Deploying Community Codes

Joshua Gyllinsky, PhD

This is the fourth edition of this talk. With earlier versions by
Prasad Maddumage, Martin Čuma, and Joshua Alexander

The “Deploying Community Codes” VRP Session

This is the fourth edition of this talk and builds on the previous editions.

1. Prasad Maddumage, “Deploying Community Codes” Presentation and Workshop, 2017 Virtual Residency Program (VRP) workshop, Online and In-Person, July 31, 2017. *https://www.oscer.ou.edu/acirefvirtres2015_talk_deploying_community_codes_20150601.pdf*

2. Martin Čuma, “Deploying Community Codes” Presentation and Demo, 2016 Virtual Residency Program (VRP) workshop, Online and In-Person, August 8, 2016. *https://www.oscer.ou.edu/acirefvirtres2016_talk_cuma_deploycodes_20160808.pdf*

3. Joshua Alexander, “Deploying Community Codes” Presentation and Demo, 2015 Virtual Residency Program (VRP) workshop, Online and In-Person, June 1, 2015. *https://www.oscer.ou.edu/acirefvirtres2015_talk_deploying_community_codes_20150601.pdf*

- High-level.
- Not overly technical.
- Meant to give a starting place.
- Cannot possibly cover all scenarios.

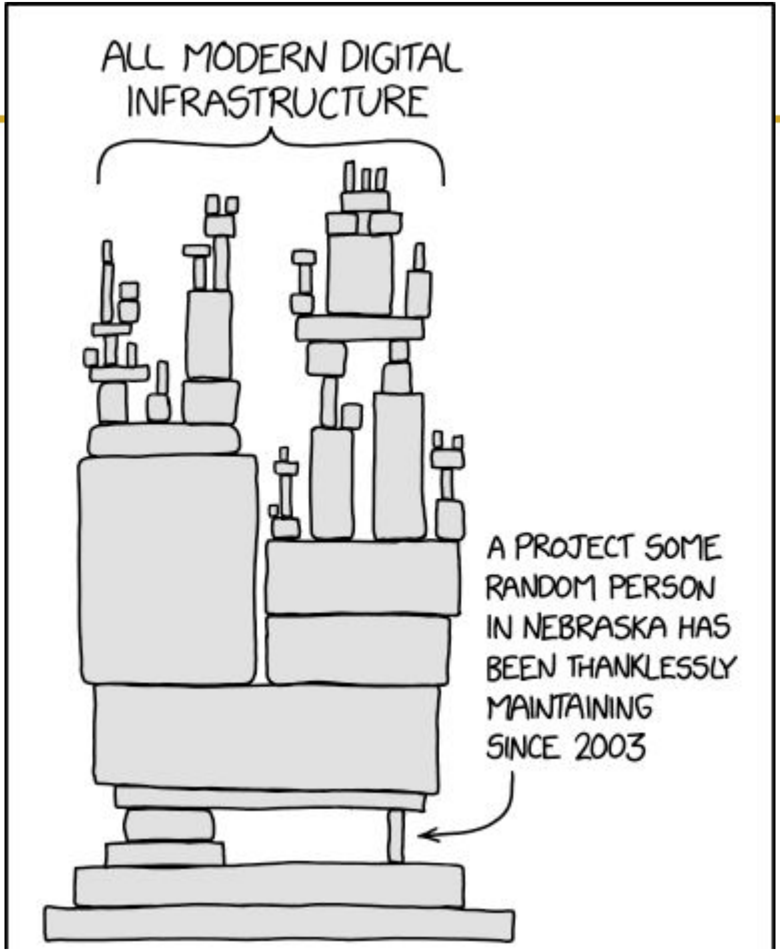
Types of “Codes”

- Computationally intensive scientific analysis (Usually)
- Any programming language (Tends to be usual suspects)
- Usually Free (sort of..)
- POSIX targeted / designed for use with UNIX-like operating systems (mostly)

Types of “Codes”: Ecosystem

Encountered code might be found in several forms in the wild, including:

- OS Level
- Environment Modules
- Full Programs
- Program Modules and Plugins (Require other complete programs)
- User Scripts


