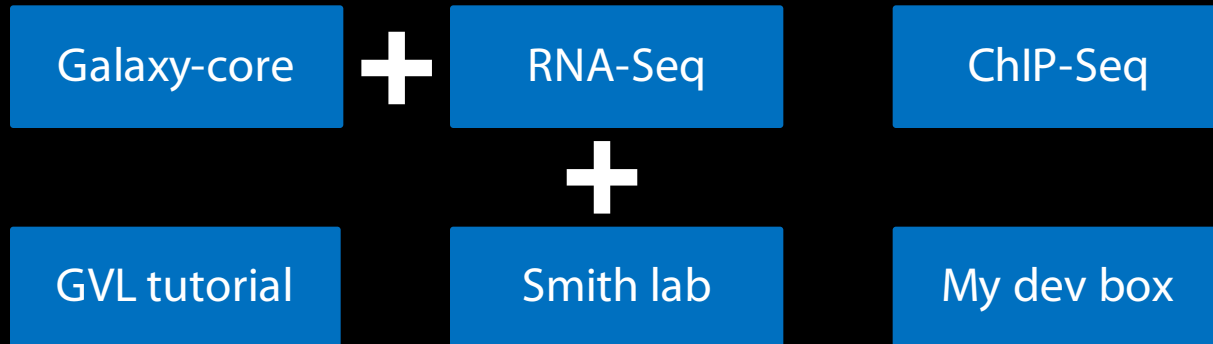
Commit the tools file and have a

1. A track-record of the installed tools
2. A repeatable Galaxy setup

Support for Galaxy Flavors

Create tailored instances of the tools file, each with a focus on different type of analysis

(Community effort?)



Adding transparency

Tools on Main and Test were installed sporadically and by hand, so no insight or feedback on when and what has changed.

In the future, tools will be installed exclusively via the Galaxy Tools role



Still work in progress...

Main toolset

```
3 tools:
4 - name: 'column_maker'
5   owner: 'devteam'
6   tool_panel_section_id: 'textutil'
7   revisions:
8     - '08a01b2ce4cd' # v1.1.0
9 - name: 'tabular_to_fasta'
10  owner: 'devteam'
11  tool_panel_section_id: 'convert'
12  revisions:
13    - '0b4e36026794' # v1.1.0
14 - name: 'fasta_to_tabular'
15  owner: 'devteam'
16  tool_panel_section_id: 'convert'
17  revisions:
18    - '9d189d08f2ad' # v1.1.0
19 - name: 'fastqtofasta'
20  owner: 'devteam'
21  tool_panel_section_id: 'convert'
22  revisions:
23    - '3571553aeb20' # v1.0.0
24 - name: 'dna_filtering'
25  owner: 'devteam'
26  tool_panel_section_id: 'filter'
27  revisions:
28    - 'a6f0d355b05f' # v1.0.3
```

