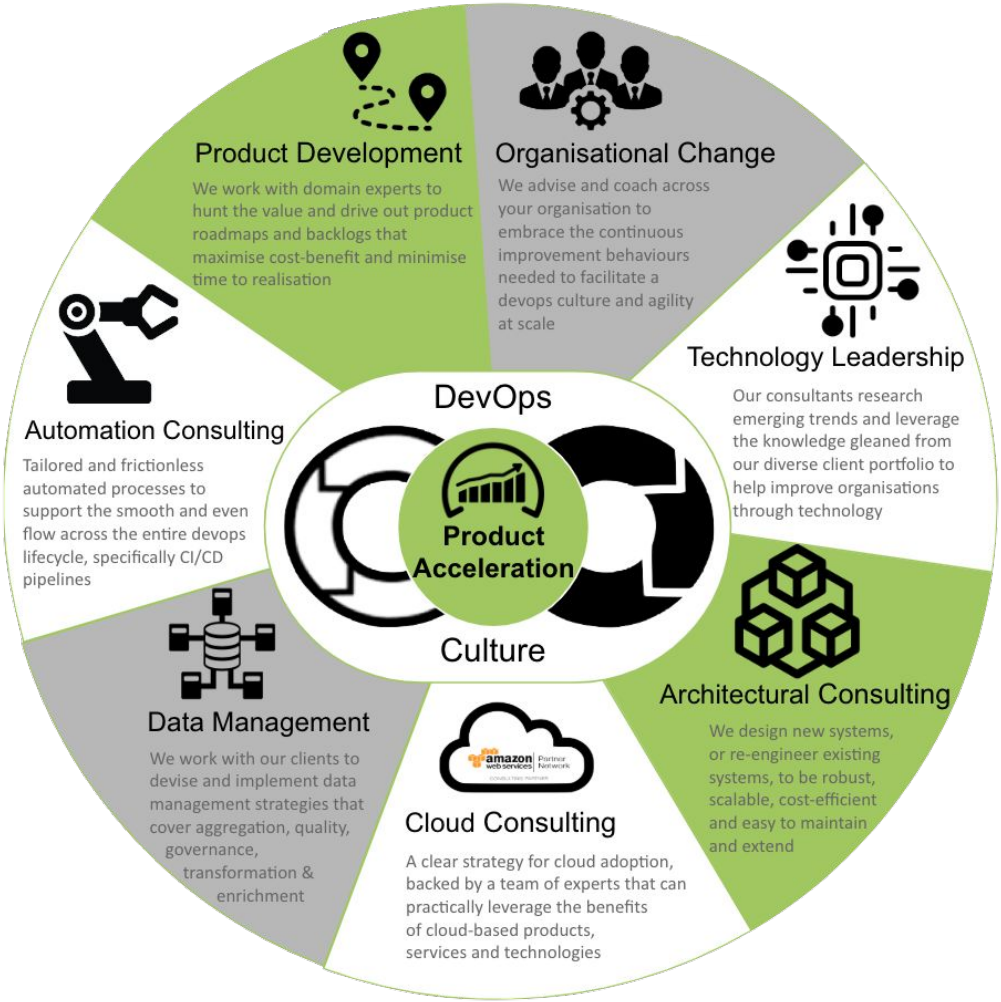
Docker Birmingham July 2019

Scientific Data Science

docker

Thanks to our Sponsor!

BlackCat /


Product Development
We work with domain experts to hunt the value and drive out product roadmaps and backlogs that maximise cost-benefit and minimise time to realisation
Organisational Change
We advise and coach across your organisation to embrace the continuous improvement behaviours needed to facilitate a devops culture and agility at scale
Technology Leadership
Our consultants research emerging trends and leverage the knowledge gleaned from our diverse client portfolio to help improve organisations through technology
Automation Consulting
Tailored and frictionless automated processes to support the smooth and even flow across the entire devops lifecycle, specifically CI/CD pipelines
DevOps
Product Acceleration
Culture
Data Management
We work with our clients to devise and implement data management strategies that cover aggregation, quality, governance, transformation & enrichment
Cloud Consulting
A clear strategy for cloud adoption, backed by a team of experts that can practically leverage the benefits of cloud-based products, services and technologies
Architectural Consulting
We design new systems, or re-engineer existing systems, to be robust, scalable, cost-efficient and easy to maintain and extend

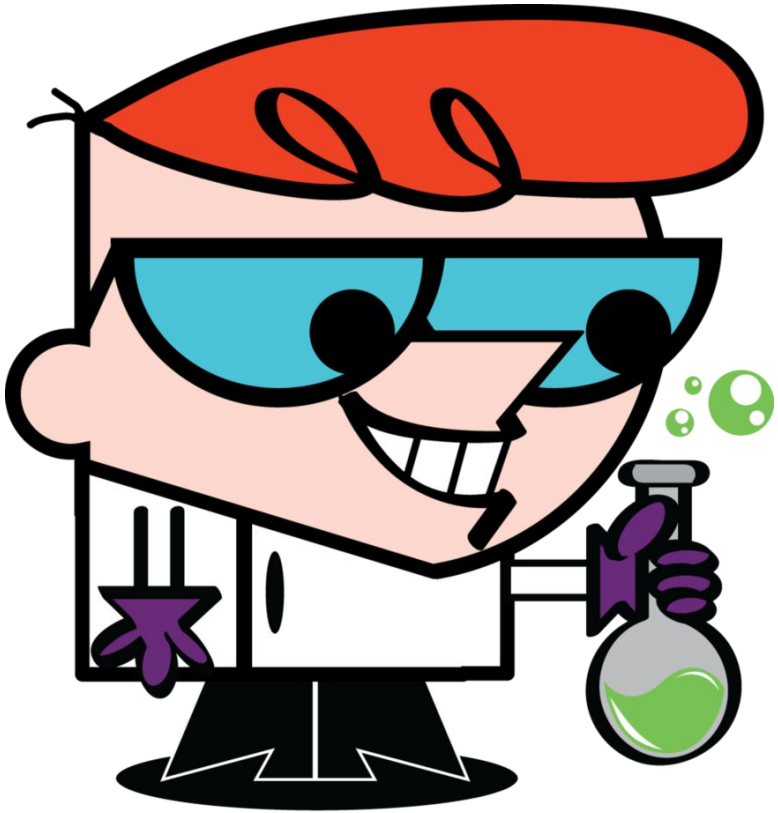
Aren’t we all…….

