



{Do}

Depend on Docker

Imagine a world where the only dependency
you need is Docker!



Alex Iankoulski

Principal Software Architect

Baker Hughes, a GE Company

<https://www.linkedin.com/in/alex-iankoulski/>



Arun Subramaniyan

VP Data Science & Analytics

Baker Hughes, a GE Company

<https://www.linkedin.com/in/arunsubramaniyan/>

Business Context



Some Problems we care about...

- Every 2 seconds a GE Jet engine takes off in the world
- ~ 30% of the world's power is produced by a GE turbine
- We are the first FullStream O&G company – Exploration ...
Production ... Transportation ... Operations

We need to model mission critical equipment and systems accurately

Traditional Enterprise Data vs Industrial Data

Upstream

Drilling Data
0.3 GB/well/day

Wireline Data
5 GB/well/day

Fiber Optic Data
0.1 GB/well/day

ESP Monitoring
0.4 GB/well/day

Seismic Data
500 GB/survey

Midstream

Process Data
4 GB/plant/day

Atmospheric Data
0.1 GB/plant/day

Pipeline Inspect
1.5 TB / 600 km

Downstream

Process Data
6 GB/plant/day

Operational Data
8 GB/plant/year

Vibration Data
7.5 GB/yr/customer

Inspections

Ultrasound: Tubes
1.2 TB/ 8 hrs

Corrosion Mon
1 GB / site / yr

Thousands of assets