ALL MODERN DIGITAL INFRASTRUCTURE
A PROJECT SOME RANDOM PERSON IN NEBRASKA HAS BEEN THANKLESSLY MAINTAINING SINCE 2003

R. Munroe, “Dependency,” *xkcd*. [Online]. Available: https://xkcd.com/2347/. [Accessed: 29-Jun-2022].

Common Needs

Matrix Operations

Fast Fourier Transforms

Linear Algebra

ODE & PDE

Other Numerical Libraries

Weather Forecasting

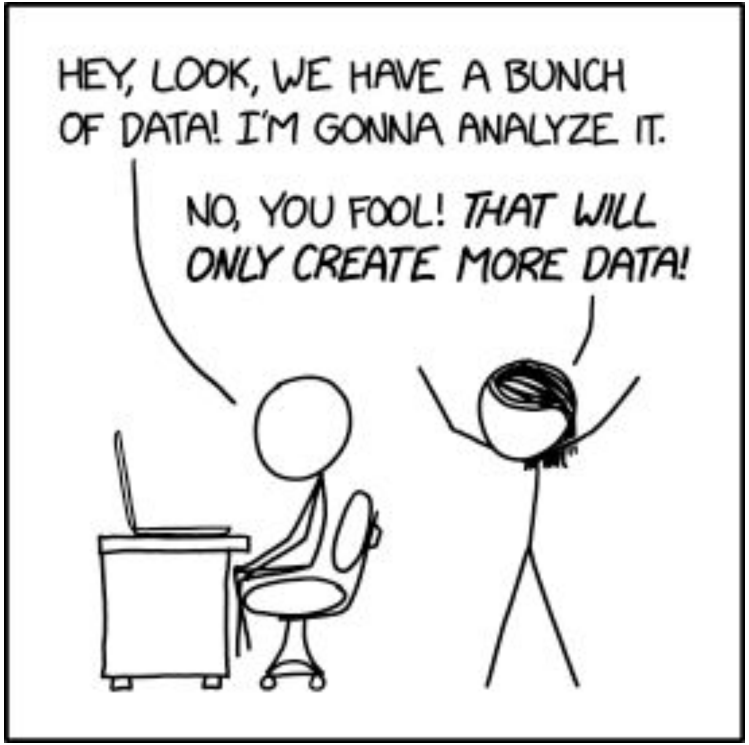
Molecular Dynamics

Ab Initio Chemistry

Bioinformatics

Weather Tracking

Star Formation


HEY, LOOK, WE HAVE A BUNCH OF DATA! I'M GONNA ANALYZE IT.
NO, YOU FOOL! *THAT WILL ONLY CREATE MORE DATA!*

R. Munroe, “Data Trap,” *xkcd*. [Online]. Available: https://xkcd.com/2582/. [Accessed: 29-Jun-2022].

Community And Commercial Programs

Commercial programs

- Licensed to user, department, or institution for specific time period with potential usage restrictions including how it can be used in research
- May need additional licensing validation / verification configurations
- May be segmented into modules with official support tiers
  - *may require additional payment structure*

Community programs

- Free (sort of), written by researchers
- Support and stability may be minimal
- Documentation may be sparse
- Official support may not be available
- May be less tested
  - *Does not necessarily mean more buggy, but might not have been tested on a similar setup to the one you are using*

Community Codes

- Languages, Drivers & Core Libraries
  - Python, R, Julia, C, Fortran, PASCAL, COBOL
- Numerical Libraries
  - FFTW, GotoBLAS, LAPACK
- Scientific Software
  - WRF, GROMACS, NWChem, BLAST, LAMMPS, TopHat
- Visualization Software
  - – WCT, Avogadro, AIPS

Commercial Codes

- Managed with critical concern given to the license methods
  - Access and number of seats, etc
  - Specific upgrade path might require very specific versions of supporting software and might only be eligible for upgrades for a specific timeframe or series
  - May require additional licencing server deployments (networking and security concerns)

- Examples include
  - VASP, Gaussian, COMSOL

Where To Deploy?

- Personal / shared computer ?
- Personal / shared remote machine ?
- Personal / shared remote virtual machine / hypervisor ?
- Personal / shared account on shared setup ?
- As part of a scheduled job ?

Deployment Concerns

- Is this **really for multiple users**?
- How will this **affect other users**?
- How will this affect other software / systems / processes / **dataflow**?
- If there are issues, will this attempt affect future attempts to deploy?
- What is the **management plan** going forward (documentation / updates / releases)?
- Are there **storage and access** concerns?
- What are the **security** concerns?
- Is the data being processed need to be incoming - how can this be built / configured?
- Where is the install media and who maintains it?

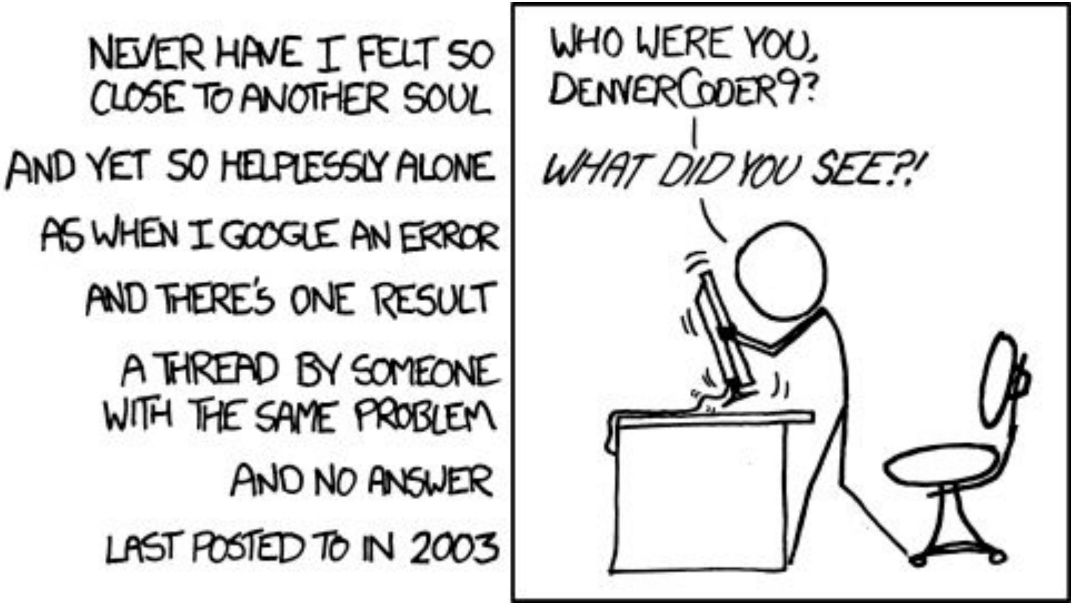
Responsibilities / Who Installs What?

