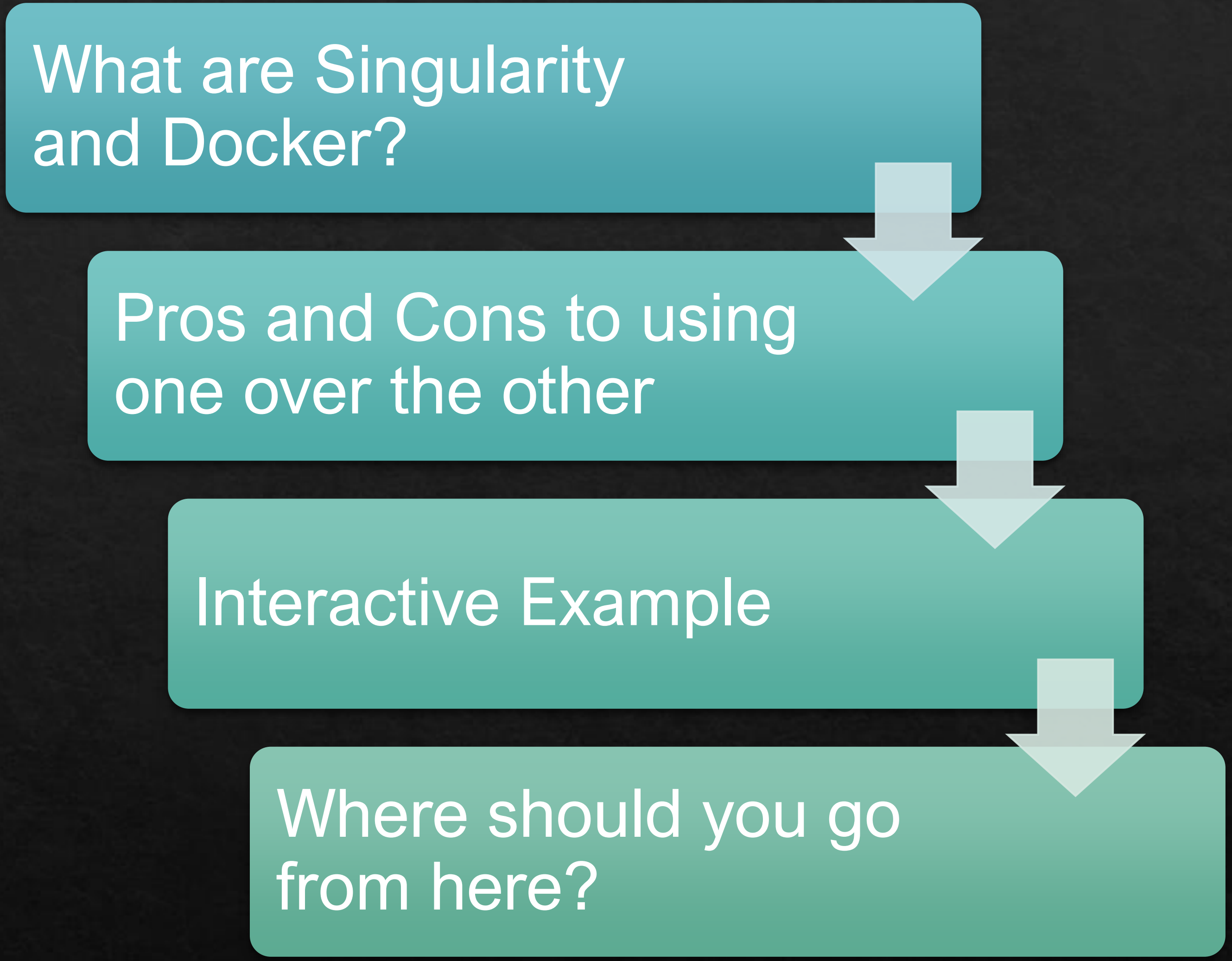
Singularity and Docker

An introduction to containerizing a researcher's workflow

Table of Contents



What are Singularity and Docker?

Pros and Cons to using one over the other

Interactive Example

Where should you go from here?

What is Singularity?

