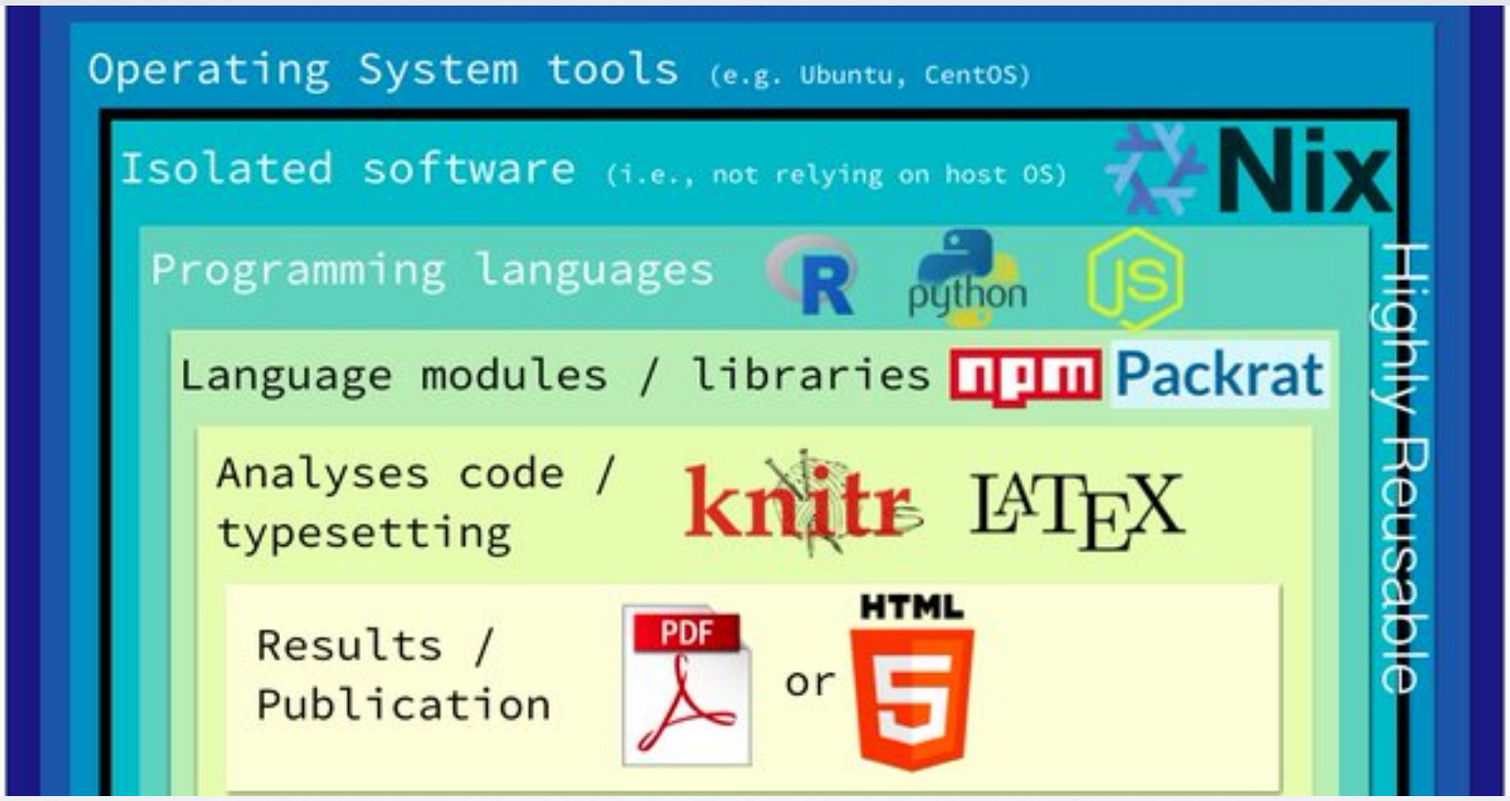
# A TRULY REPRODUCIBLE PAPER?