- Your machine (Single-User Systems)
  - You are the administrator
    - ➢ Install it yourself!
- Someone else's machine (Multi-User Systems)
  - You are likely a limited user
    - ➢ Ask the System Administrators
    - ➢ Install locally (if allowed)
    - ! **You will likely need to use a scheduler on run (slurm, etc.)**

Document How You Did It

If you find an incompatibility, please

- Push the patch
- Document
- Update open public ticket with the solution so others can benefit


NEVER HAVE I FELT SO CLOSE TO ANOTHER SOUL
AND YET SO HELPLESSLY ALONE
AS WHEN I GOOGLE AN ERROR
AND THERE'S ONE RESULT
A THREAD BY SOMEONE WITH THE SAME PROBLEM
AND NO ANSWER
LAST POSTED TO IN 2003
WHO WERE YOU, DENVERCODER9?
WHAT DID YOU SEE?!

R. Munroe, "Wisdom of the Ancients," *xkcd*. [Online]. Available: https://xkcd.com/979/. [Accessed: 29-Jun-2022].

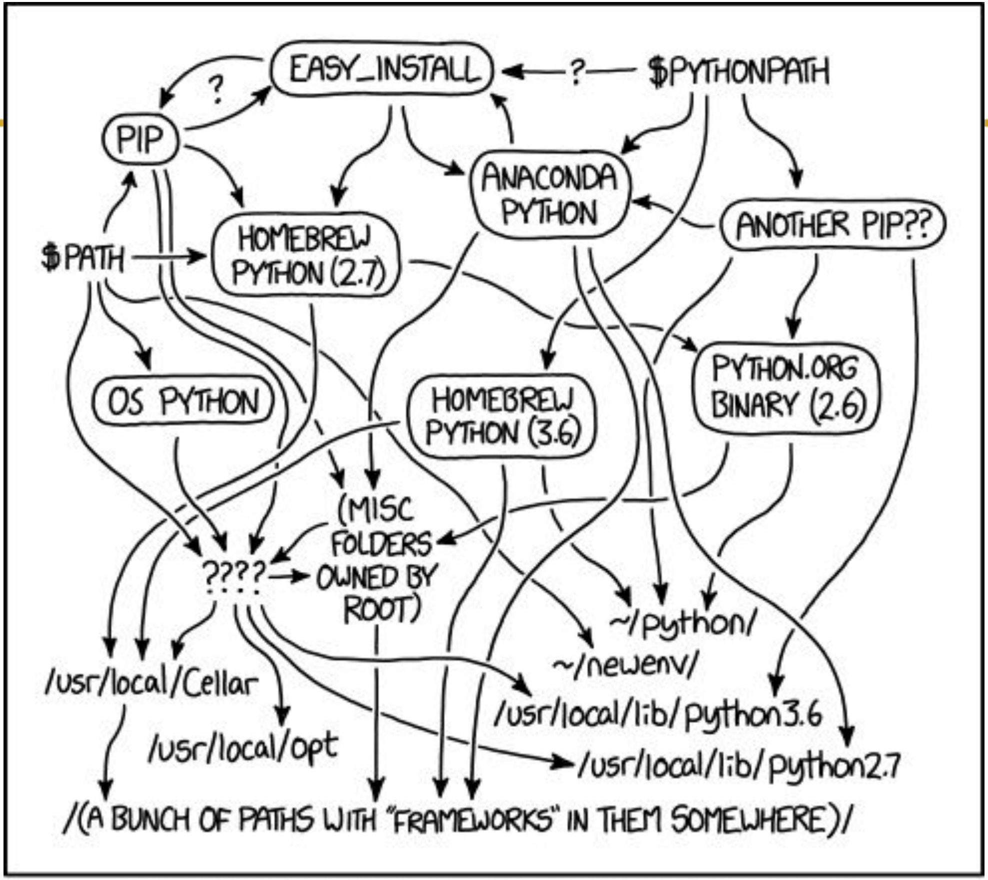
User’s View Of Common Path To Installing Programs

- From Source
  - GNU autoconf (`./configure, make, make install`)
  - Cmake (`mkdir build, cd ./build, ccmake ..`)
- From Binary
  - OS Package Manager (deb, rpm)
  - Independant Package Manager (brew)
  - Standalone Package (flatpak, appimage)

Dependency Hell

- ✓ Packages with shared dependencies
- ✓ Depend on different versions of other packages
- ✓ Likely incompatible versions

How can we deploy software while mitigating this concern?