The gold-standard for containerization on HPC clusters

Preferred choice for mission-critical workflows

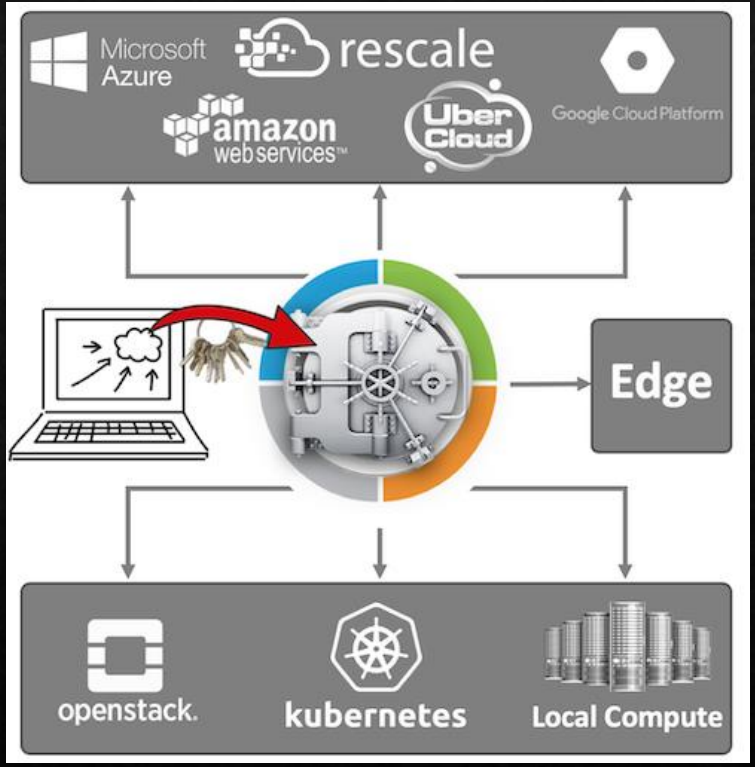
Used in aerospace, oil and gas, and biotechnology industries

Security first mindset

No one is hacking you!