by Bruno Vieira | @bmpvieira

Slides: `http://bit.ly/re-paper-nix`

# QUICK INTRO

# WHO?

Bruno Vieira

@bmpvieira

http://bit.ly/re-paper-nix

# WHO?

Bruno Vieira

@bmpvieira

Finishing PhD

Pop. Genomics


http://bit.ly/re-paper-nix

# WHO?

Bruno Vieira

@bmpvieira

Finishing PhD
Pop. Genomics

2016 Fellow
(Champion Open Science)


mozilla
Science Lab

http://bit.ly/re-paper-nix

# WHO?

Bruno Vieira

@bmpvieira

Finishing PhD
Pop. Genomics


Founder of the
**bionode.io**
community


2016 Fellow
(Champion Open Science)


Highly reusable code and tools for genomics (powered by Node.JS)

`http://bit.ly/re-paper-nix`

SCIENCE

should be

REPRODUCIBLE

and

REUSABLE

# CONTROLLING ALL VARIABLES IS HARD!!

## Field work


https://en.wikipedia.org/wiki/File:Zoobentos_sampling_Krippenbach.jpg

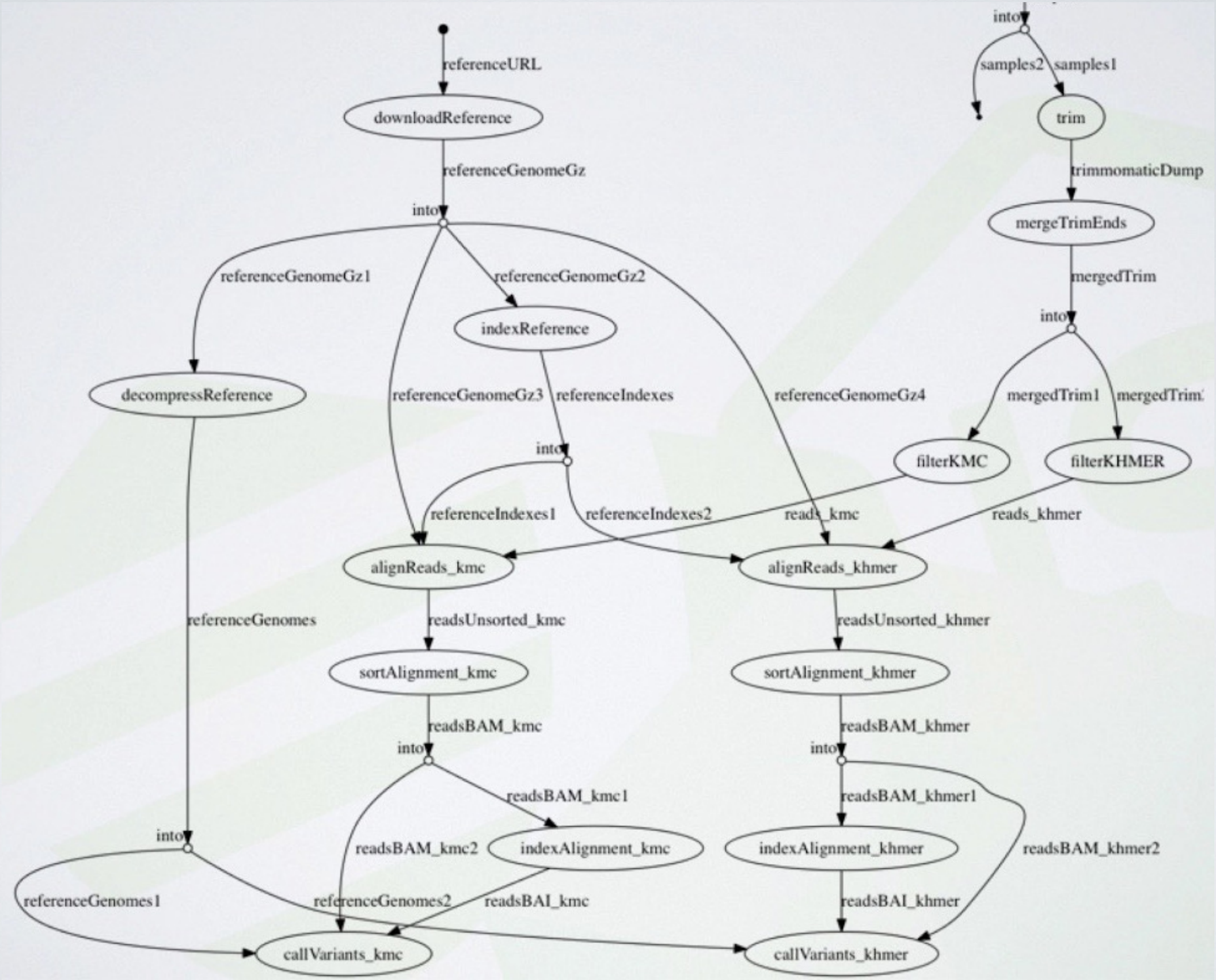
## Lab work


https://en.wikipedia.org/wiki/File:Micropipette_in_action.jpg

http://bit.ly/re-paper-nix

# EASIER FOR COMPUTING? NO!!


http://bit.ly/re-paper-nix


referenceURL
downloadReference
referenceGenomeGz
into
referenceGenomeGz1
referenceGenomeGz2
referenceGenomeGz3
referenceGenomeGz4
decompressReference
indexReference
referenceIndexes
into
referenceIndexes1
referenceIndexes2
referenceGenomes
into
referenceGenomes1
referenceGenomes2
into
samples2
samples1
trim
trimmomaticDump
mergeTrimEnds
mergedTrim
into
mergedTrim1
mergedTrim
filterKMC
filterKHMER
reads_kmc
reads_khmer
alignReads_kmc
alignReads_khmer
readsUnsorted_kmc
readsUnsorted_khmer
sortAlignment_kmc
sortAlignment_khmer
readsBAM_kmc
readsBAM_khmer
into
into
readsBAM_kmc1