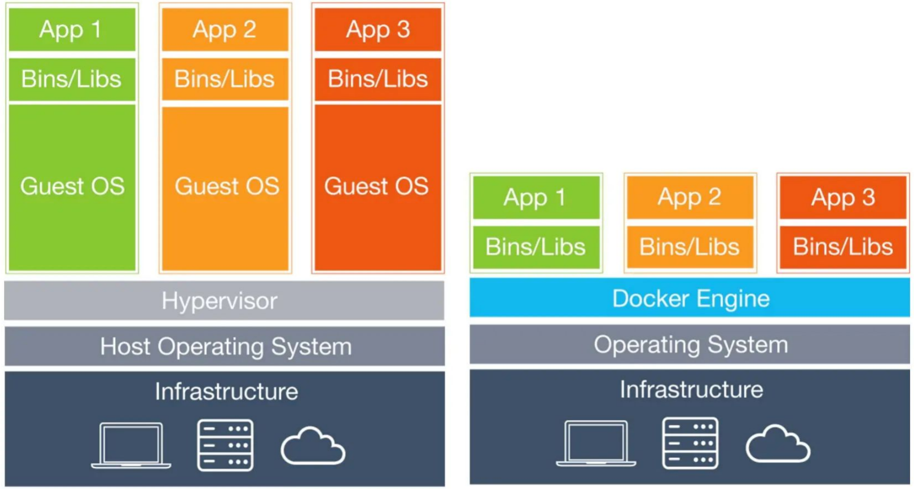


MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

R. Munroe, "Python environment," *xkcd*. [Online]. Available: https://xkcd.com/1987/. [Accessed: 29-Jun-2022].

Virtualization & Containers

- Virtualizations tools
  - Docker
  - VMWare vSphere Hypervisor
  - Virtualbox
  - QEMU
  - RedHat Virtualization
  - Oracle VM
  - KVM
  - LXD
  - Xvisor
  - **Apptainer (formerly Singularity)**
- Virtualization Management tools
  - Lots (includes kubernetes, docker swarm, vagrant)

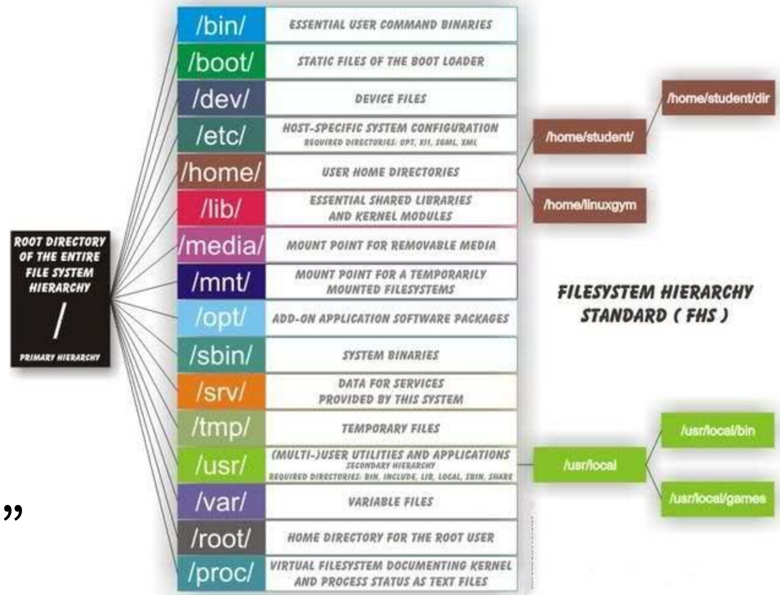


S. Carey, "What is Docker? the spark for the Container Revolution," *InfoWorld*, 02-Aug-2021. [Online]. Available: https://www.infoworld.com/article/3204171/what-is-docker-the-spark-for-the-container-revolution.html. [Accessed: 29-Jun-2022].