Next Meetup - th June 2019

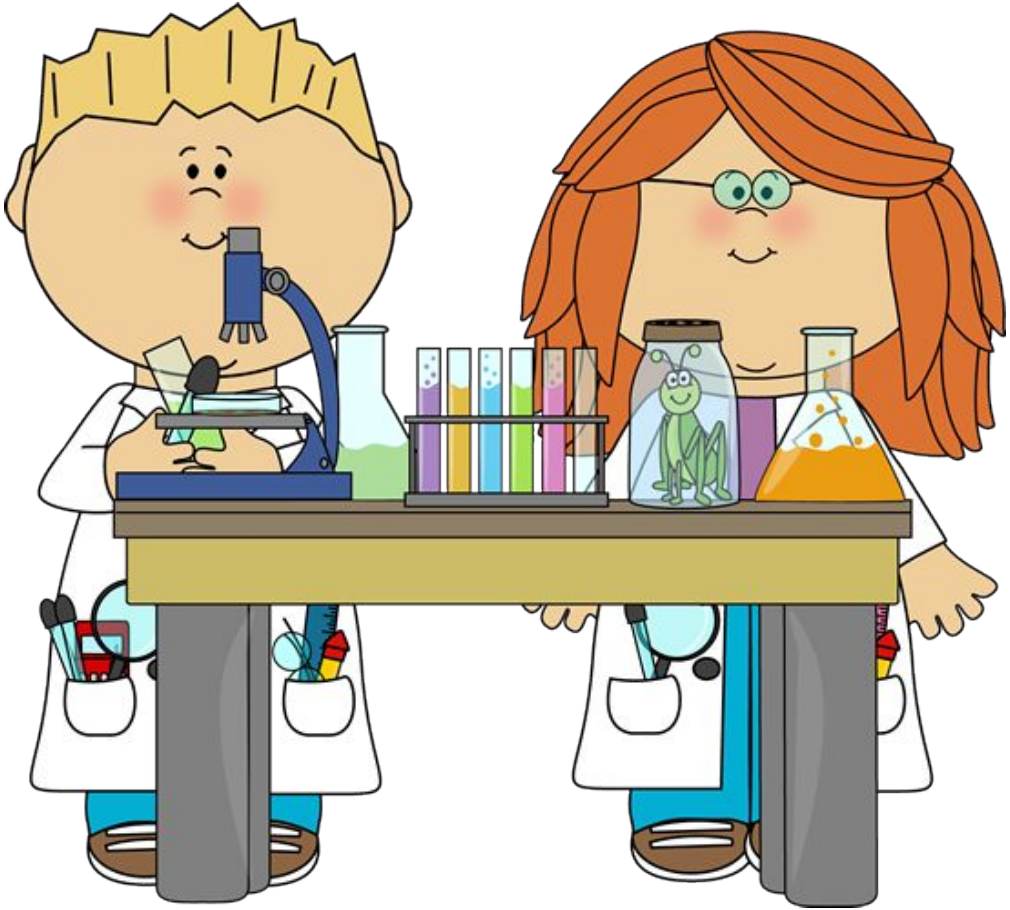
Science!

Science?

A systematic enterprise that builds and organizes knowledge in the form of testable explanations and predictions about the universe.



From this…….



To this!