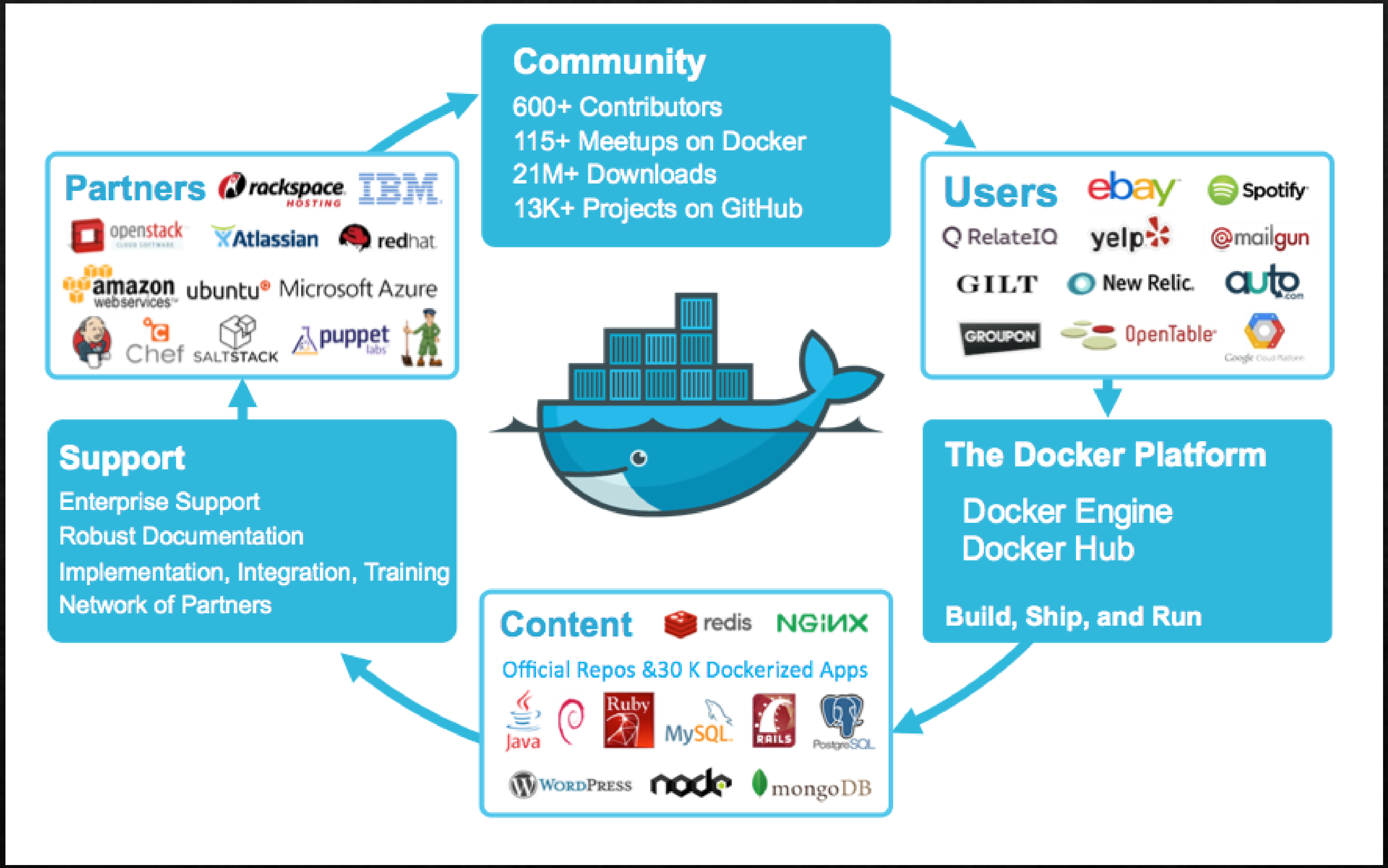
Extremely portable

Can run in any environment that has singularity RE installed



What is Docker?

- Another popular, ubiquitous containerization software
- Heavily used in the tech industry
  - Kubernetes anyone?
- Extremely adaptable to any kind of workflow
- Cross-Platform
  - Works on Linux, MacOS, and Windows



Community

600+ Contributors
115+ Meetups on Docker
21M+ Downloads
13K+ Projects on GitHub

Users

The Docker Platform

Docker Engine
Docker Hub

Build, Ship, and Run

Content

Official Repos &30 K Dockerized Apps

Support

Enterprise Support
Robust Documentation
Implementation, Integration, Training
Network of Partners

Partners

The Pros and Cons of Singularity

The Pros ☺

- Uber secure
  - Singularity containers are read-only
- Space efficient
  - Containers are compressed to conserve available storage space
- Installed on Roar
- Can easily manage system-level dependencies
- Interoperable with Docker

The Cons ☹

- Only supports static environments
- Only available for use on Linux
  - Need to use a work-around to build containers on Windows and Mac
- User community isn’t as large as compared to Docker
  - Finding support can be a little difficult

The Pros and Cons of Docker

The Pros ☺

- Ubiquitous, virtually used everywhere
- Cached builds
  - Start from where you left off if build fails
- Support dynamic environments
- Can modify containers after being built from images
- Multithread container downloads and uploads

The Cons ☹

- Not installed on the cluster
- Security is managed by user, not built-in
  - Container security has become an entire industry because of Docker
- Not resource conscious
  - Bloated containers

Interactive Example Time

Set up your build environment!

- Open the terminal on your computer
- Create directory titled **VMs**
- Change into **VMs** directory and create **SINGULARITY** directory
- In **SINGULARITY** directory, enter the following commands:
  - `vagrant init sylabs/singularity-3.6-ubuntu-bionic64`
  - `vagrant up`

Time to build your container!

- Use the following commands:

  -
    ```
    vagrant ssh
    ```

  -
    ```
    git clone https://github.com/NucciTheBoss/iask_onboarding_spring_2021.git
    ```

  -
    ```
    cd iask_onboarding_spring_2021/day_2
    ```

  -
    ```
    sudo singularity build image.sif R-4.0.3-rstudio.def
    ```
    - sudo password is “**vagrant**”
- Now we wait!

Demonstration!

Where should you go from here?

- Feel free to play around with my repository:
  - https://github.com/NucciTheBoss/iask_onboarding_spring_2021
- Experience is the greatest teacher!
  - Create your own containers
  - Play around with containers we already created
- Ask for help!
  - Will, Justin Petucci, and I use Singularity all the time
  - Add me as a collaborator on a git repository!
    - I can provide input if you need help!

"That's all Folks!"