OS Level Installs

- Administrator access
  - usually unrealistic as a deployment post base install
  - Introduces significant security and management concerns

- Package management specific to distro
  - Redhat (CentOS, RHEL, Fedora)
    - RPM
      - `sudo dnf install package`
      - `sudo rpm -i "package.rpm"`
  - Debian (Ubuntu, Mobian)
    - DEB
      - `sudo apt install package`
      - `sudo dpkg -i “package.deb”`
  - Gentoo
    - EBUILD

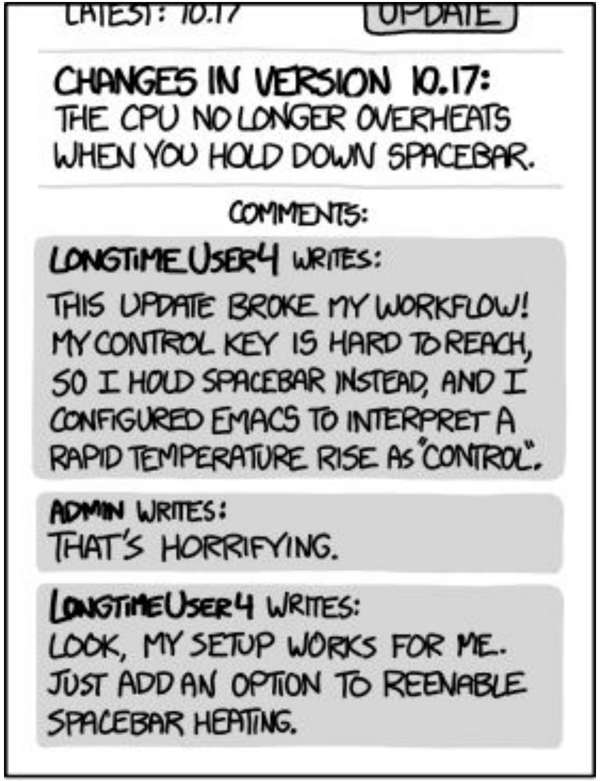

ROOT DIRECTORY OF THE ENTIRE FILE SYSTEM HIERARCHY
/
PRIMARY HIERARCHY
/bin/
ESSENTIAL USER COMMAND BINARIES
/boot/
STATIC FILES OF THE BOOT LOADER
/dev/
DEVICE FILES
/etc/
HOST-SPECIFIC SYSTEM CONFIGURATION
REQUIRED DIRECTORIES: OPT, X11, SGML, XML
/home/
USER HOME DIRECTORIES
/lib/
ESSENTIAL SHARED LIBRARIES AND KERNEL MODULES
/media/
MOUNT POINT FOR REMOVABLE MEDIA
/mnt/
MOUNT POINT FOR A TEMPORARILY MOUNTED FILESYSTEMS
/opt/
ADD-ON APPLICATION SOFTWARE PACKAGES
/sbin/
SYSTEM BINARIES
/srv/
DATA FOR SERVICES PROVIDED BY THIS SYSTEM
/tmp/
TEMPORARY FILES
/usr/
(MULTI-)USER UTILITIES AND APPLICATIONS
SECONDARY HIERARCHY
REQUIRED DIRECTORIES: BIN, INCLUDE, LIB, LOCAL, SBIN, SHARE
/var/
VARIABLE FILES
/root/
HOME DIRECTORY FOR THE ROOT USER
/proc/
VIRTUAL FILESYSTEM DOCUMENTING KERNEL AND PROCESS STATUS AS TEXT FILES
/home/student/dir
/home/student/
/home/linuxgym
FILESYSTEM HIERARCHY STANDARD (FHS)
/usr/local/bin
/usr/local
/usr/local/games

Editor, D. Smenov, E. Tsoutsouris, and Tim, “Editor,” Linux Directory Structure and Important Files Paths Explained, 07-Jan-2015. [Online]. Available: https://www.tecmint.com/linux-directory-structure-and-important-files-paths-explained/. [Accessed: 29-Jun-2022].

Environment Modules

- Alternatively, change software at the user level

- Allow user to load and unload program settings
  - Tcl/c (Tcl)
  - Lmod (Lua)


LATEST: 10.17
UPDATE
CHANGES IN VERSION 10.17:
THE CPU NO LONGER OVERHEATS WHEN YOU HOLD DOWN SPACEBAR.
COMMENTS:
LONGTIME USER4 WRITES:
THIS UPDATE BROKE MY WORKFLOW! MY CONTROL KEY IS HARD TO REACH, SO I HOLD SPACEBAR INSTEAD, AND I CONFIGURED EMACS TO INTERPRET A RAPID TEMPERATURE RISE AS "CONTROL".
ADMIN WRITES:
THAT'S HORRIFYING.
LONGTIMEUSER4 WRITES:
LOOK, MY SETUP WORKS FOR ME. JUST ADD AN OPTION TO REENABLE SPACEBAR HEATING.

EVERY CHANGE BREAKS SOMEONE'S WORKFLOW.