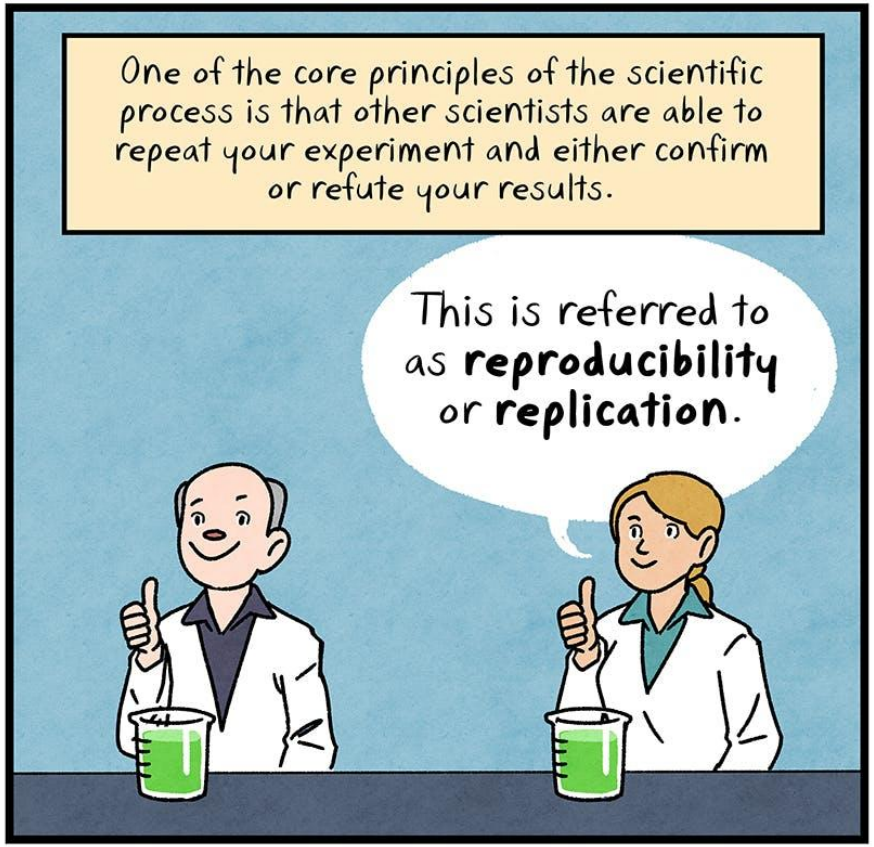
The scientific method seeks to objectively explain the events of nature in a reproducible way

The Scientific Method

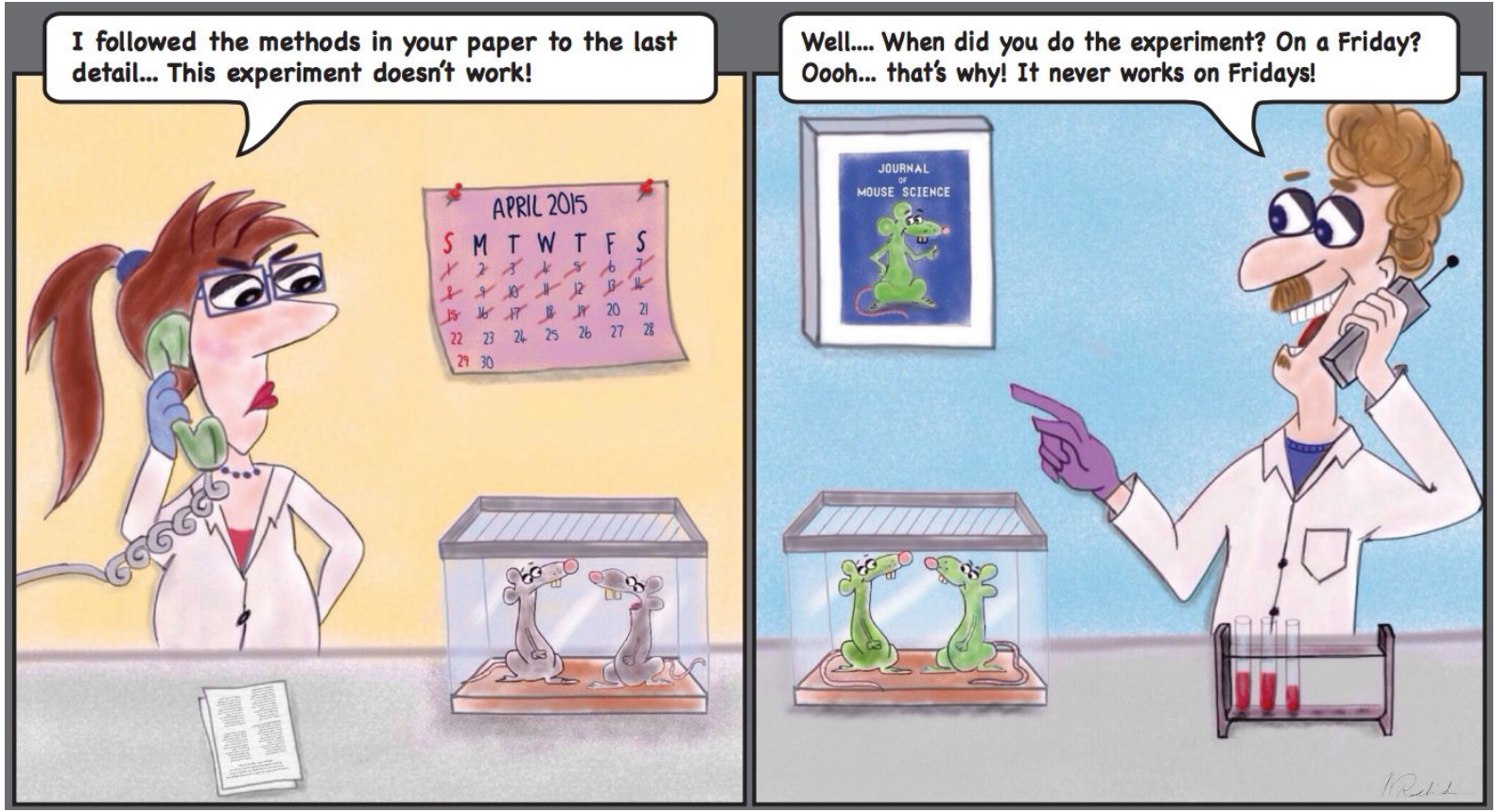
1. Define a question
2. Gather information and resources (observe)
3. Form an explanatory hypothesis
4. Test the hypothesis by performing an experiment and collecting data in a reproducible manner
5. Analyze the data
6. Interpret the data and draw conclusions that serve as a starting point for new hypothesis
7. Publish results
8. Retest (frequently done by other scientists)