readsBAM_khmer1
readsBAM_kmc2
indexAlignment_kmc
indexAlignment_khmer
readsBAM_khmer2
readsBAI_kmc
readsBAI_khmer
callVariants_kmc
callVariants_khmer

# REPRODUCIBLE PAPER

Results / Publication  or 


http://bit.ly/re-paper-nix

# REPRODUCIBLE PAPER

Typesetting and analysis code mixed:


http://bit.ly/re-paper-nix

# REPRODUCIBLE PAPER

Typesetting and analysis code mixed:

Allows to see exactly how we got to a result/figure


http://bit.ly/re-paper-talk

# REPRODUCIBLE PAPER

Typesetting and analysis code mixed:

Allows to see exactly how we got to a result/figure

The future: interactive notebooks (Jupyter, Beaker, nteract)


http://bit.ly/re-paper-talk

# REPRODUCIBLE CODE

Use whatever is necessary to manage language level dependencies

**Python**: pip; **Node.JS**: npm; **R**: Packrat


http://bit.ly/re-paper-nix

# REPRODUCIBLE CODE

```
npm install libmycodedepends --save
npm shrinkwrap
```

```
pip freeze > requirements.txt
```

Language modules / libraries

Analyses code / typesetting

Results / Publication

or

http://bit.ly/re-paper-talk

# REPRODUCIBLE CODE

```
install.packages('libmycodedepends')
packrat::snapshot()
```

Language modules / libraries npm Packrat

Analyses code / typesetting knitr LaTeX

Results / Publication  or 

http://bit.ly/re-paper-talk

# REPRODUCIBLE ENVIRONMENT

Keep track of languages versions and install/compile options


http://bit.ly/re-paper-nix

# REPRODUCIBLE ENVIRONMENT

Python 2 or 3? Which R or Node version? Compile flags?


http://bit.ly/re-paper-talk

# REPRODUCIBLE ENVIRONMENT

/bin/python? /usr/bin/python? /bin/env python?