R. Munroe, "Workflow" *xkcd*. [Online]. Available: https://xkcd.com/1172/. [Accessed: 29-Jun-2022].

Automated Building Single Programs

- Rely on external package management
  - ./Configure && Make
  - CMake
  - Scons

- Includes some package management
  - Pip (wheel)
  - Conda (forge)

- As part of package management
  - EasyBuild
  - Spack
  - Brew

INSTALL.SH

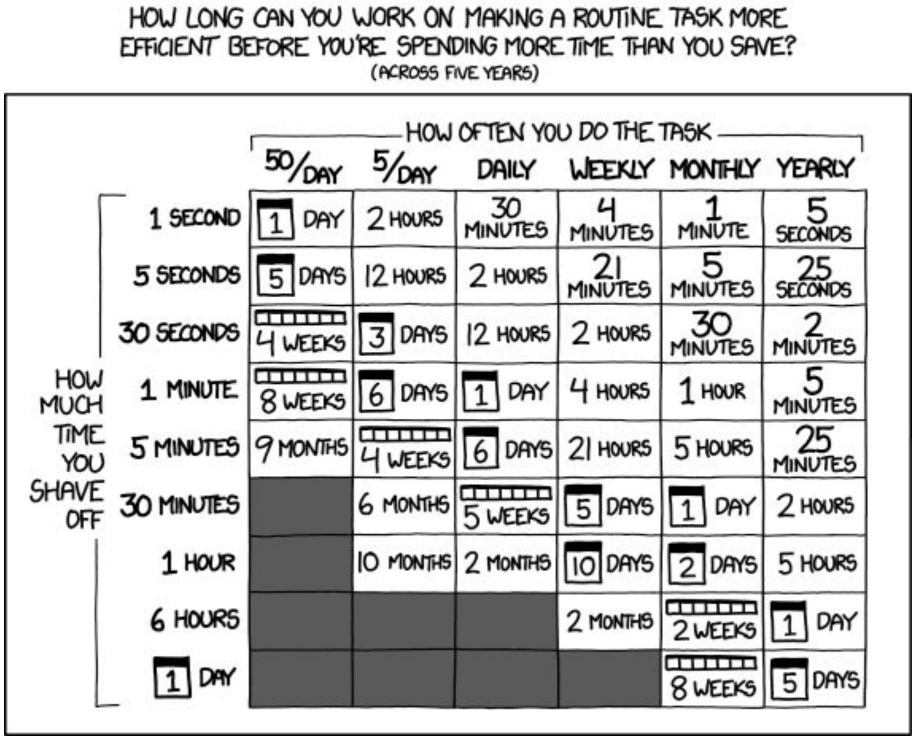
```
#!/bin/bash

pip install "$1" &
easy_install "$1" &
brew install "$1" &
npm install "$1" &
yum install "$1" & dnf install "$1" &
docker run "$1" &
pkg install "$1" &
apt-get install "$1" &
sudo apt-get install "$1" &
steamcmd +app_update "$1" validate &
git clone https://github.com/"$1"/"$1" &
cd "$1";./configure;make;make install &
curl "$1" | bash &
```

R. Munroe, “Universal Install Script” *xkcd*. [Online]. Available: https://xkcd.com/1654/. [Accessed: 29-Jun-2022].

Automation

Is automation right for deploying my community code?



HOW LONG CAN YOU WORK ON MAKING A ROUTINE TASK MORE EFFICIENT BEFORE YOU'RE SPENDING MORE TIME THAN YOU SAVE? (ACROSS FIVE YEARS)

		HOW OFTEN YOU DO THE TASK					
		50/DAY	5/DAY	DAILY	WEEKLY	MONTHLY	YEARLY
HOW MUCH TIME YOU SHAVE OFF	1 SECOND	1 DAY	2 HOURS	30 MINUTES	4 MINUTES	1 MINUTE	5 SECONDS
	5 SECONDS	5 DAYS	12 HOURS	2 HOURS	21 MINUTES	5 MINUTES	25 SECONDS
	30 SECONDS	4 WEEKS	3 DAYS	12 HOURS	2 HOURS	30 MINUTES	2 MINUTES
	1 MINUTE	8 WEEKS	6 DAYS	1 DAY	4 HOURS	1 HOUR	5 MINUTES
	5 MINUTES	9 MONTHS	4 WEEKS	6 DAYS	21 HOURS	5 HOURS	25 MINUTES
	30 MINUTES	▨	6 MONTHS	5 WEEKS	5 DAYS	1 DAY	2 HOURS
	1 HOUR	▨	10 MONTHS	2 MONTHS	10 DAYS	2 DAYS	5 HOURS
	6 HOURS	▨	▨	▨	2 MONTHS	2 WEEKS	1 DAY
	1 DAY	▨	▨	▨	▨	8 WEEKS	5 DAYS

R. Munroe, “Is It Worth the Time?” *xkcd*. [Online]. Available: https://xkcd.com/1205/. [Accessed: 29-Jun-2022].

Previously Covered In Earlier VRPs

- ✓ Autotools ./configure && Make
- ✓ Redhat Package Management (RPM)
- ✓ Spack
- ✓ EasyBuild

Debian Package (.DEB)

- Software management via apt (apt-get, aptitude, synaptic)
- Native to Ubuntu Server (Although you may also want to use snap instead)
- If you run Mobian, can even run on your linux phone
- To build your own
  - Create a folder with the necessary software
  - Write package description file (CONTROL) →
  - Write package post install file (Postinst)
  - `chmod 775 mypackage/DEBIAN/postinst`
  - `dpkg-deb --build mypackage`
  - `sudo apt-get install ./mypackage.deb`

