Introduction to Singularity: running containerized applications on HPC

Yucheng Zhang
Bioinformatics Engineer
TTS Research Technology

Feb.13, 2025

Upcoming workshop

Beginner's Guide to Bioinformatics – *Feb 27 (1–3 PM)*

Explore HPC, Linux basics, and foundational bioinformatics tools and skills.

Register Here

Overview

- What are containers and why should we use them?
- Container public registries
- Singularity basics
- Containers supporting Nvidia-GPU
- Build your own containers with Docker
- Simplify container pulling and module generation with container-mod

Pulling and Running Containerized HPC Applications

Introduction

Containers

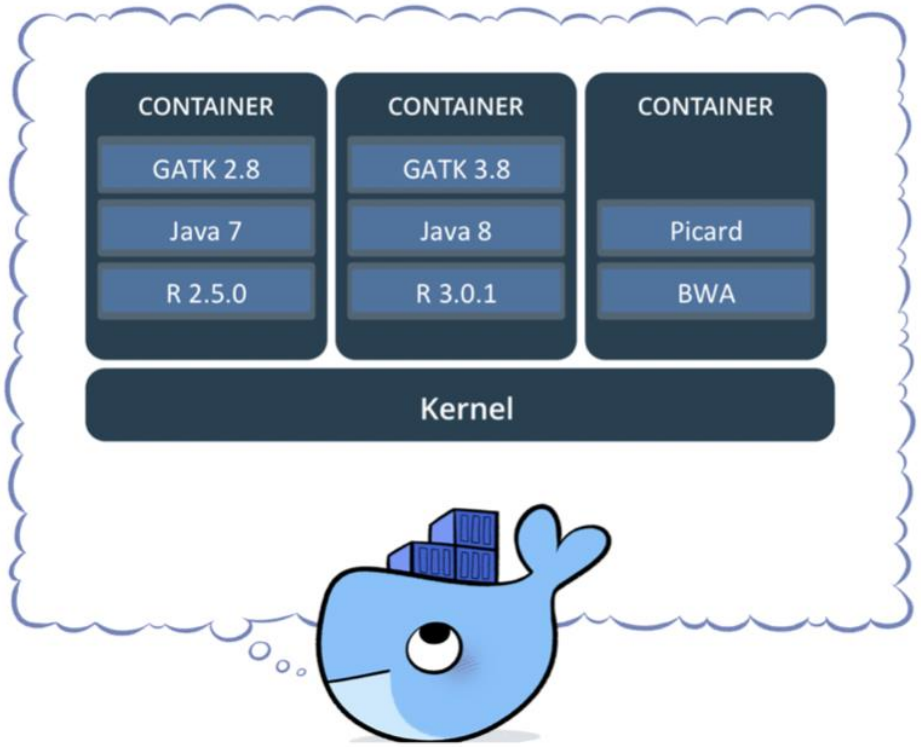
A **container** is an abstraction for a set of technologies that aim to solve the problem of how to get software to run reliably when moved from one computing environment to another.

A container **image** is simply a file (or collection of files) saved on disk that stores everything you need to run a target application or applications.

Registry: a place to store (and share) container images.

Why use containers?

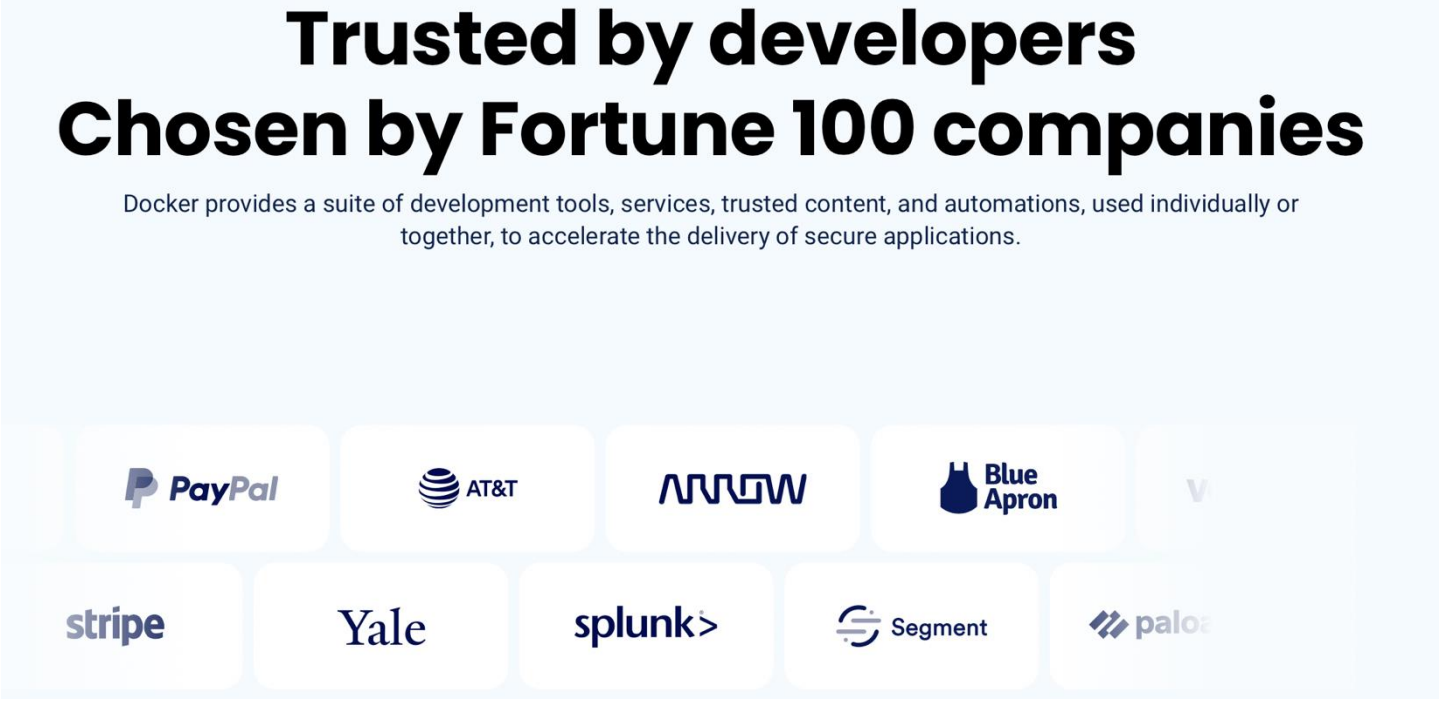
- Getting organized: containers keep things organized by isolating programs and their dependencies inside containers.
- Build once, run **almost** anywhere: containers allow us to package up our complete software environment and ship it to other computing environments.
- Reproducibility: containers can ensure identical versions of apps, libraries, compliers, etc.