ERP Systems

Transactional Data

Logistics Reports

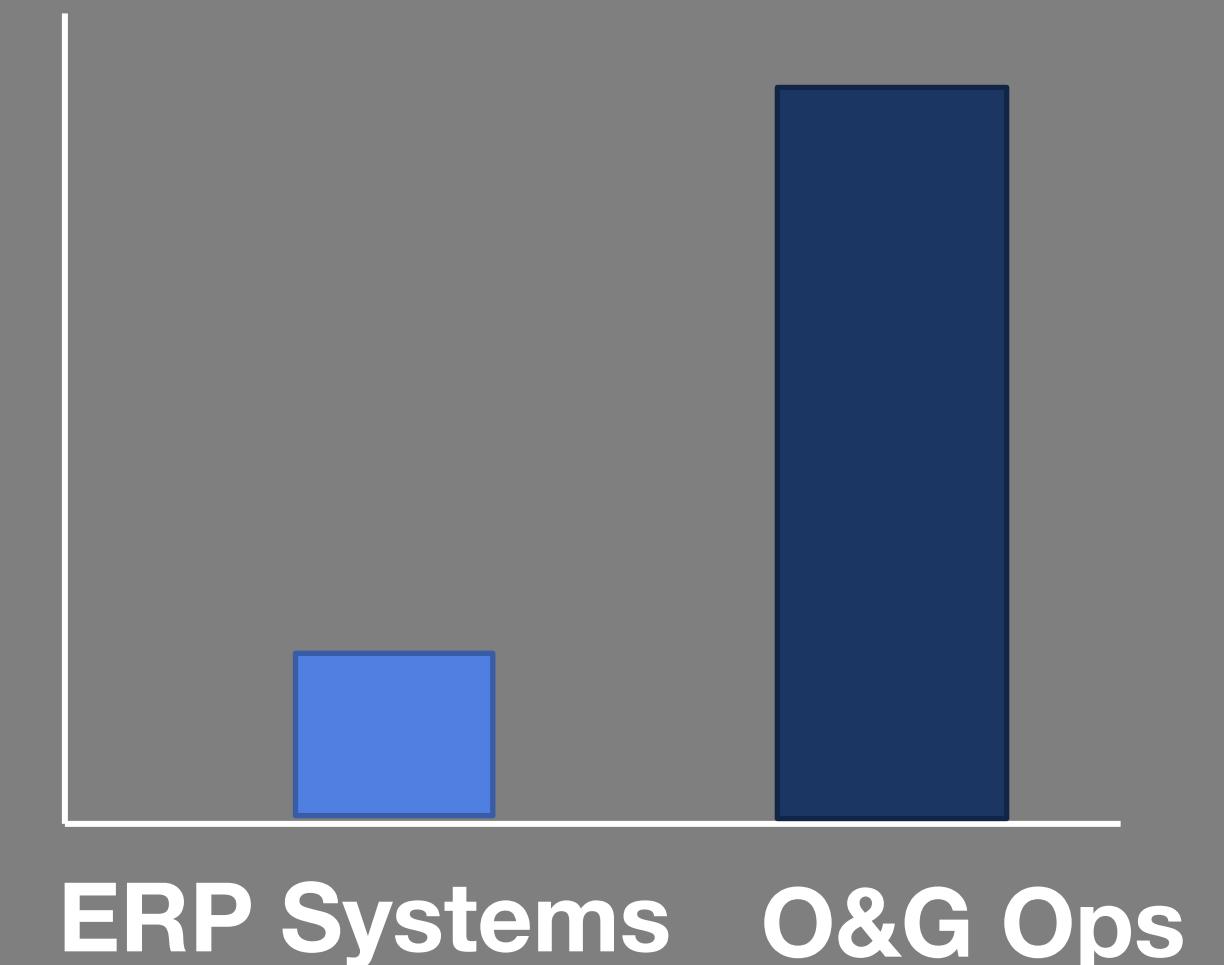
Maintenance Logs

FSE Reports

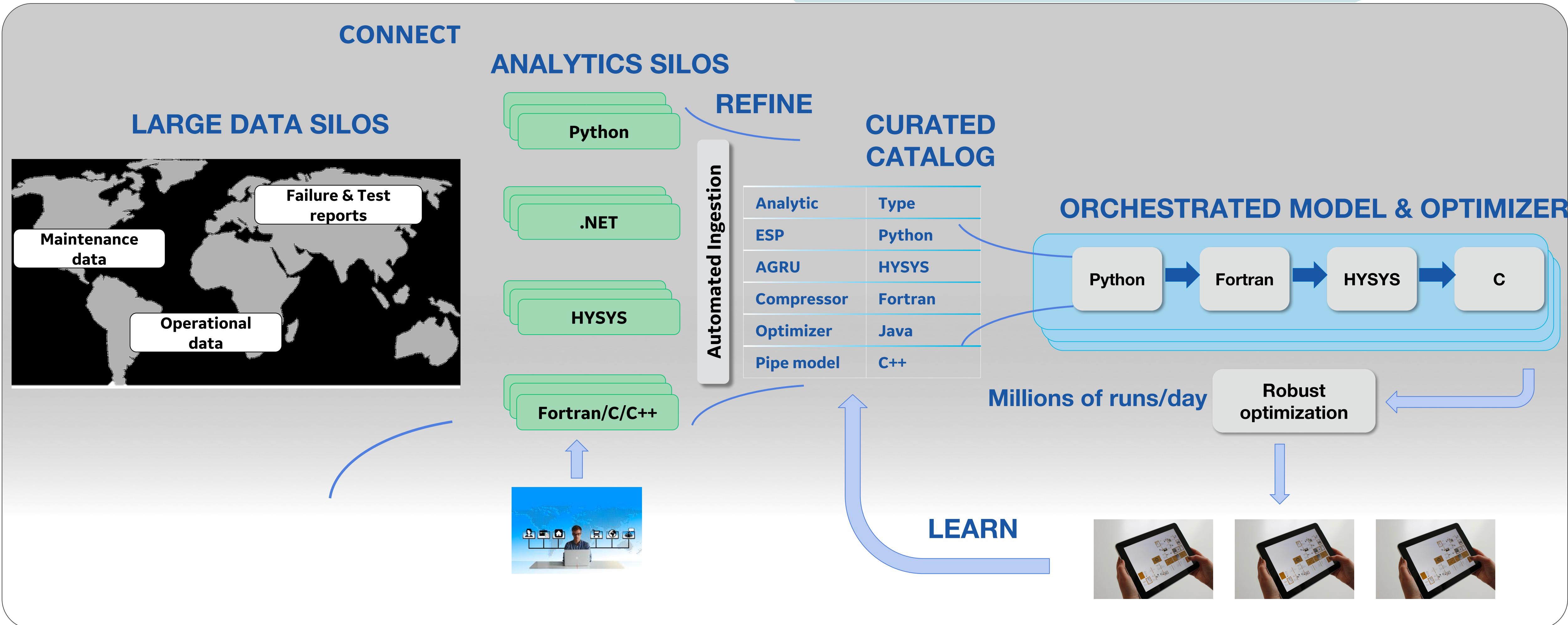
Inventory Reports

Data from industrial assets dwarf data in ERP systems

- ~10-100x more volume
- ~100-1000x more velocity

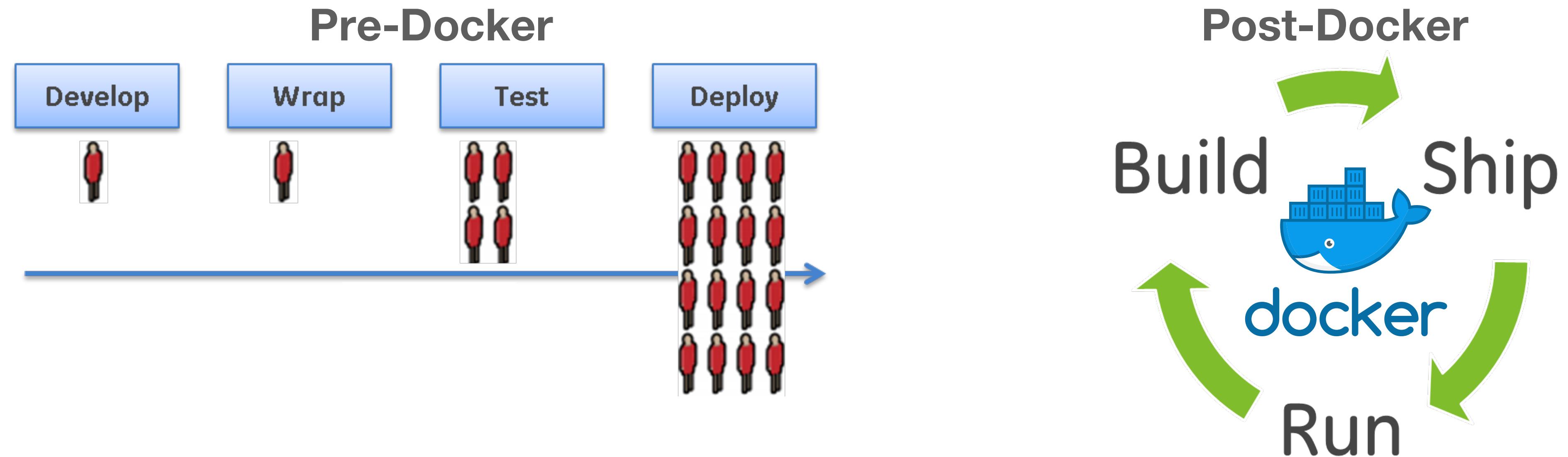


Industrial outcomes require massive scale – on demand



Silos of data/analytics + Federation + Scale + Constant Learning = Need for Containers

Democratize Analytics



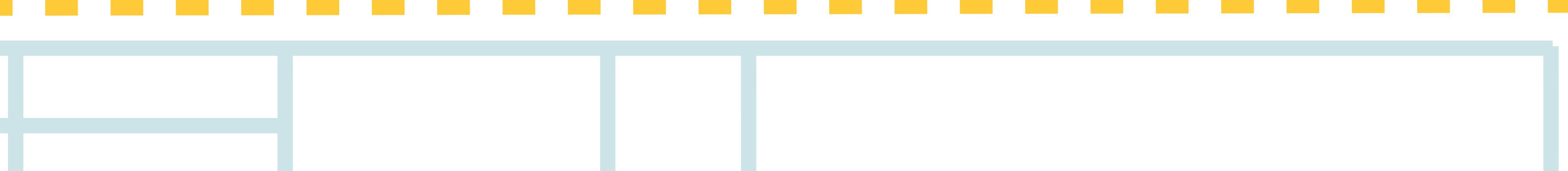
- Production deployment: 6-12 months
- New developer setup: 1 week
- Quarterly releases

- Production: ~ 1 hour
- New developer setup: 1 hour
- **Continuous releases**

10-100X reduction in time-to-deployment



A Transformational Whale of a Tale



Billions of creative people

Incredible technology



DockerCon!

Run everything in Docker!

- Billions of creative people
- Incredible technology
- Tools of mass innovation
- Explosive growth
- Move fast



Monolith

vs

Micro-service

Architecture

Image from Big Hero 6, courtesy of Disney Animation Studios



Fork in the Road

- Choose
 - Micro-services
 - Scale
 - Docker containers
 - No other dependencies