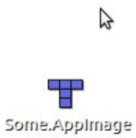
```
Package: grep
 Essential: yes
 Priority: required
 Section: base
 Maintainer: Wichert Akkerman <wakkerma@debian.org>
 Architecture: sparc
 Version: 2.4-1
 Pre-Depends: libc6 (>= 2.0.105)
 Provides: rgrep
 Conflicts: rgrep
 Description: GNU grep, egrep and fgrep.
  The GNU family of grep utilities may be the "fastest grep in the west".
  GNU grep is based on a fast lazy-state deterministic matcher (about
  twice as fast as stock Unix egrep) hybridized with a Boyer-Moore-Gosper
  search for a fixed string that eliminates impossible text from being
  considered by the full regexp matcher without necessarily having to
  look at every character. The result is typically many times faster
  than Unix grep or egrep. (Regular expressions containing backreferencing
  will run more slowly, however).
```

Mobian. [Online]. Available: https://mobian-project.org/. [Accessed: 29-Jun-2022].

Deb-Control(5) - linux manual page. [Online]. Available: https://www.man7.org/linux/man-pages/man5/deb-control.5.html. [Accessed: 29-Jun-2022].

AppImage Package (.AppImage)

- Software management via file-system (sort of)
- Linux Distribution Agnostic (mostly)
- Program is self-contained
- To build your own
  - Create a folder with the necessary software
  - ./appimagetool-x86_64.AppImage ./mypackage/
  - Can also be built with appimage-builder
  - `chmod +x mypackage.AppImage`
  - Then just run it with: `./my.AppImage`



"Appimage," AppImage. [Online]. Available: https://appimage.org/. [Accessed: 29-Jun-2022].

"Quickstart," Quickstart - AppImage documentation, 25-Nov-2020. [Online]. Available: https://docs.appimage.org/introduction/quickstart.html#how-to-run-an-appimage. [Accessed: 29-Jun-2022].

Singularity / Apptainer

- Last year was forked and merged into The Linux Foundation
- https://github.com/apptainer/apptainer
- From their website:
  - “An immutable single-file container image format, supporting cryptographic signatures and encryption.”
  - “Integration over isolation by default. Easily make use of GPUs, high speed networks, parallel filesystems on a cluster or server.”
  - “Mobility of compute. The single file SIF container format is easy to transport and share.”
  - “A simple, effective security model. You are the same user inside a container as outside, and cannot gain additional privilege on the host system by default.”

“Home,” *Apptainer*. [Online]. Available: https://apptainer.org/. [Accessed: 28-Jun-2022].

Apptainer, “Apptainer/apptainer: Apptainer: Application containers for linux,” GitHub. [Online]. Available: https://github.com/apptainer/apptainer. [Accessed: 29-Jun-2022].

Building For Multiple Architectures

easybuild

- ✓ https://easybuild.io/
- ✓ Well documented
- ✓ Can be used without root
- ✓ Built specifically for scientific programs and hpc
- ✓ Uses prebuilt build recipes

Spack

- ✓ https://spack.io/
- ✓ Well documented
- ✓ Can be used without root
- ✓ Built specifically for scientific programs and hpc
- ✓ Uses prebuilt build recipes

“Building software with ease,” *EasyBuild*. [Online]. Available: https://easybuild.io/. [Accessed: 29-Jun-2022].
Spack. [Online]. Available: https://spack.io/. [Accessed: 29-Jun-2022].

Deploying FOSS (Highly Generalized)

1. Verify the requirements of the project, and that the software package meets those needs including in both functionality and in license.
2. Identify the mainline version of the software to ensure that the correct fork is being evaluated.
3. Verify that the resources available are sufficient for the software's deployment and use, and identify system environment in which it is being deployed.
4. Identify the documentation and other resources for the software package, reading relevant material as necessary (such as known issues, security concerns, or configuration and deployment parameters).

Deploying FOSS (Highly Generalized)

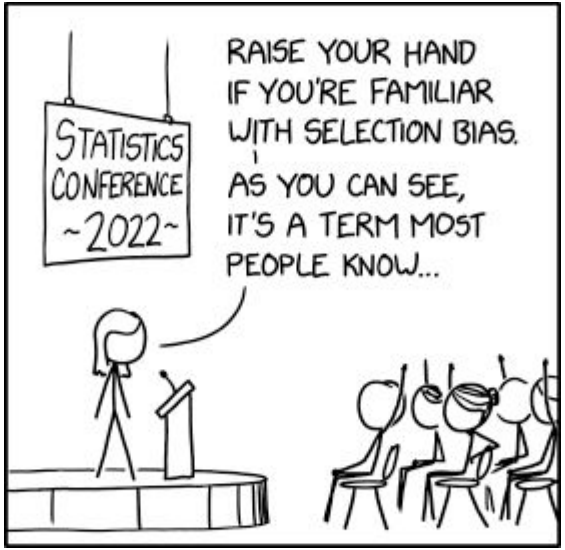
5. Actively keep an eye-out while evaluating the documentation for alternative software solutions which may address the users need more optimally, especially if the user is not as familiar with the related technology. If anything has identified, proceed as requested but additionally inform the user of the alternative solution. If the user wishes to change to the other option do so and continue otherwise.
6. Identify any necessary processes for the upgrade / deployment, and if necessary, how to safely recover quickly if a rollback as needed.
7. Identify that the user's have all necessary authorizations as required by policy, what support is needed, and the software's ownership and access, and life cycle on the university's resources.