Docker

The concept of containers emerged in 1970s, but they were not well known until the emergence of Docker containers in 2013.

Docker is an open source platform for building, deploying, and managing containerized applications.



Why not Docker on HPC?

Summarized by Stanford HPC center

- Docker requires a **daemon** running as **root** on all compute nodes, which poses security risks.
- All authenticated actions, such as login and push, execute as root. Therefore, those functions cannot be used simultaneously by multiple users on the same node.
- Docker uses **cgroups** to isolate containers, as does the Slurm scheduler, which uses **cgroups** to allocate resources to jobs and enforce limits. Those uses are unfortunately conflicting.
- Most importantly, ***allowing users to run Docker containers will give them* root *privileges* inside that container, thereby enabling them to access any of the filesystems on the cluster as root**. This opens the door to user impersonation, inappropriate file tampering or stealing.

Pulling and Running Containerized HPC Applications

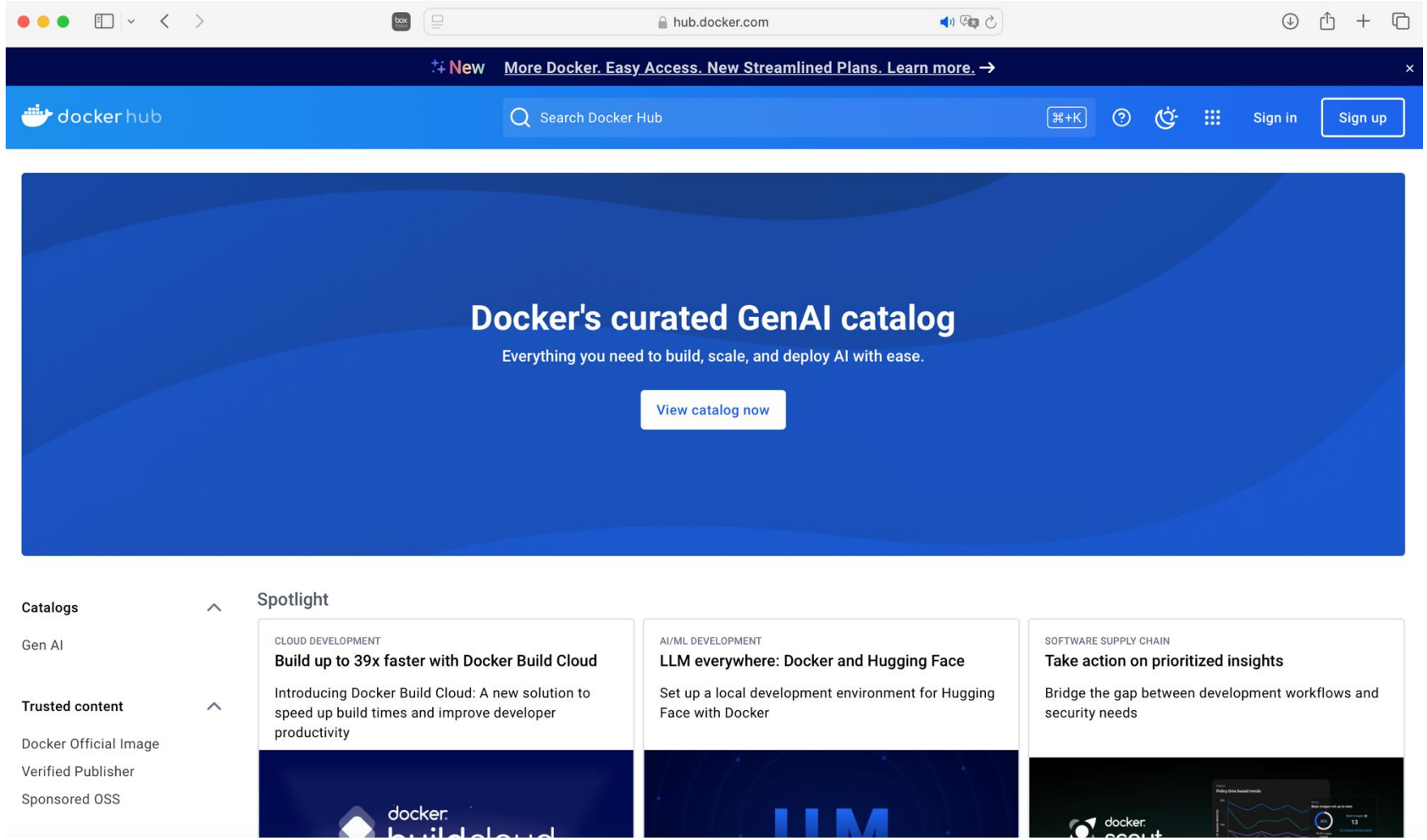
Container registries

Container registry

A **container registry** is a centralized platform where **container images** are stored, managed, and distributed. It acts like a repository for software containers, similar to how GitHub is used for managing source code.

Container registries allow users to **pull (download)** pre-built container images or **push (upload)** their own images for sharing or deployment across various environments, including HPC clusters, cloud platforms, and local machines.

Docker hub



https://hub.docker.com

docker hub

tuftsttsrt

Sign in | Sign up

TTS-Research Technology

Community Organization | Tufts University | Medford, MA | https://it.tufts.edu/about/organization/research-technology-rt | Joined October 21, 2024

Repositories

https://hub.docker.com/u/tuftsttsrt

Search by repository name

Displaying 1 to 10 of 10 repositories

tuftsttsrt/alphafold3
By tuftsttsrt • Updated 20 days ago
45 • 0

tuftsttsrt/metaphlan
By tuftsttsrt • Updated a month ago
25 • 0

tuftsttsrt/miniforge
By tuftsttsrt • Updated a month ago
24 • 0

tuftsttsrt/openmpi
By tuftsttsrt • Updated a month ago
42 • 0

RED HAT Quay.io EXPLORE TUTORIAL PRICING search SIGN IN

Quay.io Support Team is transitioning to Red Hat Customer Portal. For that reason, support(at)quay.io e-mail address will be disabled after January 1st. Please check out the following article for more information:

https://access.redhat.com/articles/7099134.