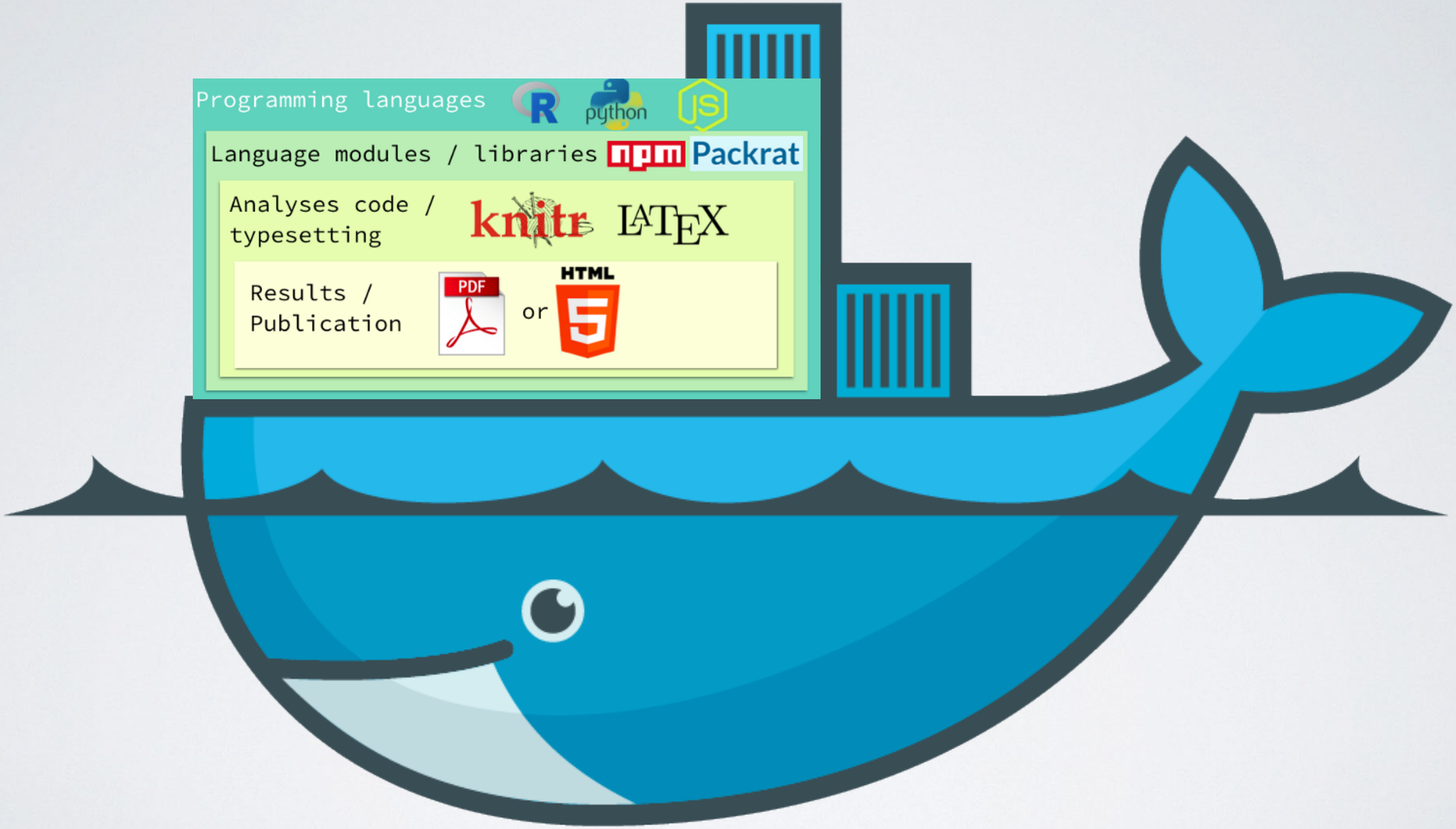

http://bit.ly/re-paper-talk

# REPRODUCIBLE ENVIRONMENT

Ubuntu 16.04 or 17.10? CentOS?