Software Lifecycle TDD

Scientific Method

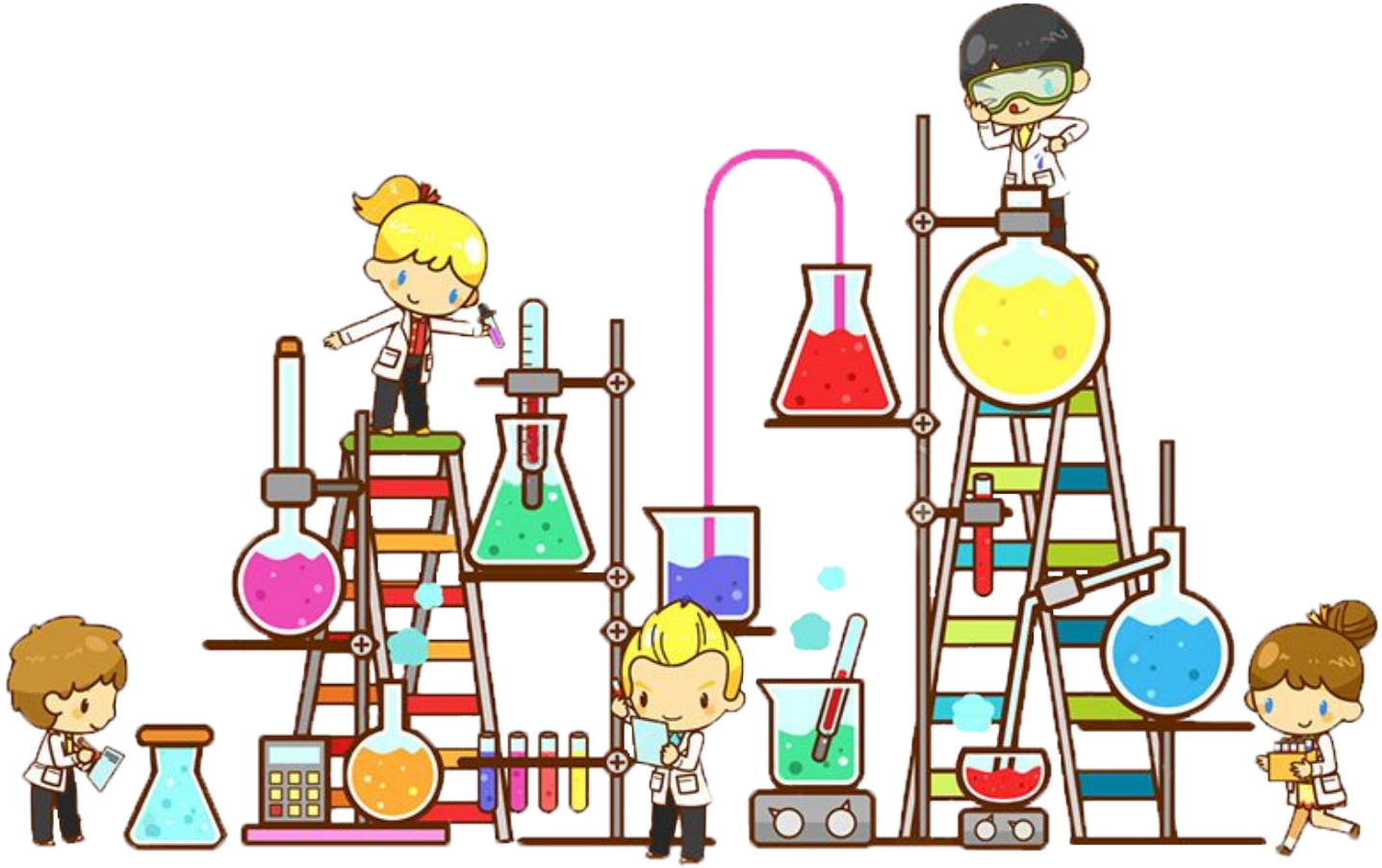


Reproducibility / Replication



Issues with Reproducibility

Designing ***Repeatable*** and ***Reproducible*** Experiments is ***hard.***

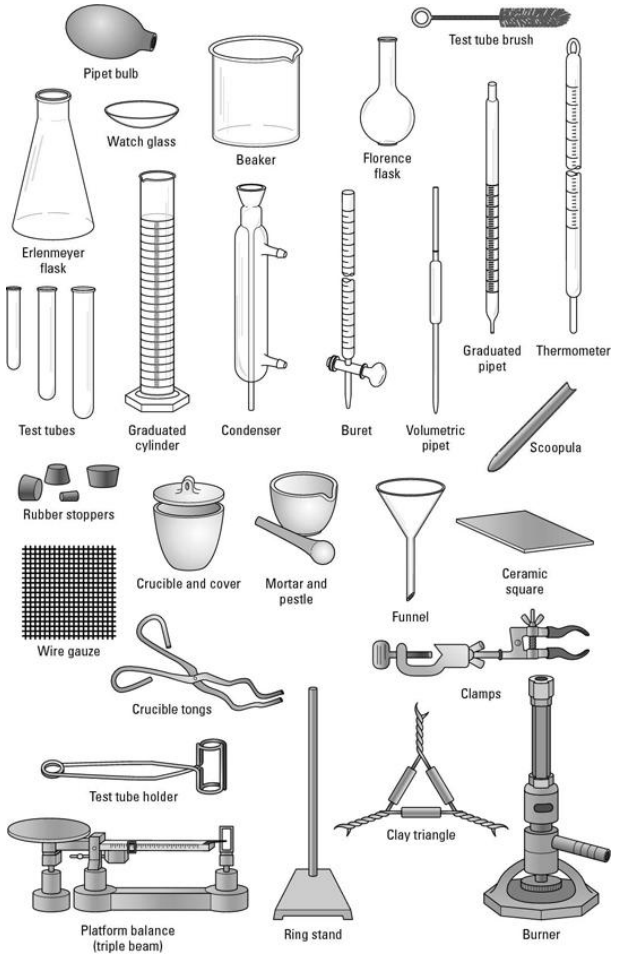


Physical Science Lab

How do we account for this?

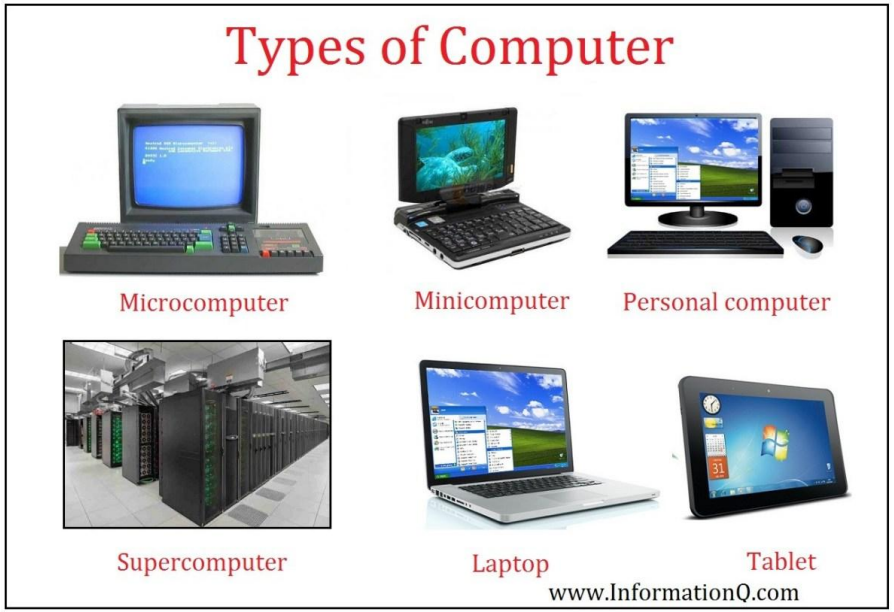
Physical Experiment Design

- Equipment types, sizes etc.
- Brand / Vendor
- Precursor Chemical types concentrations
- Environmental factors
- Stepwise Process
- Physical / Locations



Digital Experiment Design

- Type of Machine
- Version of Programming Language
- Version and state of OS
- Version of Libraries / Dependencies
- Data location & drift



Building a Digital Scientists Lab

Building a Digital Scientists Lab