Quay [builds, analyzes, distributes] your container images

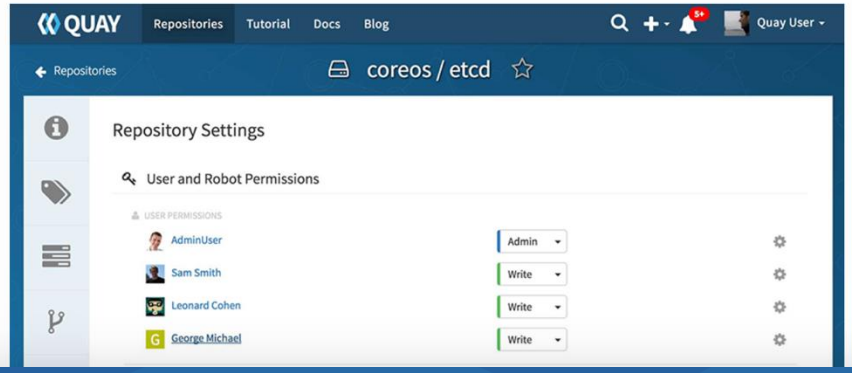
TRY FOR FREE ON PREMISES

TRY FOR FREE ON CLOUD

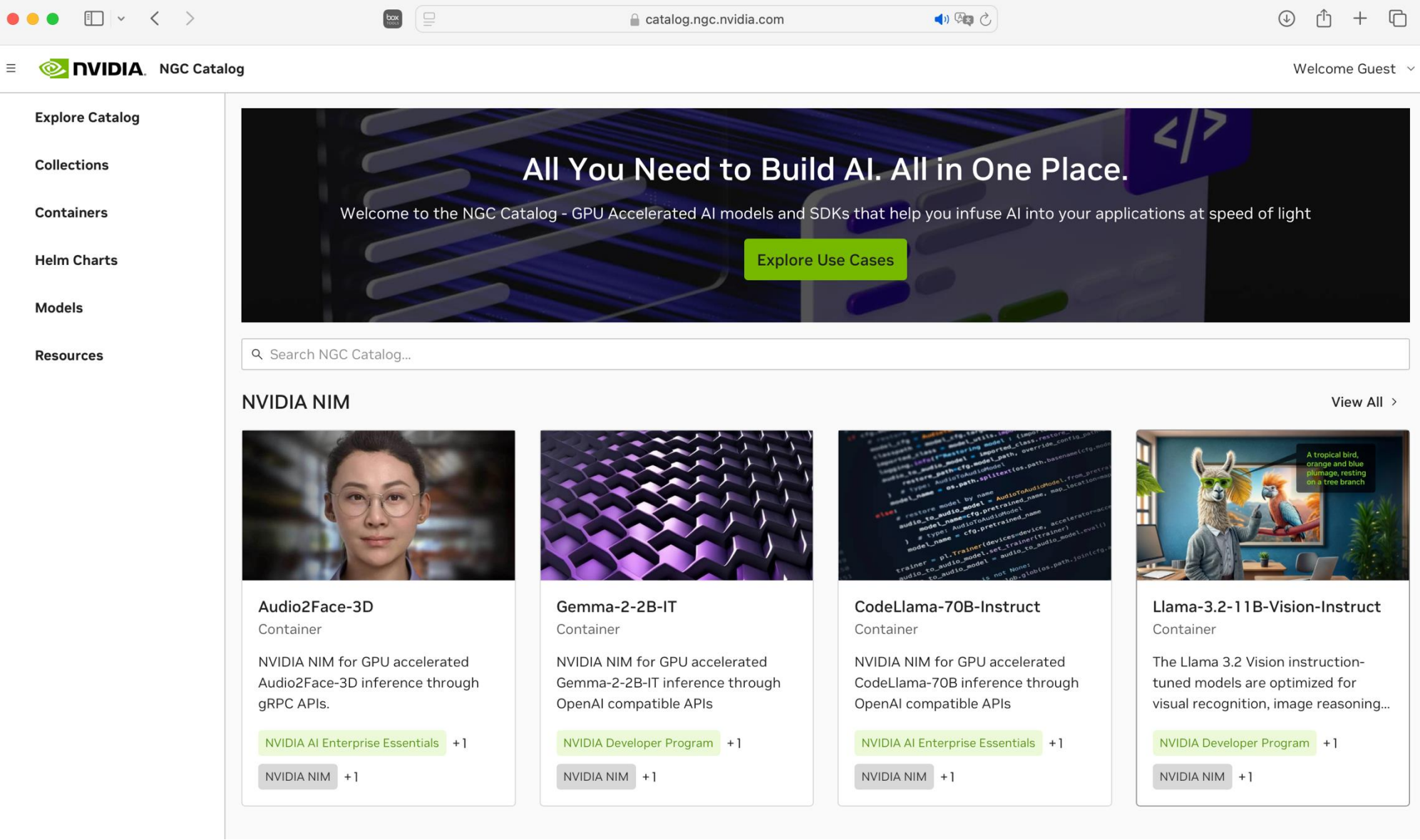
podman docker rkt

Store your containers securely

Ensure your apps are stored privately, with access that you control. Quay is teamwork optimized, with powerful access controls.