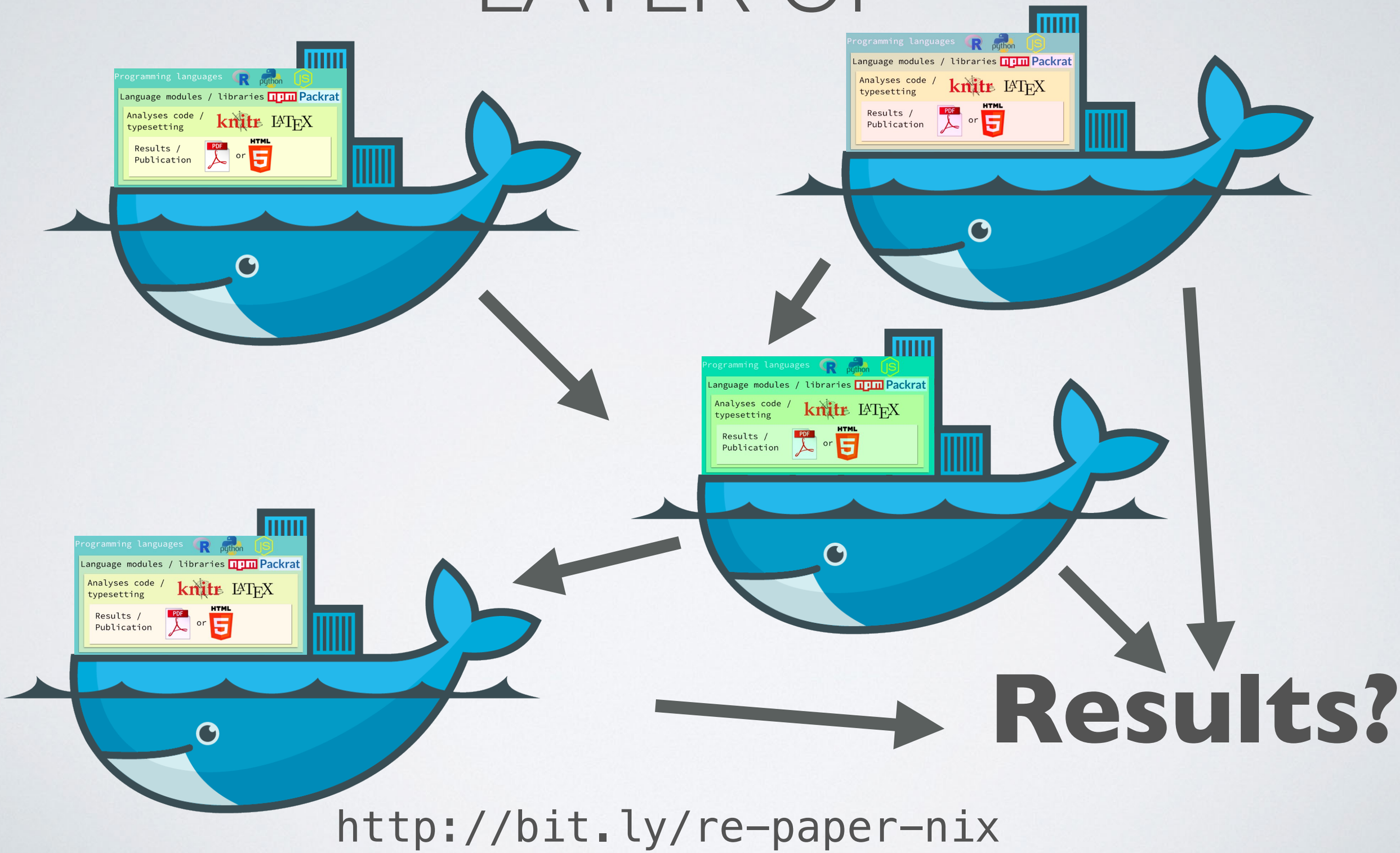

http://bit.ly/re-paper-talk

# CONTAINERS AND VM




http://bit.ly/re-paper-nix

# JUST MOVE THE PROBLEM ONE LAYER UP

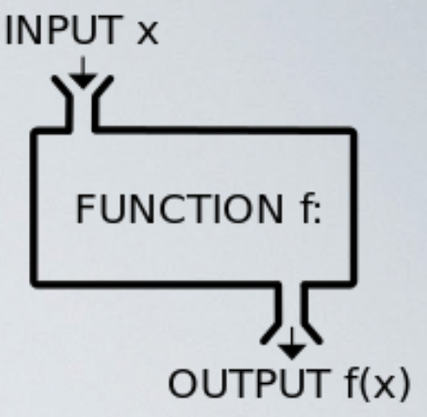



http://bit.ly/re-paper-nix

# MORE REUSABLE SOLUTION


http://bit.ly/re-paper-nix

# Functional and deterministic package manager




## Nix

### The Purely Functional Package Manager

Nix is a powerful package manager for Linux and other Unix systems that makes package management reliable and reproducible. It provides atomic upgrades and rollbacks, side-by-side installation of multiple versions of a package, multi-user package management and easy setup of build environments. Read more…

Get Nix

## Reliable

Nix's purely functional approach ensures that installing or upgrading one package cannot break other packages. This is because it won't overwrite dependencies with newer versions that might cause breakage elsewhere. It allows you to roll back to previous versions, and ensures that no package is in an inconsistent state during an upgrade.

