Docker For Scientific Applications

Joe Stubbs
Advanced Computing Interfaces - TACC
jstubbs@tacc.utexas.edu

Why Docker?

Building scientific libraries is painful...

```
>>> import numpy
Traceback (most recent call last):
Import multiarray ImportError: /usr/lib/python2.7/dist-
packages/numpy/core/multiarray.so: undefined
```

Dependency management is complicated and error prone...

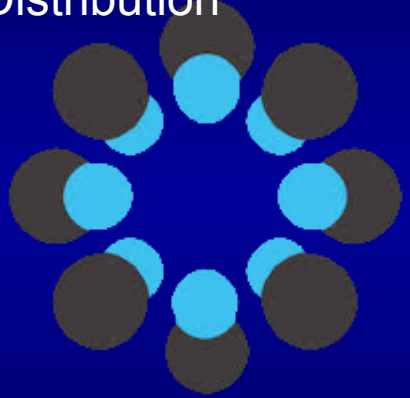
```
$virtualenv foo
$ pip install -r requirements.txt
```

Distributing the app to a new host means repeating the effort all over

Why Docker?

Easy Installation

Simplified Distribution



Isolated Environment

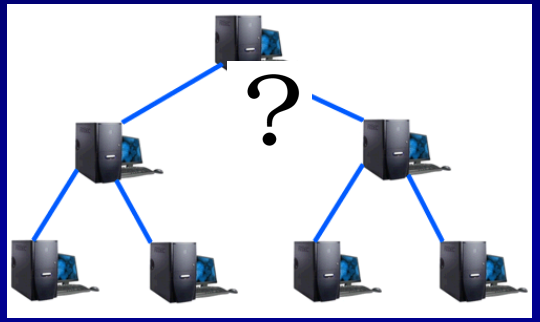
Reproducible Scientific Computations

How Easy Is It?

```
# Install Docker
$ wget -qO- https://get.docker.com/ | sh
# Install and Run an application
$ docker run taccsciapps/bwa --help
Pulling repository taccsciapps/bwa...
Usage: intro [options] ...
```

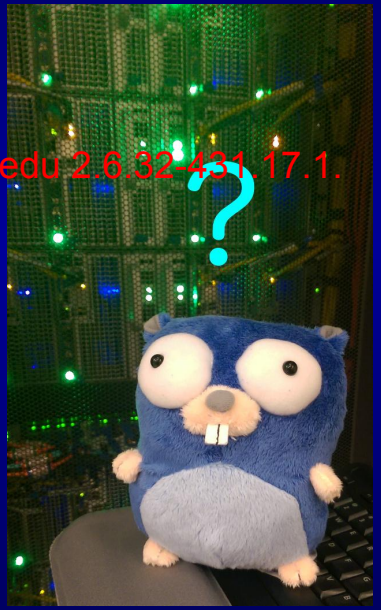
Docker Still Young

Multi-host deployments challenging



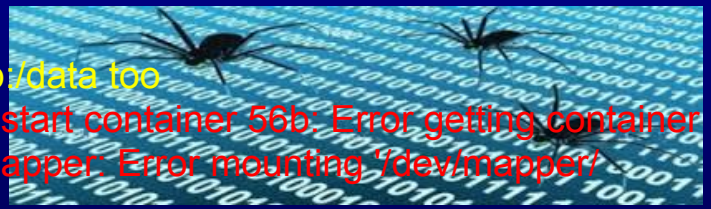
No support for older kernels

$ uname -a
stampede.tacc.utexas.edu 2.6.32-431.17.1.el6.x86_64



bugs...

$ docker run -v /tmp:/data foo
Error: start: Cannot start container 56b: Error getting container 56b from driver devicemapper: Error mounting '/dev/mapper/



TACC

Docker Rapidly Evolving

Synapse

CoreOS

Kubernetes

Flocker/Cluster HQ

Brooklyn

Docker Machine

Docker Compose

Docker Swarm

Powerstrip

SocketPlane

Shipyard

Mesos

Registrator

Consul

What is Agave?

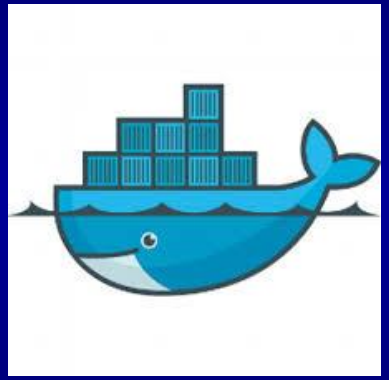
Agave is a *Science-as-a-Service* web API platform

- **Run scientific codes**
  your own or community provided codes
- **...on HPC, HTC, or cloud resources**
  your own, shared, or commercial systems
- **...and manage your data**
  reliable, multi-protocol, async data movement
- **...from the web**
  webhooks, rest, json, cors, oauth2
- **...and remember how you did it**
  deep provenance, history, and reproducibility built in

Docker @TACC

- Elastic Provisioning - Compute and Storage
- Cloud Runner for Execution
- endofday - Workflow engine
- ADAMA - data services
- Event driven compute containers… coming soon

