

Reproducible Galaxy: Improved development and administration

Aarif Mohamed Nazeer Batcha¹, Sebastian Schaaf^{1,2}, Guokun Zhang¹, Sandra Fischer¹, Ashok Varadharajan¹

¹ Department of Medical Informatics, Biometry and Epidemiology (IBE), Ludwig Maximilians University (LMU) Munich, Germany ² German Cancer Consortium (DKTK) partner site Munich, German Center for Cancer Research (DKFZ) Heidelberg, Germany

Reproducible research, by definition, is the distribution of research publication along with data, software source code, and tools required to reproduce the results discussed in the *publication.* - orgmode.org¹

Galaxy is an optimal platform for such reproducibility efforts.

An ideal Galaxy production environment offers such reproducibility for users. However, such luxury is often neglected for developers and administrators. A developer always aims for progress, while admins require consistancy. In most situations, they face limiting constraints like time or computing resources and have to try to figure out common grounds at the cost of reproducibility.

The Munich NGS-FabLab² is the first crossinfrastructure for medical institutional IT research at the LMU, developed due to local contraints like the impossibillity to go for cloud services. Although under constant develop-ment, it has already become essential as a common and reproducible work environment for users, admins and developers.

Such goal has been achieved in the past by establishing a Galaxy framework via an automated shell setup script³, which got constantly less manageable. As an improvement, its functionalities were reviewed and re-coded into Ansible scripts.

Ansible³ is an open-source, easy-to-script and very powerful automation tool, intended to simplify deployment, configuration and management of computer nodes over SSH, with the latter being the only dependancy on the target host. A key element here are the dedicated scripts, called 'playbooks'. The respective human-readable markup languange provides high-level commands and a range of preconfigured control packages (e.g. for permissions, config file manipulation and service handling). Consequently, Ansible is in use for a longer time in Galaxy Main. Ansible playbooks have been deployed by Galaxy Main for administration and maintanence and are available online⁵ as an inspiration. In constrast, our Ansible playbooks⁶ are more dedicated to establish a clean and orderly working environment in a reproducible manner by setting up a selection of core modules/services (Fig. 1) bottom up from scratch in order to achieve a working server as close as possible to the production environment. New Ansible snippets can be patched to other instances easily.

In a nutshell, this poster presents and publishes our framework to be a source of inspiration and our contribution to enhance reproducible development. Additionally, it might be indeed deployed as a convenient and scalable working horse for third parties, who neither want to go for less powerful solutions based on e.g. Docker⁷ nor for cloud services like Cloudman⁸, but local hardware and/or VMs.



and Ulrich Mansmann^{1,2}



Fig. 1 – Overview about core modules/services deployed by our playbook. All of them are independent and combinable. In principle, functionalities can be performed by alternate software packages and exchanged as a module (e.g. SGE against another job distributor). As most of the interfaces are realized via network protocols, they are in principle unbound from their (physical or virtual) location, enhancing scalability.

Script Description

Our actual infrastructure¹ runs on a single server with mass storage array attached. A virtualization layer creates separate dedicated servers, sharing common resources wherever possible via TCP/IP.



eliminary Inquiry	 Select installation procedure Enter root password for the host
Host ENV	 Create root bashrc variables Create users and groups
	→ NFS server setup
	 NFS kernel installation Create mount points for galaxy,
PostgreSQL	 Install postgres and dependent Configure postgresql for access databases and users for the ho
Apache HTTP SERVER	 Install apache2 Add modules and configure SSI
Galaxy	 Create galaxy .bashrc variables Install Galaxy from GitHub Configure galaxy.ini file
	→ Grid engine setu
GRID ENGINE	 Install tool dependencies Configure grid engine Install master, execution and submission hosts

NGS-FabLab Framework

Sc
Ansible (YN
Ubuntu 14
~15 minute
Hosts must Hosts must Internet co Script depl
Scripts can SLES, basic must be in

¹http://orgmode.org/worg/orgcontrib/babel/intro.html#orgh ²Schaaf *et al*. 2014 - The Munich FabLab – A glimpse on an IT infrastructure for medical sequ data (Poster P10 at GGC2014) ³Nazeer Batcha *et al.* 2014, Pract experiences from the Munich FabLab (Poster P3 at GCC2014)



- For Galaxy submission host:
- Create job configuration file
- Edit galaxy.ini file for grid configuration

ript Information

ML-formatted playbooks)

.04, SLES 11 SP3 (Suse Linux Enterprise Server) es (incl. all compilations)

t be accessible via SSH

t have identical root password

onnection for downloads

loyment machine must have Ansible installed be executed on blank Ubuntu machines. For zypper repositories for developer packages ntegrated before execution.

NGS-FabLab Framework

	⁴ <u>http://www.ansible.com/home</u>
neadline31	⁵ https://github.com/galaxyproject/usegala
NGS-	<u>xy-playbook</u>
	⁶ <u>https://bitbucket.org/ibe/galaxy-setup-</u>
Jence	<u>scripts.git</u>
	⁷ https://www.docker.com/
ical	⁸ <u>https://wiki.galaxyproject.org/CloudMan</u>
NGS-	
)	