https://quay.io

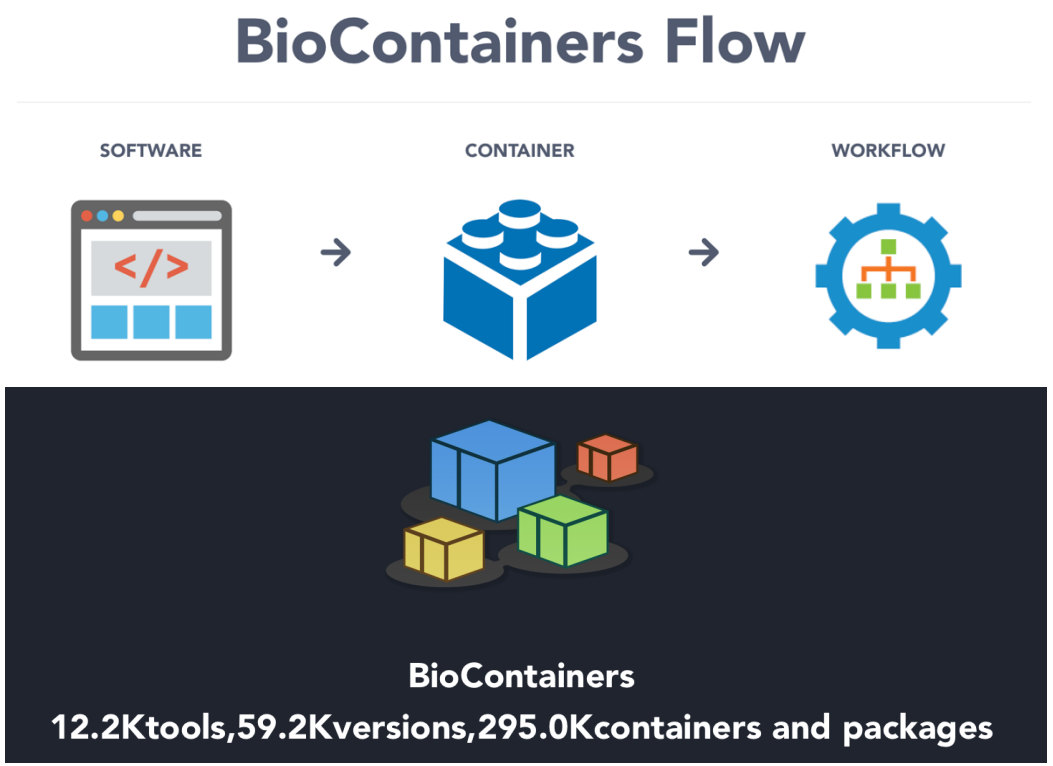


https://catalog.ngc.nvidia.com

BioContainers

BIOCONDA®

- BioContainers is integrated with Bioconda, which is the conda channel for bioinformatics applications.
- BioContainers registry is the largest registry for bioinformatics applications.
- As of today, BioContainers provides containers for over 13 thousand bioinformatics applications.



You can find almost all bioinformatics applications from here:
https://bioconda.github.io/conda-package_index.html

Pulling and Running Containerized HPC Applications

Singularity basics

Load singularity module

```
module purge
module load singularity/3.8.4
module list
```

```
$ module avail singularity

   singularity/2.6.1    singularity/3.1.0    singularity/3.5.3
   singularity/3.6.1    singularity/3.8.4 (D)

$ module load singularity
$ module list
Currently Loaded Modules:
  1) squashfs/4.4   2) singularity/3.8.4
```

Singularity basics

Detailed singularity user guide is available at: **sylabs.io/guides/3.8/user-guide**

singularity [options] <subcommand> [subcommand options …]

- **pull**
- **exec**
- **shell**
- build
- run
- push
- Instance
- help
- …

Singularity: Docker for HPC systems

- Singularity was developed in 2015 as an open-source project by researchers at Lawrence Berkeley National Laboratory (LBNL) led by Gregory Kurtzer.
- Singularity is emerging as the containerization framework of choice in HPC environments.
    1. **Enable researchers to package entire scientific workflows, libraries, and even data.**
    2. **Can use docker images.**
    3. **Does not require root privileges to run.**

pull: download a container from a given URI

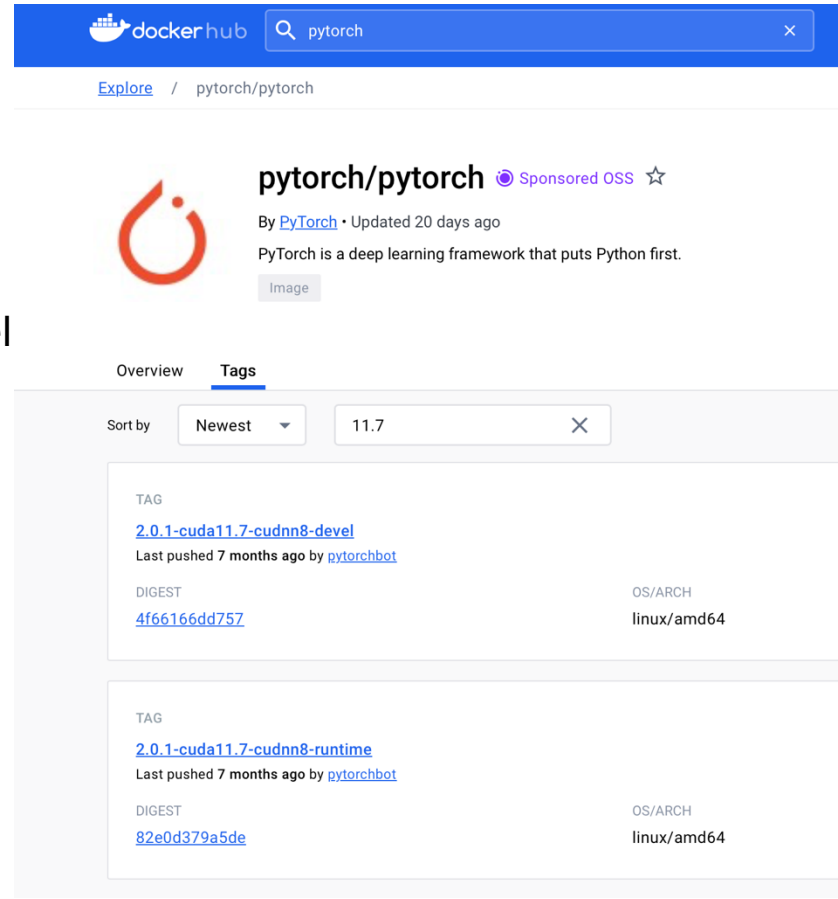
singularity pull [output file] URI

Example:

singularity pull pytorch_2.0.1.sif **docker://**pytorch/pytorch:2.0.1-cuda11.7-cudnn8-devel