<http://www.trippyfood.com/2010/01/01/right-is-right-and-left-is-wrong/>

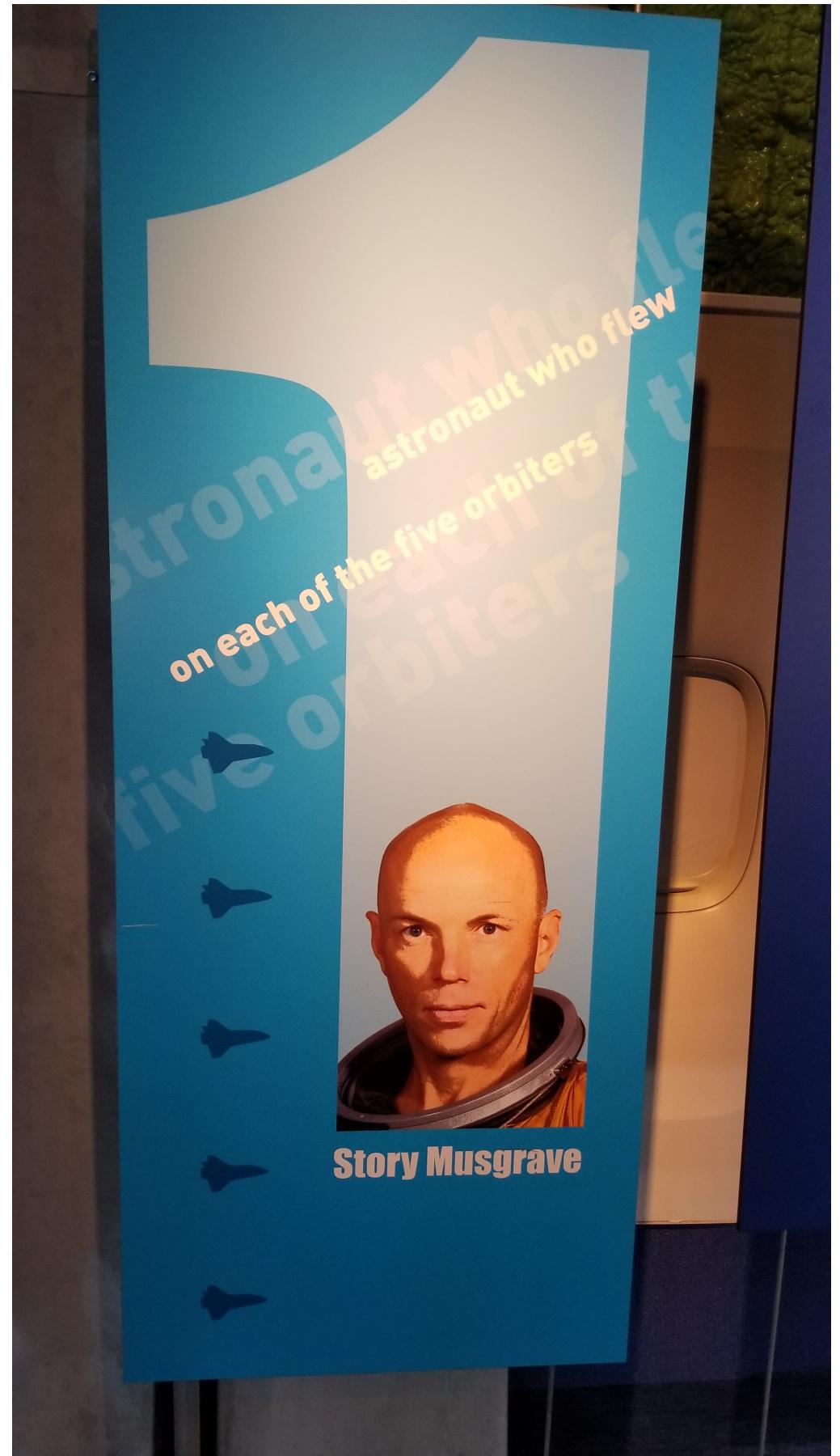


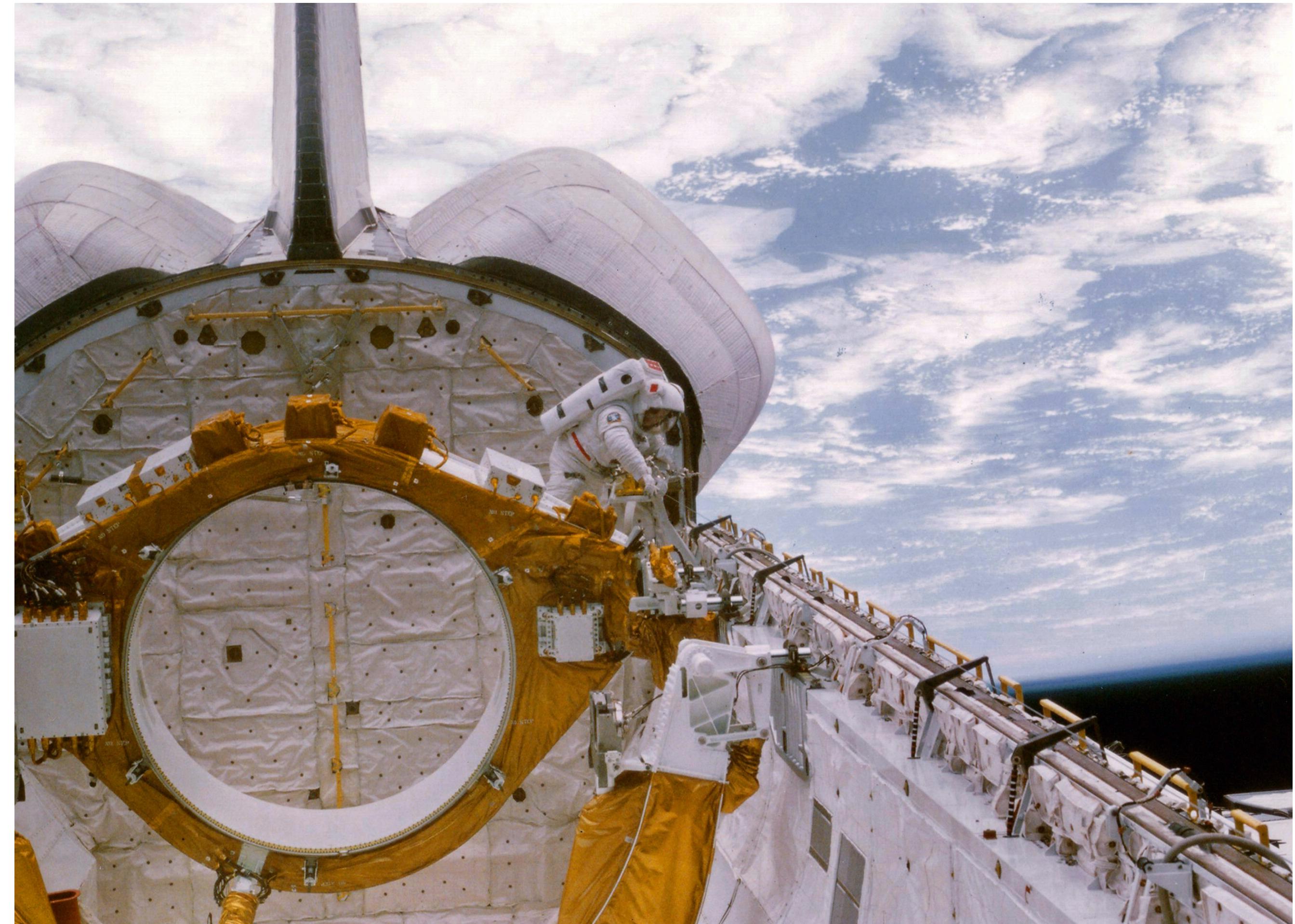
If you can imagine it,
you can build it!