- Git, Github, Gitlab
  - Distributed version control
  - Compare Files
  - Share Files
  - Collaborate
- Jupyter Notebooks
  - Browser based Document experiments
  - Document results
  - Interactive
  - Kernel support for many languages
  - Works neatly with VCS
  - Markdown, Graphs, LaTeX

Simple spectral analysis

An illustration of the Discrete Fourier Transform using windowing, to reveal the frequency content of a sound signal.

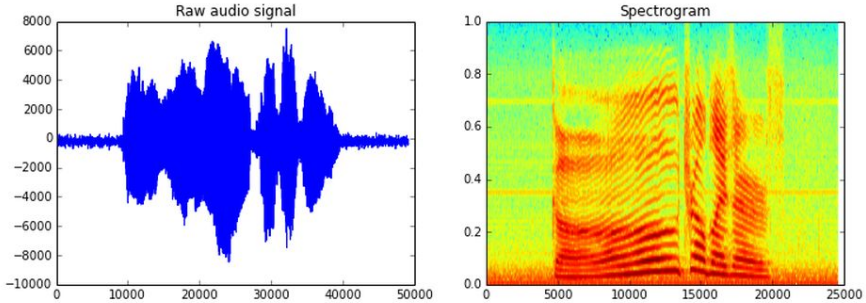
$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \qquad k = 0, \ldots, N-1$$

We begin by loading a datafile using SciPy's audio file support:

In [1]:
```
from scipy.io import wavfile
rate, x = wavfile.read('test_mono.wav')
```

And we can easily view its spectral structure using matplotlib's builtin `specgram` routine:

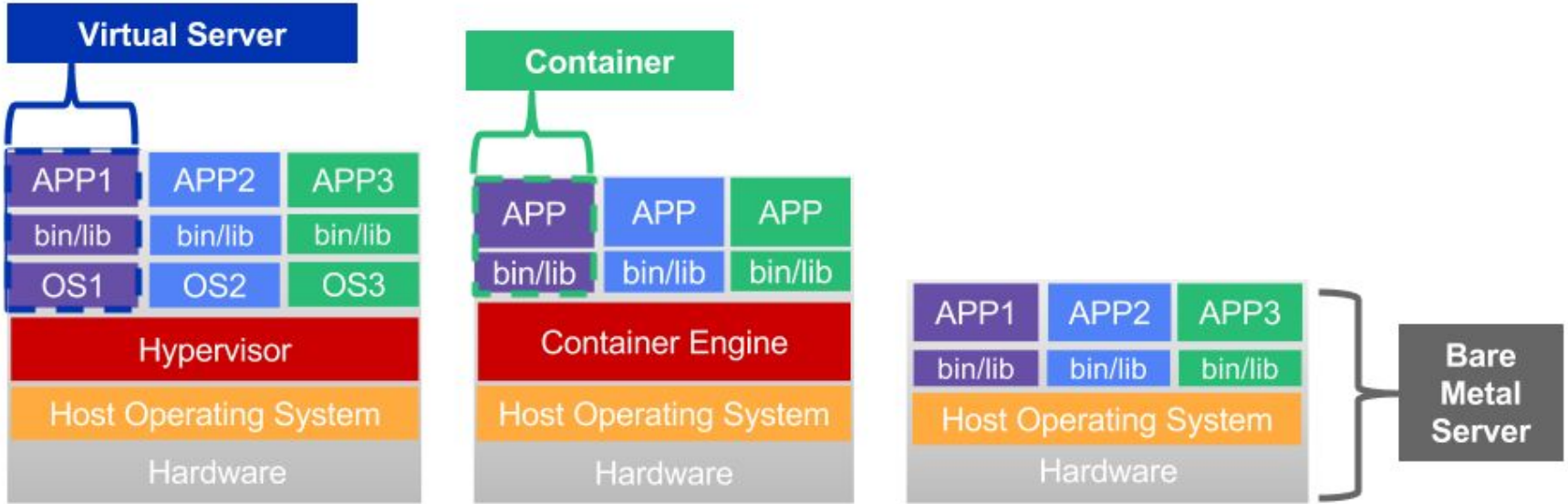
In [2]:
```
%matplotlib inline
from matplotlib import pyplot as plt
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
ax1.plot(x); ax1.set_title('Raw audio signal')
ax2.specgram(x); ax2.set_title('Spectrogram');
```



Versions, versions, versions…….

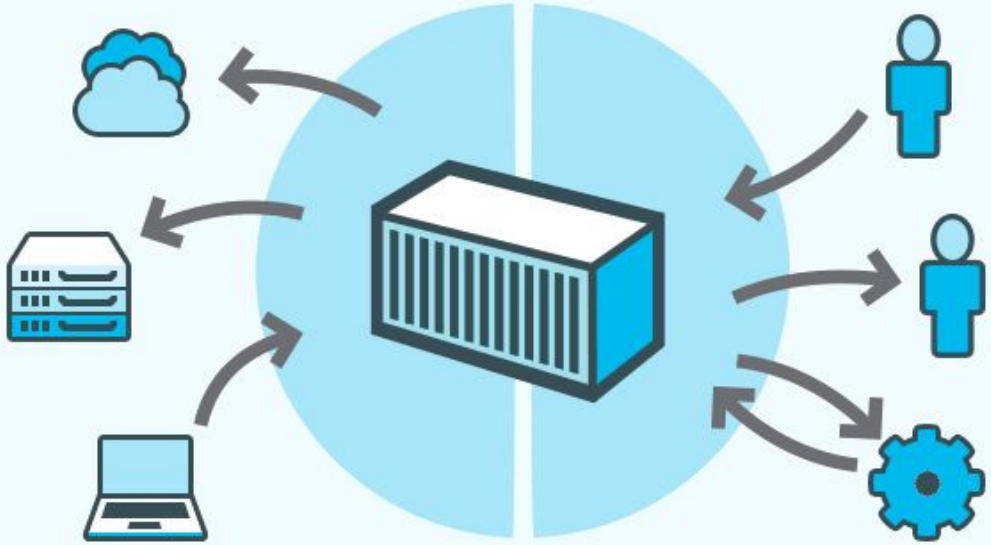
If only there was a way to package all those versions in a complete digitally sharable “thing”.....

Compute Options: VM, Container, Bare Metal



What Is Docker?

An open platform for distributed applications



Docker Engine

A portable, lightweight application runtime and packaging tool.

Learn More

Docker Hub

A cloud service for sharing applications and automating workflows.

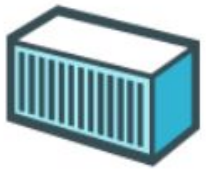
Docker Science Lab

- Build ***immutable*** portable programmatically defined labs
- Run in a consistent environment, with a low barrier to entry
- Share and version files and labs with peers
- Use git to manage files for the lab definition and experiments

Docker Science Lab

Build

Develop an app using Docker containers with any language and any toolchain.



Ship

Ship the “Dockerized” app and dependencies anywhere - to QA, teammates, or the cloud - without breaking anything.

Run

Scale to 1000s of nodes, move between data centers and clouds, update with zero downtime and more.