Custom Name

URI



Complete supported URIs list can be found here:

https://docs.sylabs.io/guides/3.8/user-guide/cli/singularity_pull.html

pull: examples

```
docker pull quay.io/biocontainers/blast:<tag>

(see `blast/tags`_ for valid values for ``<tag>``)
```

docker pull quay.io/biocontainers/blast:2.15.0--pl5321h6f7f691_1 → **singularity** pull **docker://**quay.io/biocontainers/blast:2.15.0--pl5321h6f7f691_1

TAG

2.1.2-cuda11.8-cudnn8-devel

Last pushed **14 days ago** by pytorchbot

docker pull pytorch/pytorch:2.1.…

DIGEST	OS/ARCH	COMPRESSED SIZE
66b41f1755d9	linux/amd64	8.48 GB

docker pull pytorch/pytorch:2.1.2-cuda11.8-cudnn8-devel → **singularity** pull **docker://**pytorch/pytorch:2.1.2-cuda11.8-cudnn8-devel

TAG

2.15.0.post1-gpu

Last pushed **a month ago** by tensorflowpackages

docker pull tensorflow/tensorflo…

DIGEST	OS/ARCH	COMPRESSED SIZE
dc97d4feec5d	linux/amd64	3.45 GB

docker pull tensorflow/tensorflow:2.15.0.post1-gpu → **singularity** pull **docker://**tensorflow/tensorflow:2.15.0.post1-gpu

reddit.com

reddit | r/docker | Search in r/docker | Get App | Log In

Home
Popular
Answers BETA

TOPICS
Internet Culture (Viral)
Games
Q&As
Technology
Pop Culture
Movies & TV
See more

RESOURCES
About Reddit
Advertise

r/docker • 1 yr. ago
travprev

How do you know you're downloading a safe / legitimate container?

It seems like the docker libraries are the wild west... possibly worse than the Play Store for Android. How do you know, or how can you verify that the container you're downloading is safe, free of spyware, malware, etc.?

Sorry if this seems like a 101 question, but I am very new to docker. Thanks.

16 | 21 | Share

Add a comment

Sort by: Best | Search Comments

elementary_os • 1y ago

the first thing is we don't download containers, we pull images and create a container from it. The images marked as "Trusted build" are built by Docker on their server from the sources given by the user. You can easily check the Dockerfile from where the image have been built to check for malicious code.

You also have the 'official' images (those that does not start with 'somthing/') that are officially supported by Docker. If you trust Docker, inc, you can trust those image.

29 | Reply | Award | Share

r/docker | Join

Docker: An open source project to pack, ship and run any application as a lightweight container

[Docker](http://www.docker.io) is an open-source project to easily create lightweight, portable, self-sufficient containers from...

Show more

Public

222K Members | 45 Online | Top 1% Rank by size

r/selfhosted • 24 days ago
Your favorite SelfHosting Youtubers?
236 upvotes · 112 comments

r/falloutnewvegas • 1 yr. ago
How do you know if a mod is working?
3 upvotes · 3 comments

r/tryhackme • 1 yr. ago
How do you find the IP of a server you are trying to access?

https://www.reddit.com/r/docker/comments/1ax26jp/how_do_you_know_youre_downloading_a_safe/

shell: run a shell within a container

singularity shell [options] image

Example:

singularity shell pytorch_2.1.1.sif

```
Singularity> more /etc/os-release
NAME="Ubuntu"
VERSION="20.04.5 LTS (Focal Fossa)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 20.04.5 LTS"
VERSION_ID="20.04"
```

```
Singularity> python
Python 3.10.11 (main, Apr 20 2023, 19:02:41) [GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import torch
>>> torch.cuda.is_available()
True
>>> torch.cuda.get_device_name(0)
'NVIDIA A100 80GB PCIe'
```

Singularity> exit ## To exit container, and go back to host

exec: run executables/scripts

singularity exec [options] image command

Examples:

singularity exec pytorch_2.1.1.sif **python**

singularity exec r_4.3.1_scrnaseq.sif **Rscript** myscript.R

singularity exec trinityrnaseq_trinityrnaseq:2.15.1.sif **Trinity -h**

Cache

$ ncdu $HOME

```
ncdu 1.15.1 ~ Use the arrow keys to navigate
--- /cluster/home/yzhang85 ------------------
   67.9 GiB [##########] /.singularity
    5.1 GiB [          ] /.apptainer
    1.3 GiB [          ] /R
  797.7 MiB [          ] /.local
  595.6 MiB [          ] /.conda
  480.3 MiB [          ] /.vscode-server
  341.7 MiB [          ] /.cache
  318.7 MiB [          ] /nf-core
  313.1 MiB [          ]  r-base_4.3.2.sif
  223.1 MiB [          ] /.config
  189.9 MiB [          ] /ood_tufs_apps
  185.4 MiB [          ] /bin
  140.4 MiB [          ] /svn
   76.2 MiB [          ] /ondemand
   74.2 MiB [          ] /.nextflow
```

To mitigate this, users can either run the singularity pull command with --disable-cache

singularity pull --disable-cache URI

or manually clean $HOME/.singularity/cache

or export **SINGULARITY_CACHEDIR**=/cluster/tufts/XXXX

Don't run ncdu on login nodes

Pulling and running Containerized HPC Applications

Environment modules

NGC container environment modules

NGC container environment modules are lightweight wrappers that make it possible to transparently use NGC containers as environment modules.

1. Allow HPC users to utilize familiar environment module commands.
2. Leverage all the benefits of containers, including portability and reproducibility.