- # Just imagine ...
- # With Docker the answer is “Yes”
- # All you need to know is
know what you need
- # There is a container for that
- # I can’t believe this just worked!
- # “There is no spoon”

Image courtesy of Disney Animation Studios

We see containers everywhere





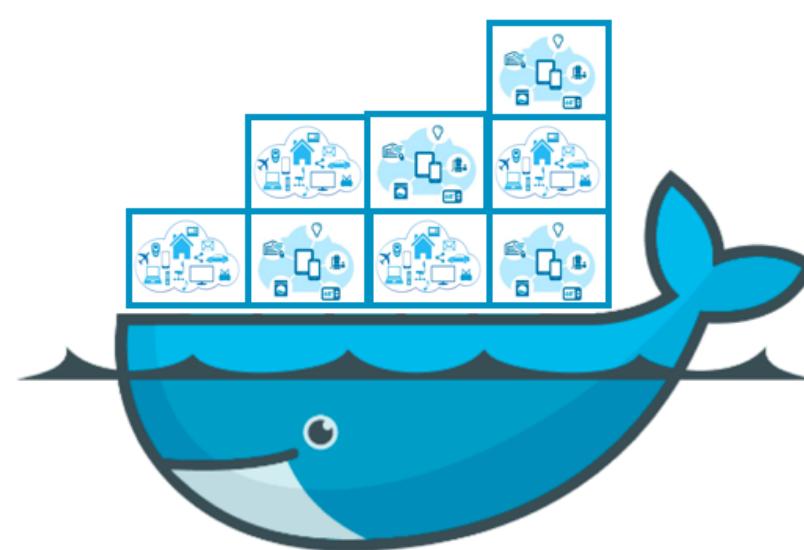
Dr. Story Musgrave

docker
con¹⁸
SF

Images courtesy of Dr. Story Musgrave

Using Containers

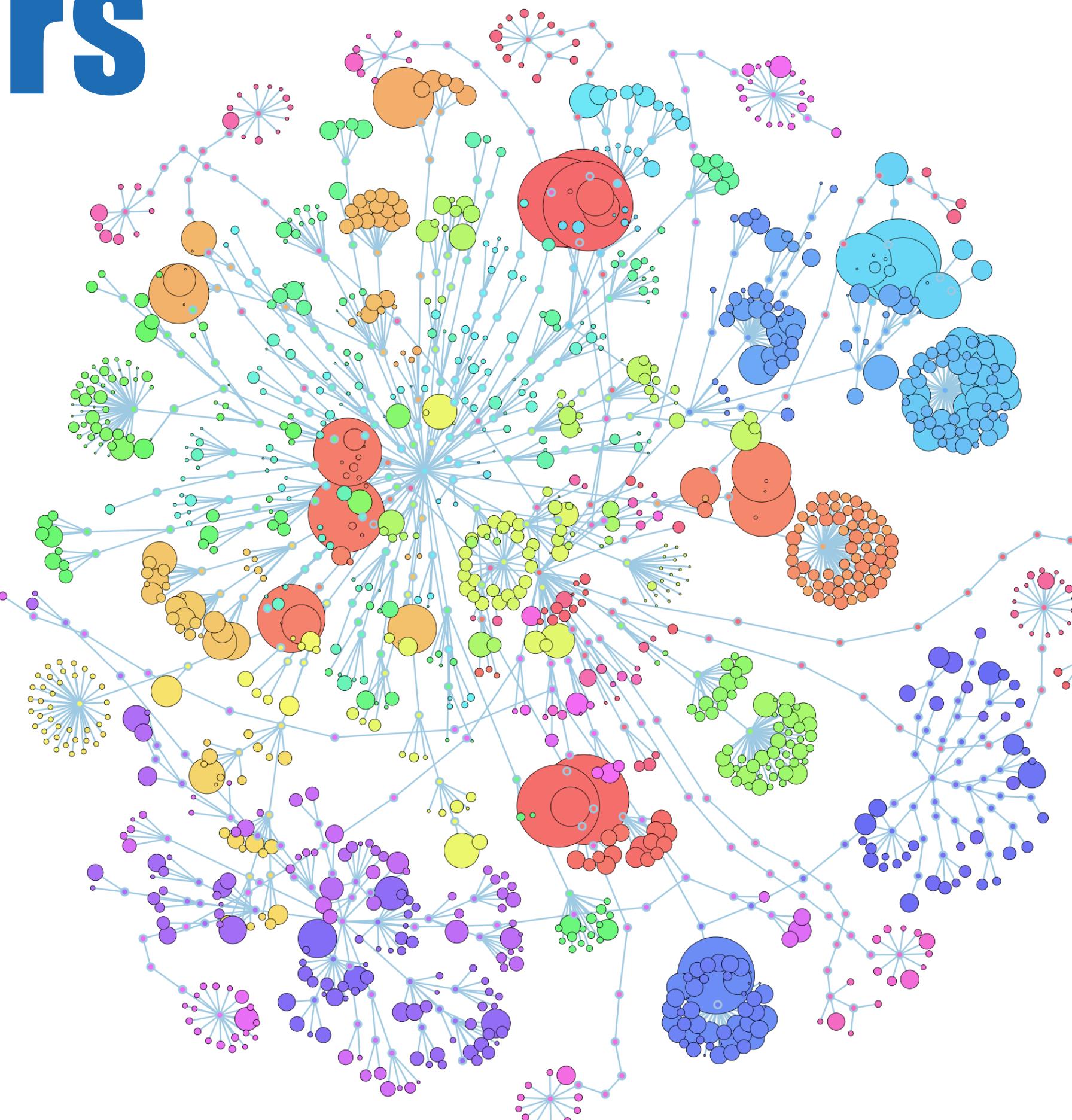
- Dependency management



Containerized application

Containerize

{dependencies encapsulated,
complexity reduced}

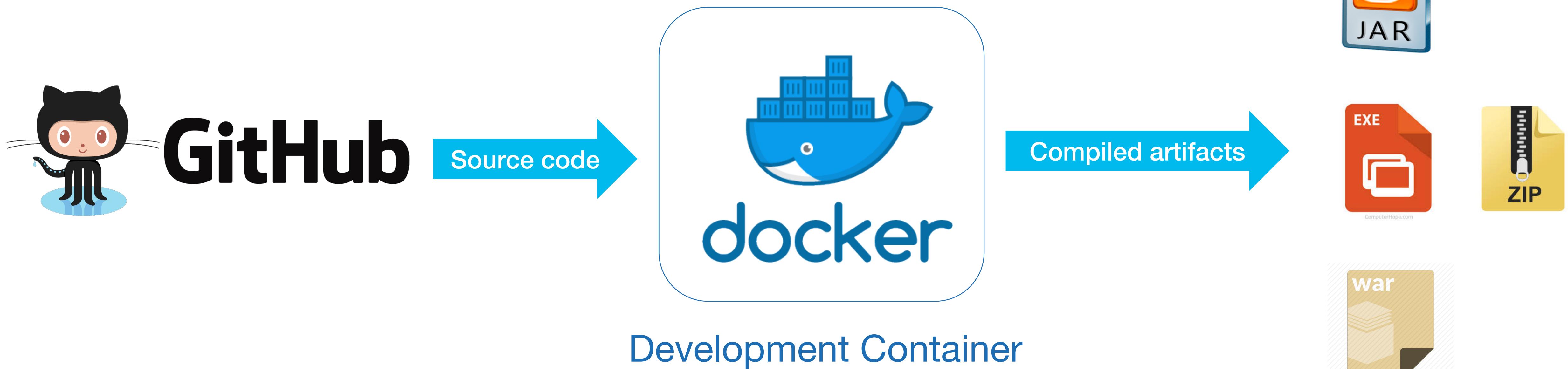


Example: semtk-opensource

Code Flower

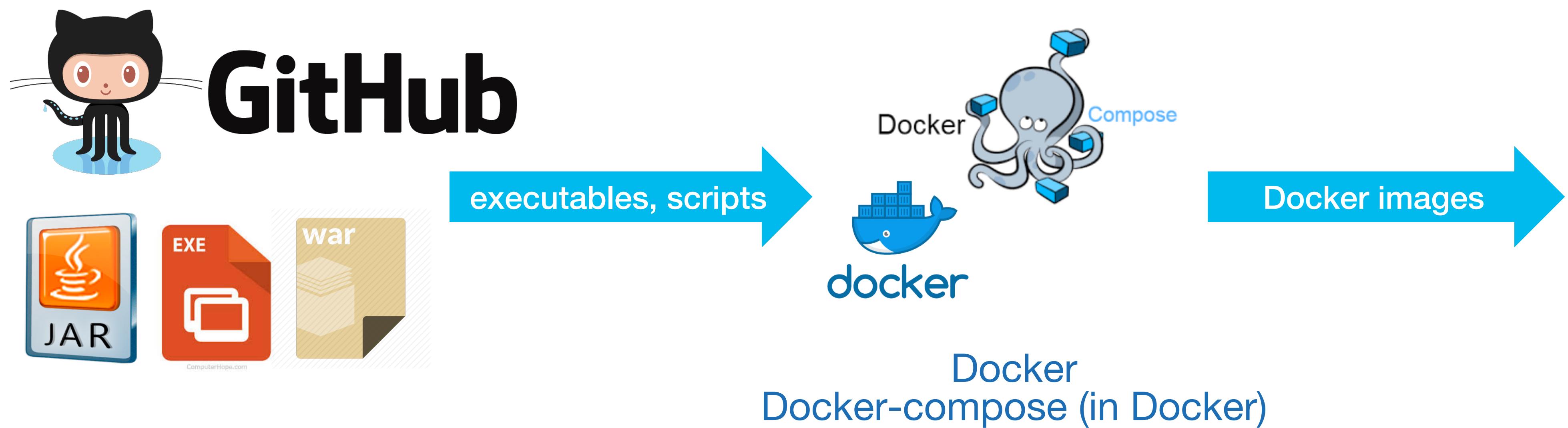
Using Containers

- Compile source code



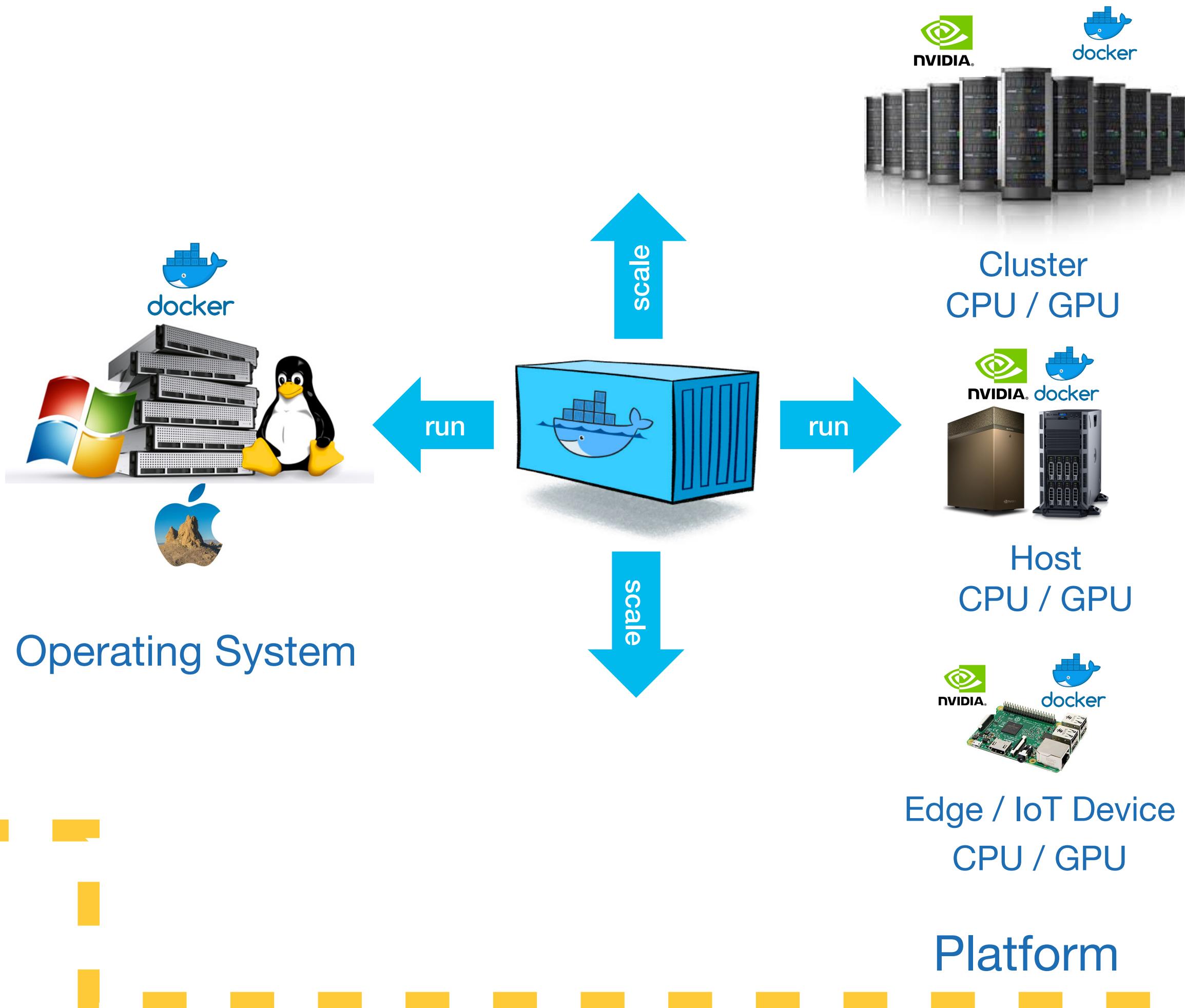
Using Containers

- Build container images



Scale up & down at will

- Ship and Run anywhere



Using Containers Everywhere

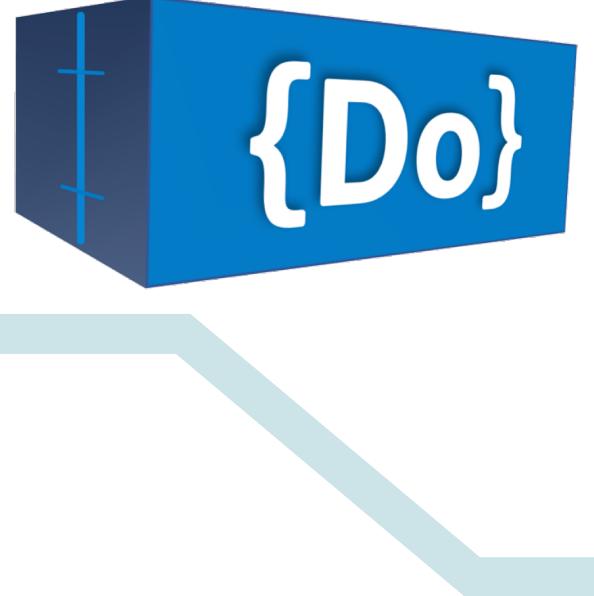
- **CI / CD Systems** – Use Docker in Docker to build and ship container images
- **Software Test execution** – Create environment, execute test, remove container
- **Development** – IDE in a container, Cloud IDE, super easy developer onboarding
- **Diagnostics, Troubleshooting, Support** – Run Docker to collect diagnostic info
- **Patching** - Run Docker command to fix and patch installed software
- **Backups** – Run your backup job in a container
- **Firmware updates** via containers!
- No reason to run anything outside of a container anymore ☺



Depend-on-Docker



Depend-on-Docker is



- **Philosophy** – don't install anything, build a container instead
- **Pattern** – needs only access to source and Docker
- **Project** – available at <https://github.com/bhgedigital/depend-on-docker>
- **Simple** – create a new DoD project with a single command
- **Flexible** – defaults provided, but customizable
- **Universal** – works with Linux and Windows containers

Depend-on-Docker Example Semantics Toolkit (semtk)