Deploying FOSS (Highly Generalized)

8. Once the deployment process is then generally understood, follow the necessary deployment process.
9. When possible keep issues tracked and any patches in software or documentation pushed back upstream for others users.
10. Finally, schedule a follow-up for assessing the software's continued use / deployment and any needs of the researcher, department, or university.

Don’t Forget To Ask Questions

- Ask questions if you don’t understand the documentation
  - It tends to be those who write the documentation are not those who need it.
- Finding A Community
  - Does the project have a main page?
  - Is there a slack / discord / IRC / forum / user-group?
  - Are others at your institution using this software (or find another institution you can ask)


STATISTICS CONFERENCE ~2022~
RAISE YOUR HAND IF YOU'RE FAMILIAR WITH SELECTION BIAS.
AS YOU CAN SEE, IT'S A TERM MOST PEOPLE KNOW...

R. Munroe, “Selection Bias” *xkcd*. [Online]. Available: https://xkcd.com/2618/. [Accessed: 29-Jun-2022].

References (1 of 2)

1. Apptainer, “Apptainer/apptainer: Apptainer: Application containers for linux,” GitHub. [Online]. Available: https://github.com/apptainer/apptainer. [Accessed: 29-Jun-2022].
2. Deb-Control(5) - linux manual page. [Online]. Available: https://www.man7.org/linux/man-pages/man5/deb-control.5.html. [Accessed: 29-Jun-2022].
3. Editor, D. Smenov, E. Tsoutsouris, and Tim, “Editor,” Linux Directory Structure and Important Files Paths Explained, 07-Jan-2015. [Online]. Available: https://www.tecmint.com/linux-directory-structure-and-important-files-paths-explained/. [Accessed: 29-Jun-2022].
4. Joshua Alexander, “Deploying Community Codes” Presentation and Demo, 2015 Virtual Residency Program (VRP) workshop, Online and In-Person, June 1, 2015.
5. Martin Čuma, “Deploying Community Codes” Presentation and Demo, 2016 Virtual Residency Program (VRP) workshop, Online and In-Person, August 8, 2016.
6. Mobian. [Online]. Available: https://mobian-project.org/. [Accessed: 29-Jun-2022].
7. Prasad Maddumage, “Deploying Community Codes” Presentation and Workshop, 2017 Virtual Residency Program (VRP) workshop, Online and In-Person, July 31, 2017.
8. R. Munroe, “Data Trap,” xkcd. [Online]. Available: https://xkcd.com/2582/. [Accessed: 29-Jun-2022].
9. R. Munroe, “Dependency,” xkcd. [Online]. Available: https://xkcd.com/2347/. [Accessed: 29-Jun-2022].
10. R. Munroe, “Is It Worth the Time?” xkcd. [Online]. Available: https://xkcd.com/1205/. [Accessed: 29-Jun-2022].

References (2 of 2)

11. R. Munroe, “Selection Bias” xkcd. [Online]. Available: https://xkcd.com/2618/. [Accessed: 29-Jun-2022].
12. R. Munroe, “Universal Install Script” xkcd. [Online]. Available: https://xkcd.com/1654/. [Accessed: 29-Jun-2022].
13. R. Munroe, “Wisdom of the Ancients,” xkcd. [Online]. Available: https://xkcd.com/979/. [Accessed: 29-Jun-2022].
14. R. Munroe, “Workflow” xkcd. [Online]. Available: https://xkcd.com/1172/. [Accessed: 29-Jun-2022].
15. Spack. [Online]. Available: https://spack.io/. [Accessed: 29-Jun-2022]
16. S. Carey, “What is Docker? the spark for the Container Revolution,” InfoWorld, 02-Aug-2021. [Online]. Available: https://www.infoworld.com/article/3204171/what-is-docker-the-spark-for-the-container-revolution.html. [Accessed: 29-Jun-2022].
17. “Appimage,” AppImage. [Online]. Available: https://appimage.org/. [Accessed: 29-Jun-2022].
18. “Building software with ease,” EasyBuild. [Online]. Available: https://easybuild.io/. [Accessed: 29-Jun-2022].
19. “Home,” Apptainer. [Online]. Available: https://apptainer.org/. [Accessed: 28-Jun-2022].
20. “Quickstart,” Quickstart - AppImage documentation, 25-Nov-2020. [Online]. Available: https://docs.appimage.org/introduction/quickstart.html#how-to-run-an-appimage. [Accessed: 29-Jun-2022].

Thank you for your time.

Questions?

32