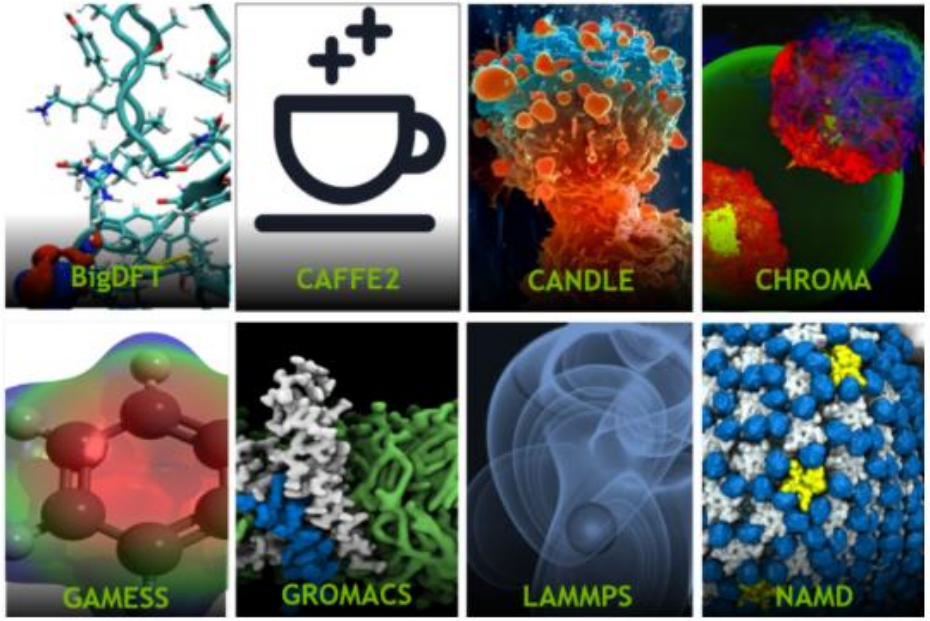
https://github.com/NVIDIA/ngc-container-environment-modules

Simplifying HPC Workflows with NVIDIA NGC Container Environment Modules

By Akhil Docca and Scott McMillan

Discuss (2) 0 Like

Tags: AI, Deep Learning, HPC / Supercomputing, machine learning, NGC, singularity



Nvidia GPU-optimized tools for deep learning, machine learning, and high-performance computing.

https://catalog.ngc.nvidia.com/

```
[yzhang85@s1cmp006 ~]$ module load ngc
[yzhang85@s1cmp006 ~]$ module av

------------------------------------------------------------------ /cluster/tufts/ngc/modules ------------------------------------------
   gromacs/2021.3    gromacs/2023      lammps/10Feb2021    nvhpc/21.5    parabricks/4.4.0-1            (D)    tensorflow/2.15.0
   gromacs/2022.5    gromacs/2023.2    lammps/15Jun2023    nvhpc/21.9    pytorch/2.5.1-cuda12.1-cudnn9 (L)
```

NVIDIA NGC

PyTorch

 # Load ngc

module load ngc

 # Check available applications

module avail

 # Load and run specific tools

module load pytorch/2.5.1-cuda12.1-cudnn9

No need to load cuda and cudnn modules

```
[yzhang85@s1cmp006 ~]$ module load ngc
[yzhang85@s1cmp006 ~]$ module load pytorch/2.5.1-cuda12.1-cudnn9
[yzhang85@s1cmp006 ~]$ python
Python 3.11.10 | packaged by conda-forge | (main, Oct 16 2024, 01:27:36) [GCC 13.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import torch
>>> # Check if GPU is available
>>> device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
>>> print(f"Using device: {device}")
Using device: cuda
>>> # Simple tensor computation on GPU
>>> x = torch.tensor([1.0, 2.0, 3.0], device=device)
>>> y = torch.tensor([4.0, 5.0, 6.0], device=device)
>>> result = x + y
>>> print(f"Result on {device}: {result}")
Result on cuda: tensor([5., 7., 9.], device='cuda:0')
```

Biocontainers @ Tufts

```
------------------------------------------------------------ /cluster/tufts/biocontainers/modules ------------------------------------------------------------
abcreg/0.1.0                        fasttree/2.1.11                 (D)  nf-core-fetchngs/1.11.0                 qiime2/2023.5
abyss/2.3.7                         filtlong/0.2.1                       nf-core-fetchngs/1.12.0          (D)  qiime2/2023.7
af2_binder_design/1.0.0             flye/2.9                             nf-core-funcscan/1.1.4                qiime2/2023.9
alphafold/2.3.0                     flye/2.9.1                           nf-core-funcscan/1.1.5           (D)  qiime2/2024.2          (D)
alphafold/2.3.1                     flye/2.9.2                      (D)  nf-core-hic/2.1.0                     qualimap/2.2.1
alphafold/2.3.2             (D)     fqtk/0.3.0                           nf-core-mag/2.5.2                     qualimap/2.3            (D)
alphafold3/3.0.0                    gatk4/4.2.6.1                        nf-core-mag/2.5.4                     r-bioinformatics/4.3.2
amplify/2.0.0                       gatk4/4.3.0.0                        nf-core-mag/3.0.0                     r-bioinformatics/4.4.0  (D)
angsd/0.939                         gatk4/4.5.0.0                   (D)  nf-core-mag/3.0.2                     r-scrnaseq/4.2.3
angsd/0.940                 (D)     genomad/1.8.1                        nf-core-mag/3.1.0                (D)  r-scrnaseq/4.3.1
bakta/1.9.3                         geomx_ngs_pipeline/3.1.1.6           nf-core-metatdenovo/1.0.0             r-scrnaseq/4.3.2
bbmap/38.93                         guppy/6.4.6                          nf-core-metatdenovo/1.0.1        (D)  r-scrnaseq/4.4.0        (D)
bbmap/38.96                 (D)     guppy/6.5.7                     (D)  nf-core-methylseq/2.6.0               r-shinyngs/1.8.5
bbtools/39.00                       hap.py/0.3.12                        nf-core-multiplesequencealign/1.0.0   raven-assembler/1.8.1
bcftools/1.13                       hclust2/1.0.0                        nf-core-nanoseq/3.1.0                 raxml-ng-mpi/1.2.0
bcftools/1.14                       hisat2/2.2.1                         nf-core-nanostring/1.2.1              raxml-ng-mpi/1.2.2      (D)
bcftools/1.17                       hmmer/3.3.2                     (D)  nf-core-nanostring/1.3.0         (D)  relion/4.0.1
bcftools/1.20               (D)     homer/4.11                           nf-core-pairgenomealign/1.0.0         relion/5.0              (D)
beast2/2.6.3                        htseq/2.0.2                          nf-core-pangenome/1.1.0               rmats2sashimiplot/2.0.4
beast2/2.6.4                        humann/3.8                           nf-core-pangenome/1.1.1               rmats2sashimiplot/3.0.0 (D)
beast2/2.6.6                (D)     impute2/2.3.2                        nf-core-pangenome/1.1.2          (D)  rnaquast/2.2.3
bedops/2.4.39                       iqtree/2.3.0                         nf-core-proteinfold/1.1.0             rosettafold2/default
bedtools/2.30.0                     iqtree/2.3.6                    (D)  nf-core-raredisease/2.0.1             rosettafold2na/0.2
bedtools/2.31.0                     kallisto/0.46.2                      nf-core-rnafusion/3.0.1               salmon/1.9.0
biobakery_workflows/3.0.0.a.7       kallisto/0.48.0                 (D)  nf-core-rnafusion/3.0.2          (D)  salmon/1.10.1           (D)
biobakery_workflows/3.1     (D)     kneaddata/0.12.0                     nf-core-rnaseq/3.14.0                 samtools/1.16
biopython/1.83                      kraken2/2.1.3                        nf-core-rnaseq/3.16.0                 samtools/1.17
blast/2.15.0                        krakentools/1.2                      nf-core-rnaseq/3.17.0            (D)  samtools/1.21
blast/2.16.0                (D)     macs2/2.2.7.1                        nf-core-rnasplice/1.0.2               scanpy/1.10.1
bowtie2/2.4.2                       macs3/3.0.0a6                        nf-core-rnasplice/1.0.3               scvelo/0.3.2
bowtie2/2.5.1               (D)     masurca/4.0.9                        nf-core-rnasplice/1.0.4          (D)  signalp6/6.0-fast
breseq/0.38.2                       masurca/4.1.0                   (D)  nf-core-sarek/3.4.0                   signalp6/6.0-slow       (D)
breseq/0.38.3                       medaka/1.11.1                        nf-core-sarek/3.4.1                   snpeff/5.2
breseq/0.39.0               (D)     megahit/1.2.9                        nf-core-sarek/3.4.3                   snpsift/5.2
busco/5.4.1                         megan/6.25.9                         nf-core-sarek/3.4.4              (D)  spaceranger/1.3.1
busco/5.4.7                 (D)     meme/5.5.5                           nf-core-scnanoseq/1.0.0               spaceranger/2.0.0
```