<https://github.com/ge-semtk/semtk>

```
[iankouls@semtk semtk]$ ls
build.sh           LICENSE.txt
compile.sh         nodeGroupExecutionService
compose.sh          sparqlDB
compose.yml        nodeGroupService
conf               sparqlExtDispatchService
docs              nodeGroupStoreService
documentationFiles ontologyInfoService
hiveService        sparqlGraphIngestionService
pom.xml            sparqlGraphLibrary
README.md          sparqlGraphResultsService
                    sparqlGraphStatusService
                    sparqlGraphWeb
[iankouls@semtk semtk]$
```

```
[INFO] Installing /wd/nodeGroupService/target/nodeGroupService-2.0.1-SNAPSHOT.jar
to /root/.m2/repository/com/ge/research/semtk/nodeGroupService/2.0.1-SNAPSHOT/node
GroupService-2.0.1-SNAPSHOT.jar
[INFO] Installing /wd/nodeGroupService/pom.xml to /root/.m2/repository/com/ge/rese
arch/semtk/nodeGroupService/2.0.1-SNAPSHOT/nodeGroupService-2.0.1-SNAPSHOT.pom
[INFO] -----
[INFO] Reactor Summary:
[INFO]
[INFO] semtk-oss 2.0.1-SNAPSHOT ..... SUCCESS [ 0.313 s]
[INFO] sparqlGraphLibrary ..... SUCCESS [ 4.435 s]
[INFO] sparqlGraphIngestionService ..... SUCCESS [ 0.492 s]
[INFO] oracleService ..... SUCCESS [ 0.194 s]
[INFO] hiveService ..... SUCCESS [ 0.064 s]
[INFO] sparqlQueryService ..... SUCCESS [ 0.068 s]
[INFO] sparqlGraphStatusService ..... SUCCESS [ 0.057 s]
[INFO] sparqlGraphResultsService ..... SUCCESS [ 0.072 s]
[INFO] standaloneExecutables ..... SUCCESS [ 15.711 s]
[INFO] nodeGroupStoreService ..... SUCCESS [ 0.155 s]
[INFO] ontologyInfoService ..... SUCCESS [ 0.061 s]
[INFO] sparqlExtDispatchService ..... SUCCESS [ 0.054 s]
[INFO] nodeGroupExecutionService ..... SUCCESS [ 0.052 s]
[INFO] NodeGroup Service 2.0.1-SNAPSHOT ..... SUCCESS [ 0.049 s]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 22.335 s
[INFO] Finished at: 2018-06-14T01:53:37Z
[INFO] -----
[iankouls@semtk semtk]$
```



```
[iankouls@semtk semtk]$ ./compose.sh up -d