Elastic Storage and Compute



Create Docker hosts in public clouds and register them in Agave

Cloud Runner

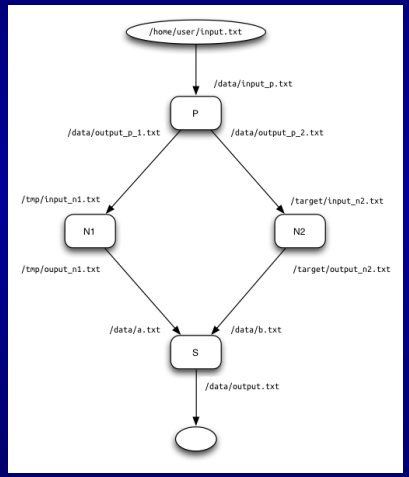
Launch Jobs In The Cloud With A Single Command

1. Specify a work directory with executable and data
2. (Optional) Include a Dockerfile to customize your image

TACC

Workflow Engine

Represent Complex Workflows using YAML



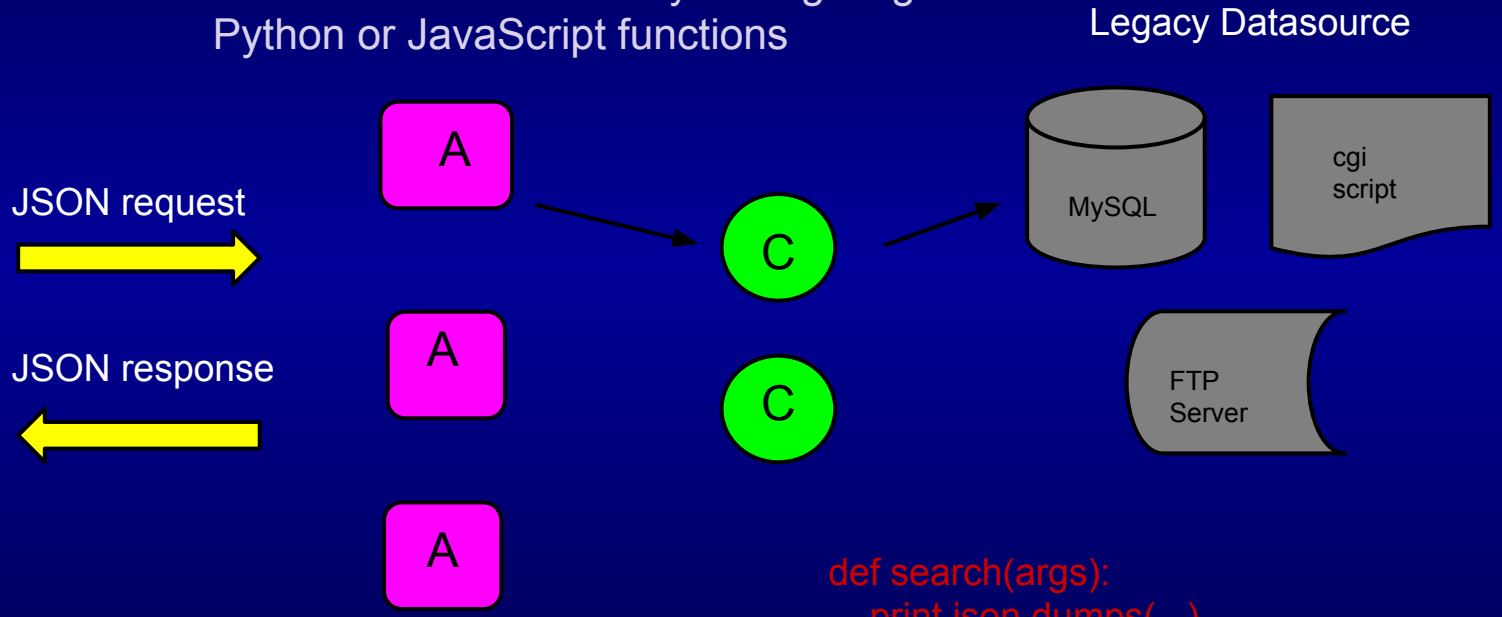

```
inputs:
   - input 1<- /home/jstubbs/workflows/examples/input.txt

outputs:
   - S.output

processes:
   P:
      image: user/image_P
      inputs:
         - inputs.input1 -> /data/input.txt
      outputs:
         - /data/output.txt -> output_P
      command: python p.py /data/input.txt
   N1:
      image: user/image_Q
      inputs:  P.output_P -> /tmp/input.txt
   . . . .
```

Execute on localhost or using Cloud Runner (experimental)

TACC

ADAMA

Create Dataservices that scale by writing single Python or JavaScript functions




```
def search(args):
    print json.dumps(...)
```

Compute Containers

(In early development…)

Run arbitrary containers in response to Agave events:

- file uploads
- job completions
- metadata updates

Trigger execution with messages (think actor model)

Event context/message injected into the container environment

Thanks!

Questions?