At Tufts HPC, we look forward to deploying bioinformatics applications based on containers wherever possible.

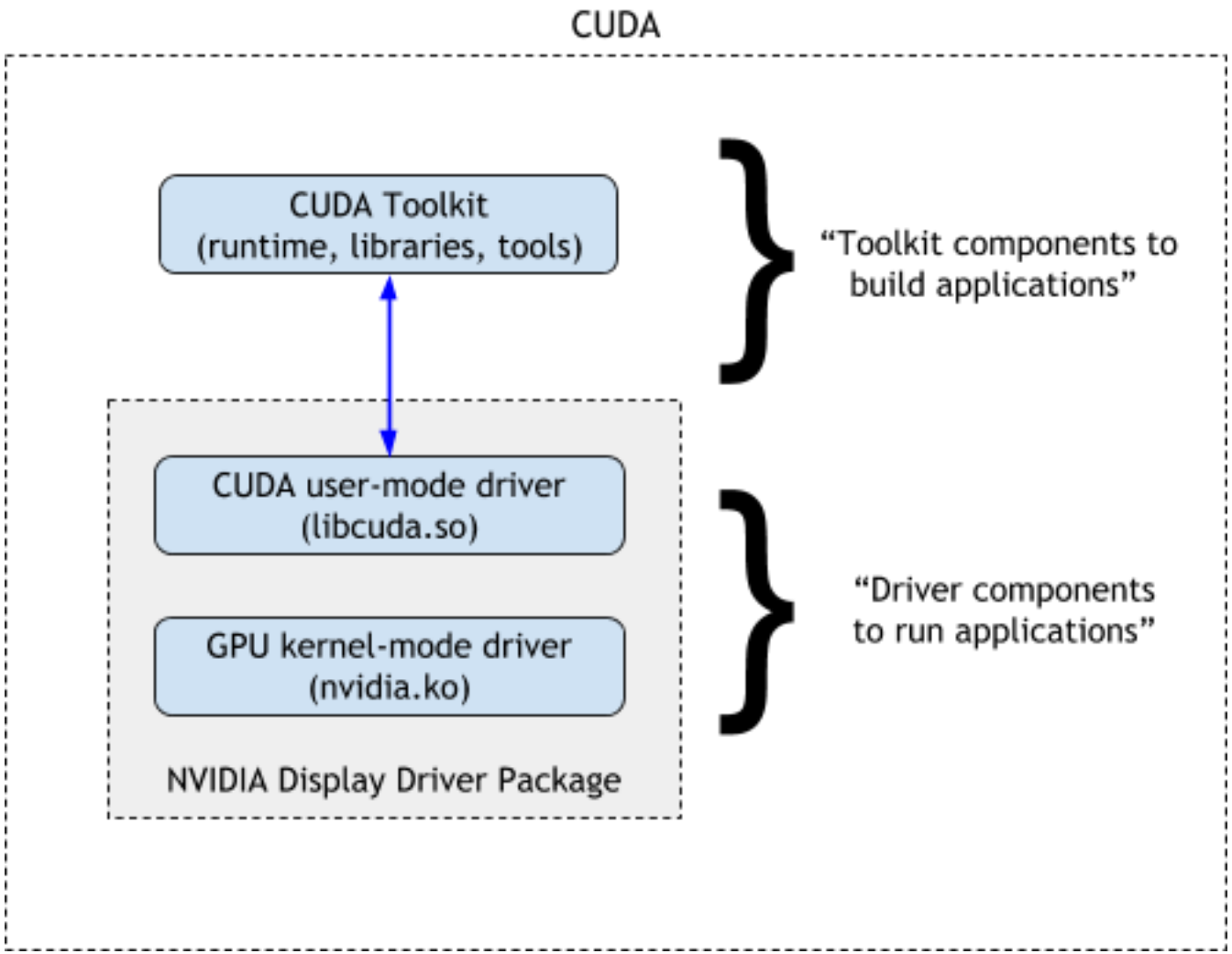
Pulling and Running Containerized HPC Applications

Nvidia GPU

CUDA Compatibility

In order to run a CUDA application, the system should have a CUDA enabled GPU and an NVIDIA display driver that is compatible with the CUDA Toolkit that was used to build the application itself. If the application relies on dynamic linking for libraries, then the system should have the right version of such libraries as well.