Creating network "host_semtknet" with driver "bridge"
Creating host_semtk-sparqlgraph-results_1 ... done
Creating host_semtk-nodegroup-store_1 ... done
Creating host_semtk-ingestion_1 ... done
Creating host_semtk-hive_1 ... done
Creating host_semtk-sparqldb_1 ... done
Creating host_semtk-nodegroup_1 ... done
Creating host_semtk-dispatch_1 ... done
Creating host_semtk-nodegroup-execution_1 ... done
Creating host_semtk-sparql-query_1 ... done
Creating host_semtk-sparqlgraph-status_1 ... done
Creating host_semtk-sparqlgraph-web_1 ... done
Creating host_semtk-ontology-info_1 ... done
[iankouls@semtk semtk]$ ./compose.sh ps
      Name            Command     State        Ports
-----+-----+-----+-----+
host_semtk-dispatch_1   /bin/sh -c cd /service; ls ... Up      0.0.0.0:12053->8080/tcp
host_semtk-hive_1       /bin/sh -c cd /service; ls ... Up      0.0.0.0:12055->8080/tcp
host_semtk-ingestion_1  /bin/sh -c cd /service; ls ... Up      0.0.0.0:12091->8080/tcp
host_semtk-nodegroup-execution_1 /bin/sh -c cd /service; ls ... Up      0.0.0.0:12058->8080/tcp
host_semtk-nodegroup-store_1 /bin/sh -c cd /service; ls ... Up      0.0.0.0:12056->8080/tcp
host_semtk-nodegroup_1   /bin/sh -c cd /service; ls ... Up      0.0.0.0:12059->8080/tcp
host_semtk-ontology-info_1 /bin/sh -c cd /service; ls ... Up      0.0.0.0:12057->8080/tcp
host_semtk-sparql-query_1 /bin/sh -c cd /service; ls ... Up      0.0.0.0:12050->8080/tcp
host_semtk-sparqldb_1    /bin/sh -c /startup.sh Up      0.0.0.0:1111->1111/tcp,
                           0.0.0.0:2420->8890/tcp
host_semtk-sparqlgraph-results_1 /bin/sh -c cd /service; ls ... Up      0.0.0.0:12052->8080/tcp
host_semtk-sparqlgraph-status_1  /bin/sh -c cd /service; ls ... Up      0.0.0.0:12051->8080/tcp
host_semtk-sparqlgraph-web_1   /bin/sh -c /usr/local/tomc ... Up      0.0.0.0:80->8080/tcp
[iankouls@semtk semtk]$
```

Depend-on-Docker Example Semantics Toolkit (semtk)



<https://semtk.research.ge.com>

Welcome to semtk.research.ge.com.