## Reproducible

Nix builds packages in isolation from each other. This ensures that they are reproducible and don't have undeclared dependencies, so if a package works on one machine, it will also work on another.

## Great for developers

Nix makes it trivial to set up and share build environments for your projects, regardless of what programming languages and tools you're using. For instance, running the command "`nix-shell '<nixpkgs>' -A firefox`" gives you a Bash shell in which all of Firefox's build-time dependencies are present and all necessary environment variables are set.

## Multi-user, multi-version

Nix supports multi-user package management: multiple users can share a common Nix store securely, don't need to have root privileges to install software, and can install and use different versions of a package.

## Source/binary model

Conceptually, Nix builds packages from source, but can transparently use binaries from a *binary cache* if available. This combines the flexibility of source package management with the convenience of binary package management.

## Portable

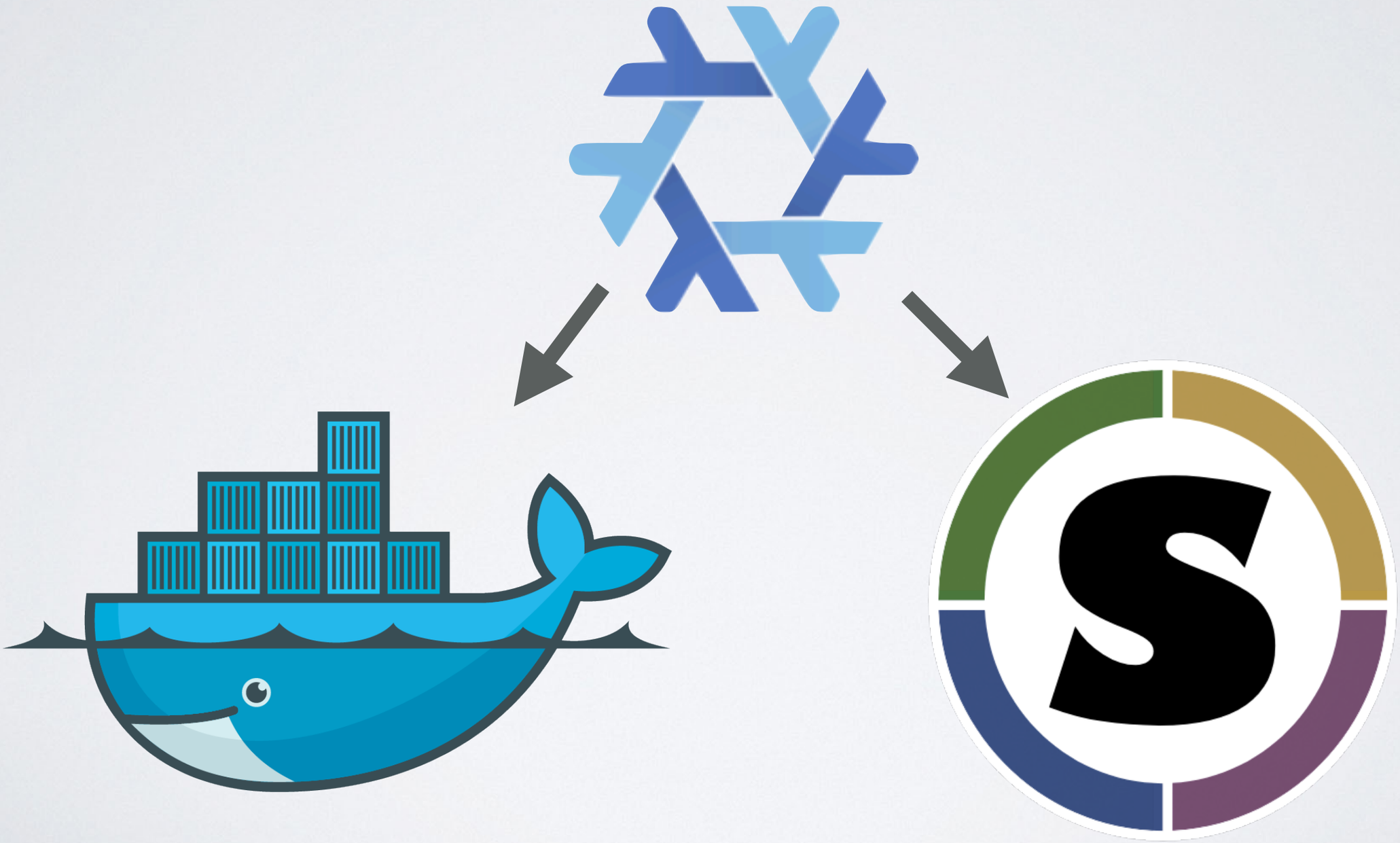
Nix runs on Linux, Mac OS X and other systems. Nixpkgs, the Nix Packages collection, contains thousands of packages, many pre-compiled.