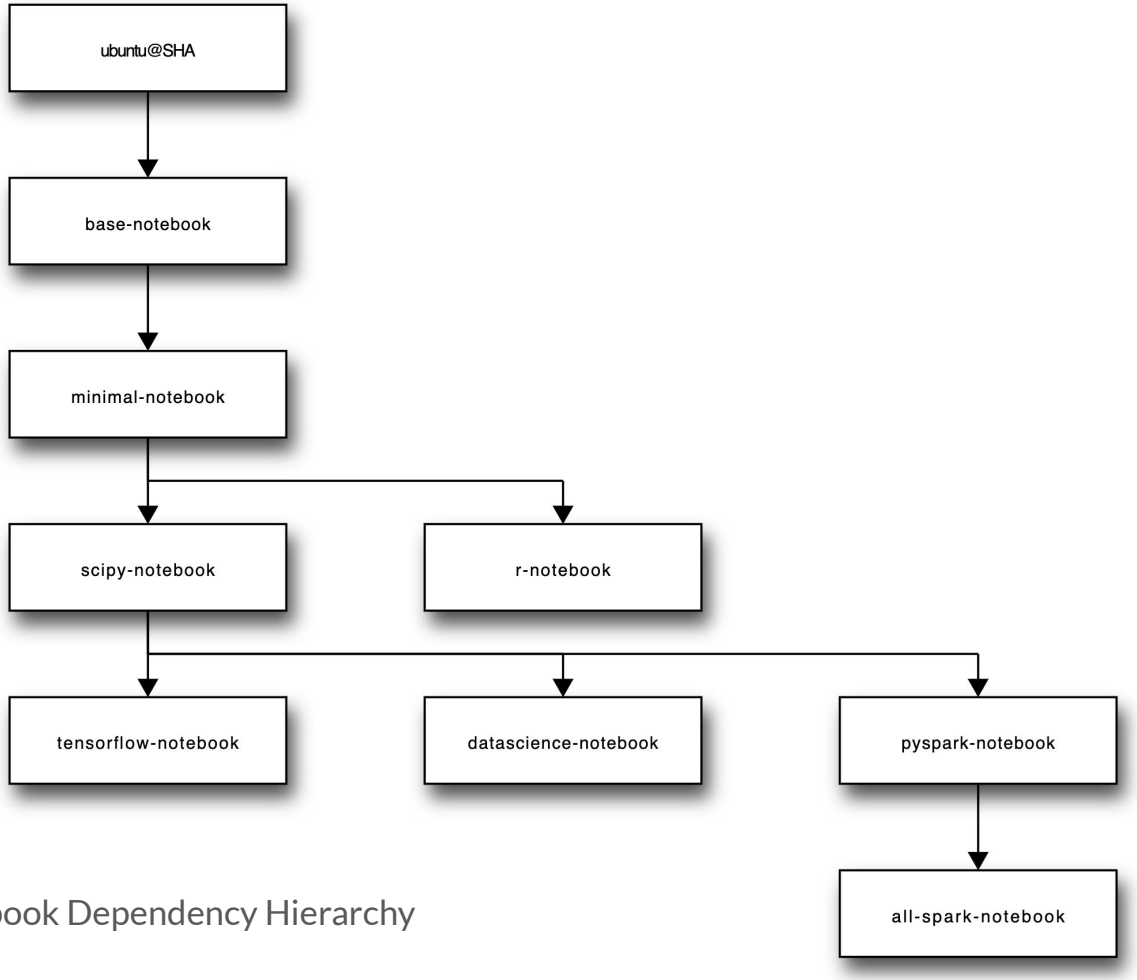
https://jupyter-docker-stacks.readthedocs.io/en/latest/

Jupyter Lab BOM

- Git Repo & Cloud Hosting (GitHub, GitLab, Bitbucket, etc)
- Docker Engine (Portability, Reproducibility)
- Docker Compose (Parameterisation,
- Modern Web Browser

Typical Workflow

- Create a git project for your experiment
- Choose a base Jupyter Image
- Install additional libraries in a customer docker FROM the base Image
- Push custom image to an image registry
- Define experiment in Jupyter Notebook
- Commit and push to remove git repo to share all work



Jupyter Notebook Dependency Hierarchy

Technically.....

- A ***docker-compose*** file describing image build paths and container definition
- Optionally, a ***Dockerfile*** to build a new base image
- Available ports for the Jupyter services (no collisions)
- Git.
- Bind mount repo path to store Notebooks

```
version: '3.4'

services:
  notebook:
    image: mattjtodd/mad-science-1:0.1.0
    Build: ./
    environment:
      - JUPYTER_ENABLE_LAB=yes
    ports:
      - 8888:8888
    command: start-notebook.sh --NotebookApp.token=''
    volumes:
      - $PWD/work:/home/jovyan/work
```

Simple Docker Compose File

```
# Note user of Digest for tag!
FROM jupyter/minimal-notebook:d4cbf2f80a2a

# Reduce version variation
RUN pip install numpy==1.16.4

# Dependency Hell! (transitive ranges)
RUN pip install matplotlib==3.1.0
```

Simple extension Dockerfile

```
$ docker-compose build

$ docker-compose push

$ docker-compose up
```