The Semantics Toolkit (SemTk) is a research project of the Knowledge Discovery Lab at [GE Global Research](#) in Niskayuna NY.

It is an open source project intended to provide easy interactions W3C semantic tech stacks, starting with Virtuoso. It includes drag-and-drop query generation and data ingestion.

Find the tools running here:

- sparqlGraph** - our flagship tool for semantic data scientists
- sparqlForm** - simplified end-user front end for select queries

This demo server is hosted on AWS. The tools are fully functional and available for your use. The Virtuoso triple store is for temporary use only. It is automatically and frequently wiped clean. The documentation explains how you can use this instance of SPARQLgraph with your own local triple store.

<https://github.com/ge-semtk/semtk/wiki>

Home

SemTk team @ Knowledge Discovery Lab edited this page on Apr 16 · 23 revisions

SemTk: Semantics Toolkit

Built on the [W3C Semantic Web](#)

See the live demos at semtk.research.ge.com and out the "Hello World" demo

Overview

The Semantics Toolkit (SemTk) is a research project of the Knowledge Discovery Lab at [GE Global Research](#) in Niskayuna NY.

It is composed of two main parts:

- **SPARQLgraph / SPARQLform** - a graphical web tool for ingesting data and auto-generating SPARQL queries. This includes Javascript libraries that function as a work-alike for the Java ones.
- **SemTk Java library** - the java used to build the back-end services. It has its own API for other uses.

Clone this wiki locally
<https://github.com/ge-semtk/semtk>

Depend-on-Docker Example Tree Project

<https://github.com/iankoulski/tree>



```
1. iankouls@semtk:~/semtk (bash)
> ls
project run.sh
iankouls@bhge-ai:~/Projects/2018/github.com/bhgedigital/demos/tree Alex |
> ./run.sh
docker container run --rm -it -v /Users/iankouls/Projects/2018/github.com/bhgedigital/demos/tree:/Users/iankouls/Projects/2018/github.com/bhgedigital/demos/tree iankoulski/tree /Users/iankouls/Projects/2018/github.com/bhgedigital/demos/tree
/Users/iankouls/Projects/2018/github.com/bhgedigital/demos/tree
└── project
    └── Depend
        └── on
            └── Docker!
└── run.sh

4 directories, 1 file
iankouls@bhge-ai:~/Projects/2018/github.com/bhgedigital/demos/tree
>
```

Depend-on-Docker Example Enigma Project

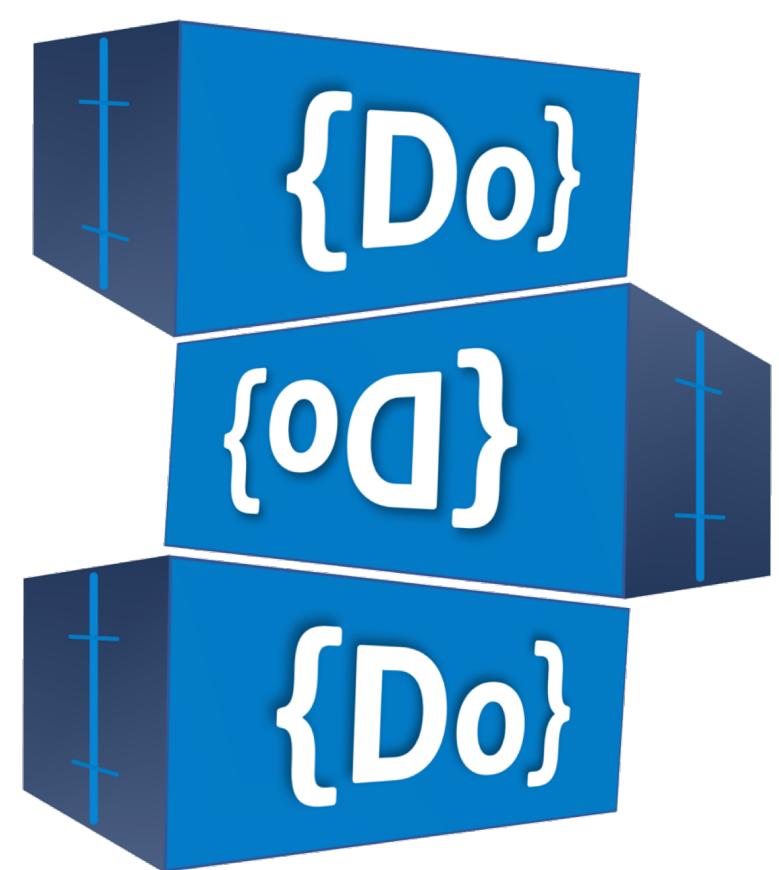
<https://github.com/bhgedigital/enigma>

```
1. iankouls@semtk:~/semtk (bash)
> docker run -it --rm -v ~/keys:/keys -v $(pwd):/wd bhgedigital/enigma:2.0 encrypt "Depend on Docker"
507a8fed072374875c06a252fc0ed3c6d2163a289e3c3168664af666bac30d6c
iankouls@bhge-ai:~/Projects/2018/github.com/bhgedigital/demos/enigma
>
```

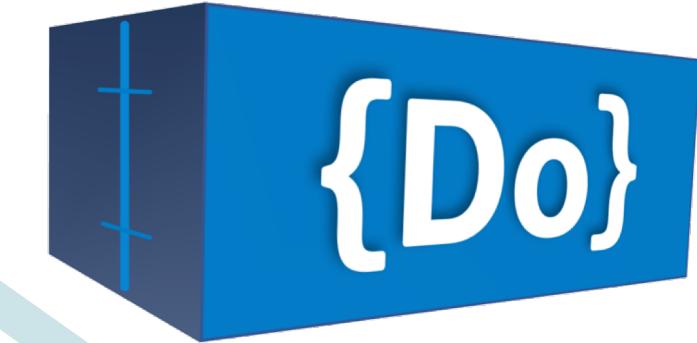
```
1. iankouls@semtk:~/semtk (bash)
> docker run -it --rm -v ~/keys:/keys -v $(pwd):/wd bhgedigital/enigma:2.0 decrypt
507a8fed072374875c06a252fc0ed3c6d2163a289e3c3168664af666bac30d6c
Depend on Docker
iankouls@bhge-ai:~/Projects/2018/github.com/bhgedigital/enigma
>
```

```
1. iankouls@semtk:~/semtk (bash)
> ls
Container-Root README.md      decrypt.sh    keys          push.sh
Dockerfile      build.sh      encrypt.sh    pull.sh
iankouls@bhge-ai:~/Projects/2018/github.com/bhgedigital/enigma
> ./encrypt.sh "Depend on Docker"
507a8fed072374875c06a252fc0ed3c6d2163a289e3c3168664af666bac30d6c
iankouls@bhge-ai:~/Projects/2018/github.com/bhgedigital/enigma
> ./decrypt.sh 507a8fed072374875c06a252fc0ed3c6d2163a289e3c3168664af666bac30d6c
Depend on Docker
iankouls@bhge-ai:~/Projects/2018/github.com/bhgedigital/enigma
>
```