How? Each folder is a hash of
the package dependencies and configurations

```
> ls /nix/store/ | grep python | head -n 5
1723sa96bsp9cxzpn0l2c7faffxvwk2n-python2.7-docutils-0.12.drv
1h31h196ap5qvnbz6jbzn6f4afn67yz7-python2.7-unittest2-1.1.0.drv
33y24q9lcgnxmz9hkwiw8iyl3xcssxdl-python2.7-Pygments-2.1.3.drv
396zfq6rpvy83mlkgsk1hfnvc70xsmvk-python2.7-pytest-2.9.2.drv
412fz04cap48p6m2wsvg9fs4x8ym7h85-python2.7-numpy-1.11.2.drv
```

Easily add them automatically to your PATH
or use in isolated environments per project

# Easily generate lightweight containers with Nix to run on current HPC infrastructures



Powerful tool for advanced users

https://nixos.org/nix/manual

https://nixos.org/nixpkgs/manual/

Beginners blog post coming soon

https://medium.com/@bmpvieira

# HOW TO START USING IT RIGHT NOW!!!

# INSTALL NIX

```
› curl https://nixos.org/nix/install | sh
```

# INSTALL NIX

```
› curl https://nixos.org/nix/install | sh
```

Or if you worry about security and installing things from the web…

```
$ curl -o install-nix-1.11.15 https://nixos.org/nix/install
$ curl -o install-nix-1.11.15.sig https://nixos.org/nix/install.sig
$ gpg2 --recv-keys B541D55301270E0BCF15CA5D8170B4726D7198DE
$ gpg2 --verify ./install-nix-1.11.15.sig
$ sh ./install-nix-1.11.15
```

# SHEBANG

```
1    #! /bin/bash
2    echo Hello world!
3
```

# SHEBANG

```
1    #! /bin/bash
2    echo Hello world!
3
```

```
1    #! /usr/bin/env python
```

# SHEBANG

```
1    #! /bin/bash
2    echo Hello world!
3
```

```
1    #! /usr/bin/env python
```

```
1    #! /usr/bin/env Rscript
```

https://nixos.org/nix/manual/#use-as-a-interpreter

```
1    #! /usr/bin/env nix-shell
2    #! nix-shell -i real-interpreter -p packages
3
```

For too many packages to fit one line, use a **default.nix** file instead

# https://nixos.org/nix/manual/#use-as-a-interpreter

```
1    #! /usr/bin/env nix-shell
2    #! nix-shell -i real-interpreter -p packages
3
```

```
1    #! /usr/bin/env nix-shell
2    #! nix-shell -i python3 -p python3 python35Packages.matplotlib
```

For too many packages to fit one line, use a **default.nix** file instead

# https://nixos.org/nix/manual/#use-as-a-interpreter

```
1    #! /usr/bin/env nix-shell
2    #! nix-shell -i real-interpreter -p packages
3
```

```
1    #! /usr/bin/env nix-shell
2    #! nix-shell -i python3 -p python3 python35Packages.matplotlib
```

```
1    #! /usr/bin/env nix-shell
2    #! nix-shell -i bash -p python3 rPackages.ggplot2
```

For too many packages to fit one line, use a **default.nix** file instead

# ACKNOWLEDGMENTS


mozilla
Science Lab

Blog post: `http://bit.ly/re-paper`
Skill share: `http://bit.ly/re-paper-practical`
Slides: `http://bit.ly/re-paper-nix`