Test toolset

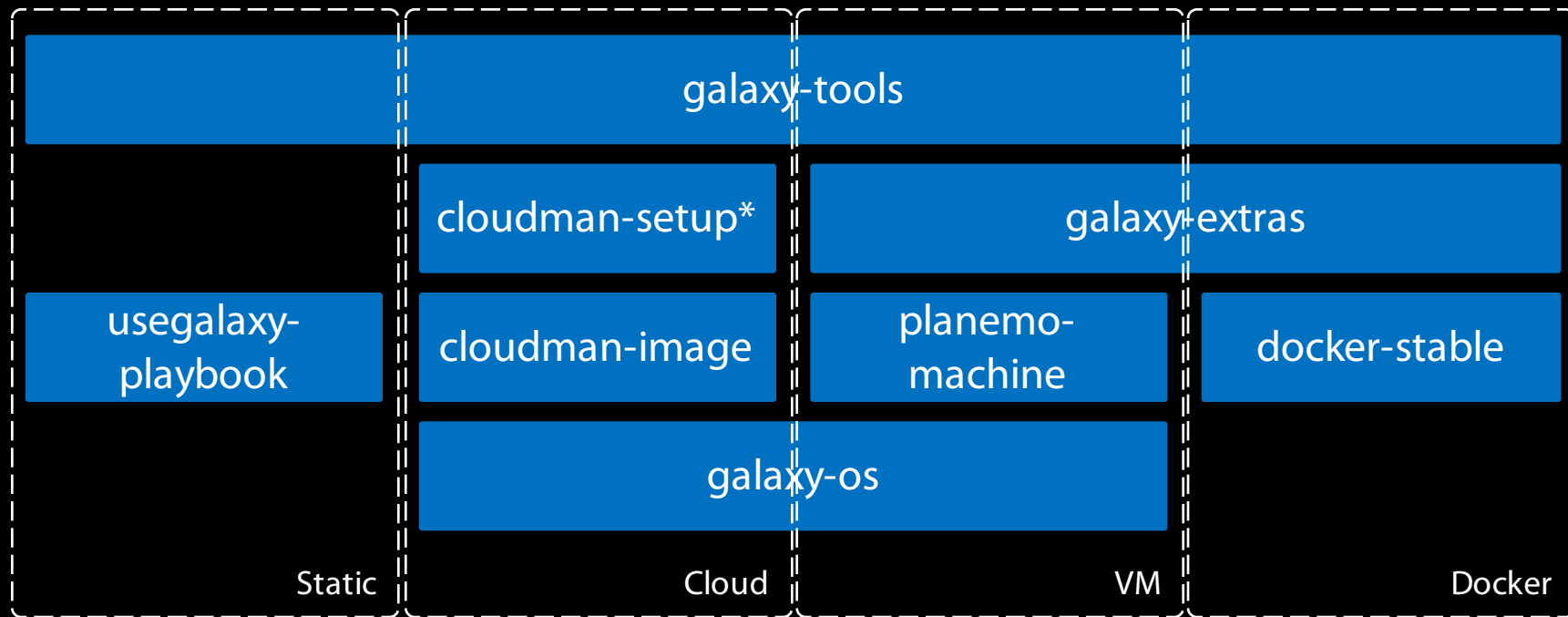
```
1 ---
2 # This is a sample file to be used as a reference
3 # tools that you wish to install into Galaxy from
4 # `install_tool_shed_tools.py` script.
5 #
6 # For each tool you want to install, you must provide
7 # * name: this is the name of the tool to install
8 # * owner: owner of the Tool Shed repository from which
9 #           the tool is installed
10 # * tool_panel_section_id: ID of the tool panel section
11 #                        that the tool is installed in
12 #                        in Galaxy's `shed_tools.yml` file
13 #                        that the specified section ID is used
14 #                        Otherwise, the tool will not be
15 #                        installed.
16 #
17 # In addition, you can specify the following optional
18 # installation properties:
19 # * tool_shed_url: the URL of the Tool Shed from which the
20 #                 tool is installed. (default: https://toolshed.galaxyproject.org)
21 # * revisions: a list of revisions of the tool, and the
22 #             revision that should be installed. (default: latest)
23 # * install_tool_dependencies: True or False - whether to
24 #                               install dependencies or not
25 # * install_repository_dependencies: True or False - whether to
26 #                                   install repository dependencies or not
```

Populating the tools file

- For existing Galaxy instances, populating the tools file is a tedious manual process
- Need a script to inspect the toolset and generate the tools file
 - Technically, should be possible with the next Galaxy release

Work in progress: github.com/galaxyproject/ansible-galaxy-tools/blob/master/files/install_tool_shed_tools.py#L205

Ansible stack -> consistent experience

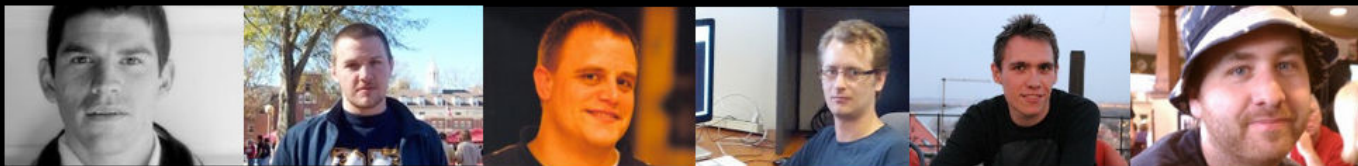


* Not a single role

Fire-and-forget obstacles

- Tools fail to install cleanly
- Delay between the tool repos and what's in the Tool Shed
- Tool panel sections must exist in Galaxy before tool installation (no API to create those)

The Galaxy Team



Enis Afgan

Dannon Baker

Dan Blankenberg

Dave Bouvier

Marten Cech

John Chilton



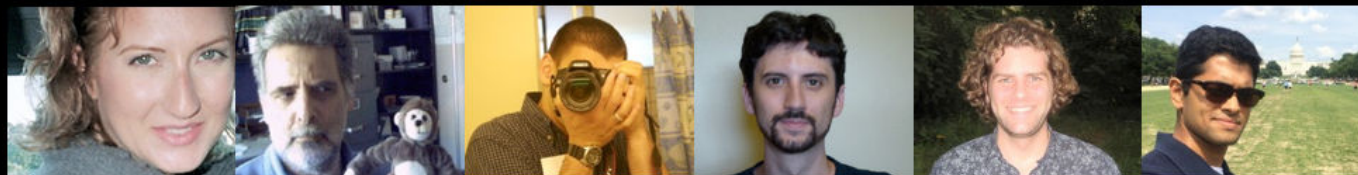
Dave Clements

Nate Coraor

Carl Eberhard

Jeremy Goecks

Sam Guerler



Jen Jackson

Ross Lazarus

Anton Nekrutenko

Nick Stoler

James Taylor

Nitesh Turaga

<http://wiki.galaxyproject.org/GalaxyTeam>