Alex I

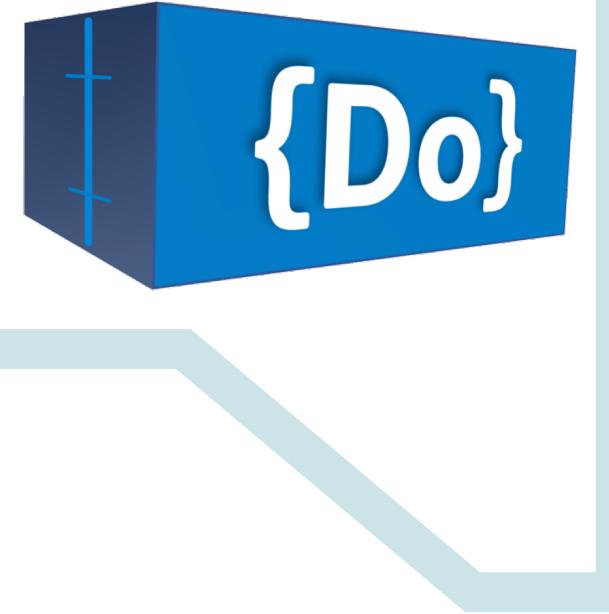


Depend-on-Docker Demo



```
iankouls@bhge-ai:~/Projects/2018/github.com/bhgedigital/demos/depend-on-docker
> curl -L http://bit.ly/dodocker | bash -s -- $(pwd)/dockercon18 > /dev/null
% Total    % Received % Xferd  Average Speed   Time   Time  Current
          Dload  Upload Total   Spent    Left  Speed
100  166  100  166    0     0  900      0 --:--:-- --:--:-- 902
100  780  100  780    0     0 2440      0 --:--:-- --:--:-- 2440
iankouls@bhge-ai:~/Projects/2018/github.com/bhgedigital/demos/depend-on-docker
> ls
create.sh  dockercon18
iankouls@bhge-ai:~/Projects/2018/github.com/bhgedigital/demos/depend-on-docker
> ls -al dockercon18/
total 96
drwxr-xr-x@ 15 iankouls  staff  510 Jun 13 19:50 .
drwxr-xr-x   4 iankouls  staff  136 Jun 13 19:49 ..
-rw-r--r--   1 iankouls  staff   82 Jun 13 19:49 .dockerignore
-rw-r--r--   1 iankouls  staff 1644 Jun 13 19:49 .env
-rw-r--r--   1 iankouls  staff 1291 Jun 13 19:49 .fun
drwxr-xr-x   4 iankouls  staff  136 Jun 13 19:49 Container-Root
-rw-r--r--   1 iankouls  staff  260 Jun 13 19:49 Dockerfile
-rwxr-xr-x   1 iankouls  staff  112 Jun 13 19:49 build.sh
-rwxr-xr-x   1 iankouls  staff  123 Jun 13 19:49 exec.sh
-rwxr-xr-x   1 iankouls  staff   65 Jun 13 19:49 logs.sh
-rwxr-xr-x   1 iankouls  staff   71 Jun 13 19:49 pull.sh
-rwxr-xr-x   1 iankouls  staff   71 Jun 13 19:49 push.sh
-rwxr-xr-x   1 iankouls  staff  199 Jun 13 19:49 run.sh
-rwxr-xr-x   1 iankouls  staff   60 Jun 13 19:49 status.sh
-rwxr-xr-x   1 iankouls  staff   63 Jun 13 19:49 stop.sh
iankouls@bhge-ai:~/Projects/2018/github.com/bhgedigital/demos/depend-on-docker
>
```

Depend-on-Docker Open Source



Depend-on-Docker was open sourced at DockerCon'18! {Apache 2.0}

Build your own Depend-on-Docker project & send us a link!

We will add your link in the Depend-on-Docker README (first 100 links)

We will send you a Depend-on-Docker sticker (while supplies last)

What are you going to build next?



References & Credits



GE Aircraft Engines

<https://www.ge.com/reports/every-two-seconds-aircraft-powered-ge-technology-takes-off-somewhere-world/>

GE Power Generation

<https://www.ge.com/reports/ultra-super-critical-badass-machines-help-make-30-percent-worlds-power/>

Container Hacks and Fun Images, Jessie Frazelle

<https://www.youtube.com/watch?v=cYsVvV1aVss>

Walt Disney Animation Studios ; screenplay by Jordan Roberts and Daniel Gerson & Robert L. Baird ; directed by Don Hall, Chris Williams ; producer Roy Conli. Big Hero 6. [Burbank, CA] :Walt Disney Studios, 2015. Available on DVD, Blu-ray and Disney Digital

Robot Fight <https://www.youtube.com/watch?v=cd6GbGMBoiU>

Microbots Presentation <https://www.youtube.com/watch?v=Eopgfzfsi7Q>

Roy Conli, Producer, Voice Artist, The Walt Disney Company

https://en.wikipedia.org/wiki/Roy_Conli

Paula Potter, Director, Rights and Clearances, The Walt Disney Company

Note: The Walt Disney Company is not a sponsor and is not in any way connected or affiliated with this presentation, in which the assets of Big Hero 6 appear.

Depend-on-Docker Project <https://github.com/bhgedigital/depend-on-docker>

Dr. Story Musgrave, Astronaut, Physician, Photographer, Public Speaker & more

https://en.wikipedia.org/wiki/Story_Musgrave

<http://www.storymusgrave.com/>

a Warner Bros. Pictures presentation ; in association with Village Roadshow Pictures - Groucho II Film partnership ; a Silver Pictures production ; produced by Joel Silver, written and directed by Andy Wachowski, Larry Wachowski. The Matrix. Burbank, CA :Warner Home Video, 2007. DVD

There is no spoon

<https://www.youtube.com/watch?v=XO0pcWxcROI>

GE SemTK Project - Paul Cudihy, Jenny Weisenberg, Justin McHugh, et.al. <https://github.com/ge-semtk/semtk>

Enigma DoD Project – Yekta Yazdani

<https://github.com/bhgedigital/enigma>

Google Images <https://images.google.com/>

Thank You!



Alex Iankoulski

<https://www.linkedin.com/in/alex-iankoulski/>

iankouls@bhge.com



Arun Subramaniyan

<https://www.linkedin.com/in/arunsubramaniyan/>

subramaa@bhge.com





Depend on Docker!