Browser @ http://127.0.0.1:8888

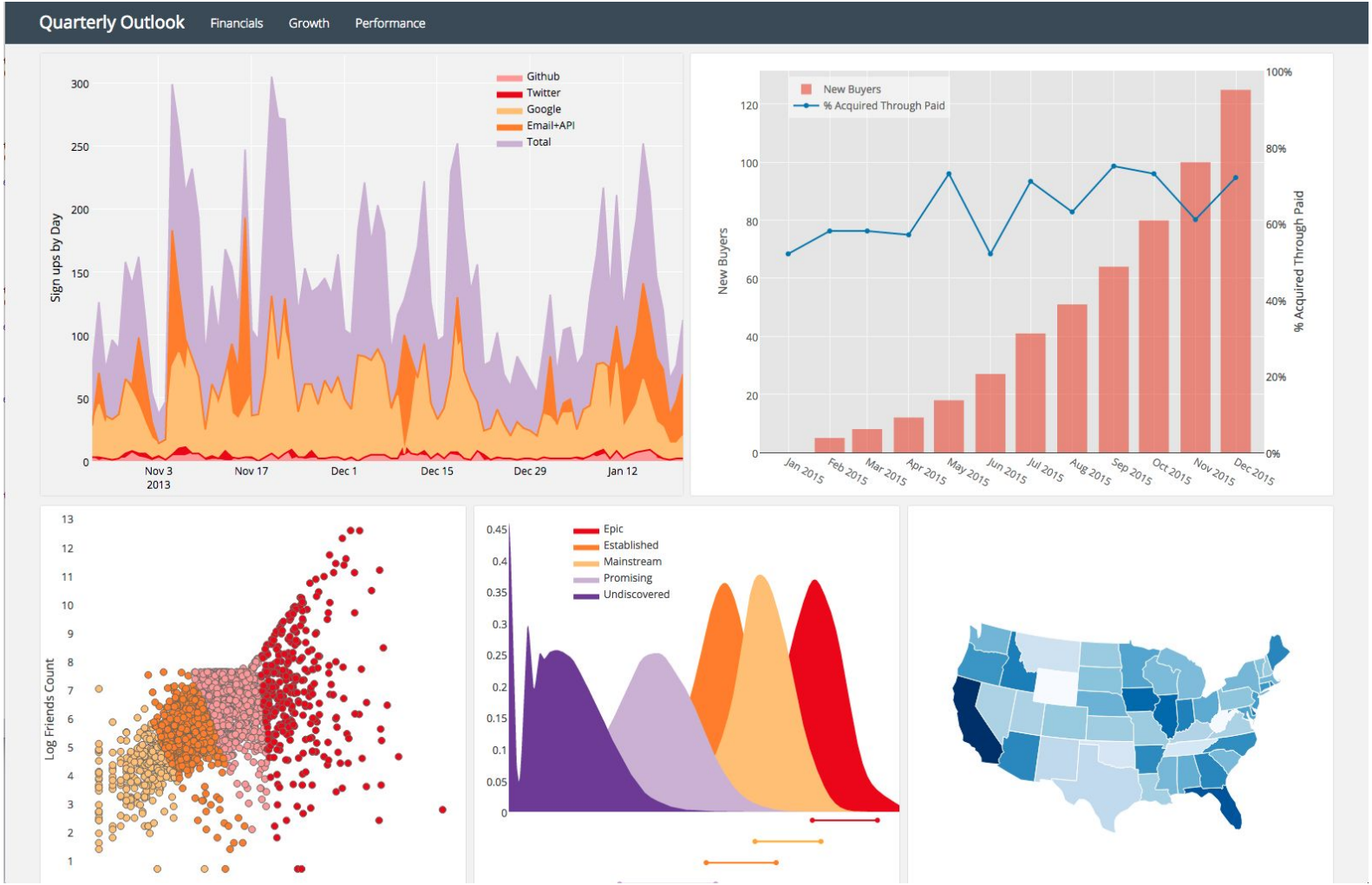
Build, Ship, Run!

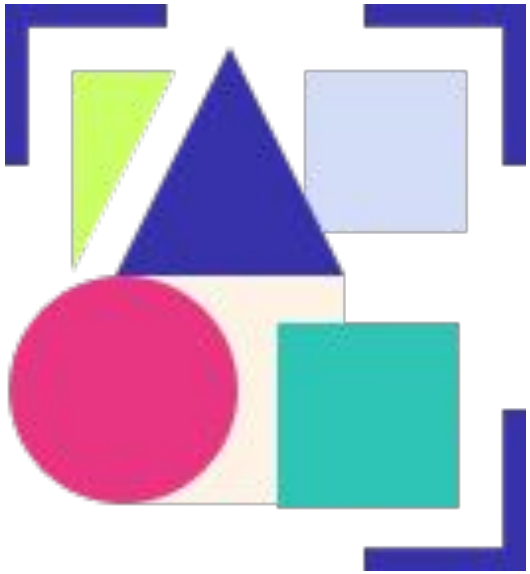
Science Artifacts

- Images are your digital lab
- Files which are bind-mounted into containers built from the images are your experiments
- VCS and Image Registries are Sharing tools

Some Examples.....

https://github.com/mattjtodd/docker-birmingham-july-2019.git


Quarterly Outlook
Financials
Growth
Performance
Github
Twitter
Google
Email+API
Total
Sign ups by Day
300
250
200
150
100
50
0
Nov 3
2013
Nov 17
Dec 1
Dec 15
Dec 29
Jan 12
New Buyers
% Acquired Through Paid
New Buyers
% Acquired Through Paid
120
100
80
60
40
20
0
100%
80%
60%
40%
20%
0%
Jan 2015
Feb 2015
Mar 2015
Apr 2015
May 2015
Jun 2015
Jul 2015
Aug 2015
Sep 2015
Oct 2015
Nov 2015
Dec 2015
Log Friends Count
13
12
11
10
9
8
7
6
5
4
3
2
1
Epic
Established
Mainstream
Promising
Undiscovered
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0



https://cnab.io

https://duffle.sh/

Binder

https://gke.mybinder.org/

Next Time.....

Expand your lab!

- Running other services to connect Notebooks to other services
- More detailed / advanced Jupyter examples
- Machine learning examples
- Spark / Flink examples

Questions?