Detailed info can be found on Nvidia website.



nvidia-smi: the maximum supported CUDA Toolkit version

```
[yzhang85@d1cmp050 ~]$ nvidia-smi
Mon Dec 16 12:01:12 2024
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 535.154.05             Driver Version: 535.154.05   CUDA Version: 12.2     |
|-----------------------------------------+----------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | GPU-Util  Compute M. |
|                                         |                      |               MIG M. |
|=========================================+======================+======================|
|   0  NVIDIA L40                     On  | 00000000:CA:00.0 Off |                    0 |
| N/A   31C    P8              35W / 300W |      0MiB / 46068MiB |      0%      Default |
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI        PID   Type   Process name                            GPU Memory |
|        ID   ID                                                             Usage      |
|=========================================================================================|
|  No running processes found                                                             |
+-----------------------------------------------------------------------------------------+
```

pytorch/pytorch

Sponsored OSS

By PyTorch • Updated 2 months ago

PyTorch is a deep learning framework that puts Python first.

IMAGE

DATA SCIENCE | LANGUAGES & FRAMEWORKS | MACHINE LEARNING & AI | GEN AI

1.3K 10M+

Overview **Tags**

Sort by Newest

Filter tags

This may not work with our GPU node

TAG

2.5.1-cuda12.4-cudnn9-devel

Last pushed **2 months ago** by pytorchbot

```
docker pull pytorch/pytorch:2.5.1-cuda12.4-cudnn9-devel
```

Copy

Digest	OS/ARCH	Compressed size
14611869895d	linux/amd64	6.91 GB

TAG

2.5.1-cuda12.4-cudnn9-runtime

Last pushed **2 months ago** by pytorchbot

```
docker pull pytorch/pytorch:2.5.1-cuda12.4-cudnn9-runtime
```

Copy

Digest	OS/ARCH	Compressed size

--nv

To run Nvidia GPU-enabled containers, add **--nv** option to **exec**, **run** or **shell** commands.

singularity shell --nv pytorch_2.1.1.sif

singularity run --nv pytorch_2.1.1.sif

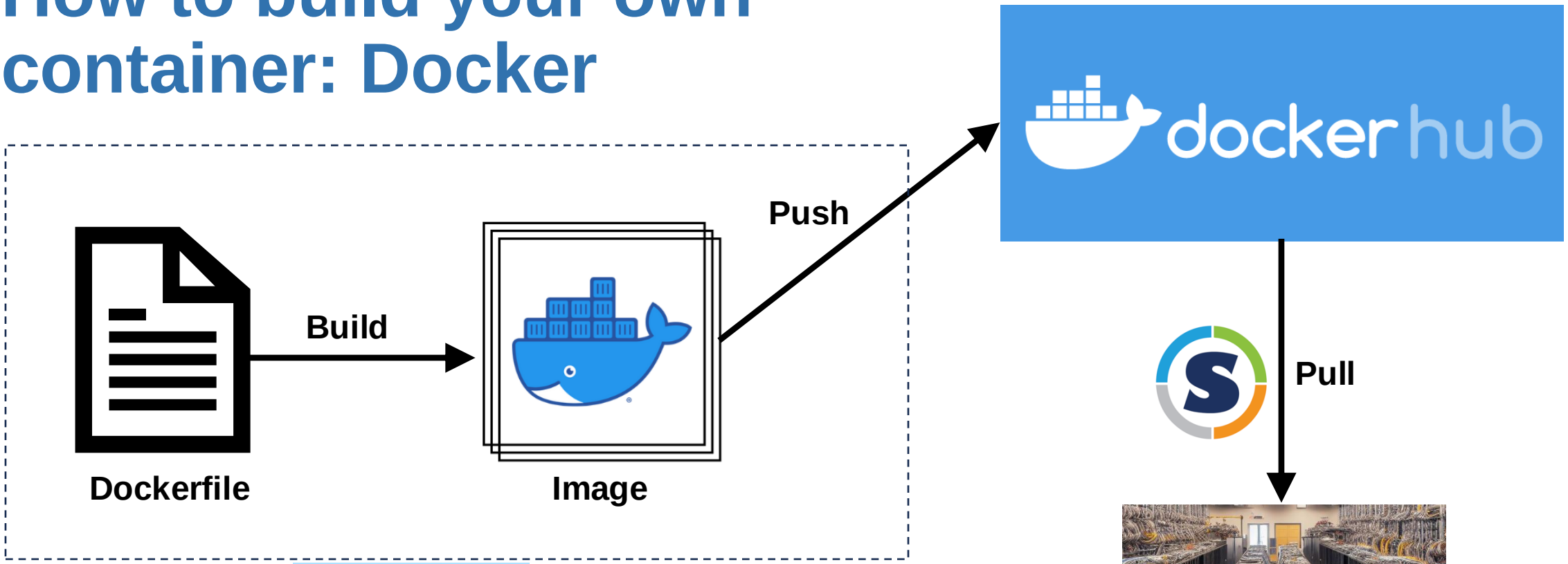
singularity exec --nv pytorch_2.1.1.sif **python**

There is no need to load cuda and cudnn modules.

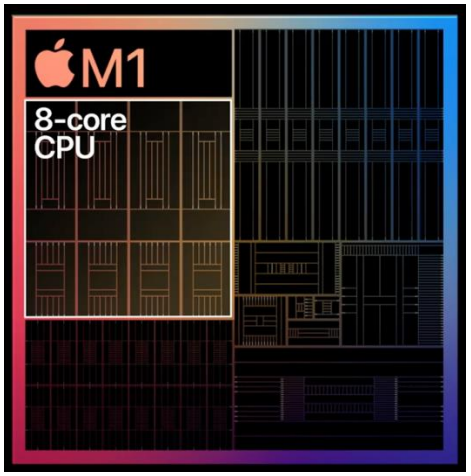
Pulling and Running Containerized HPC Applications

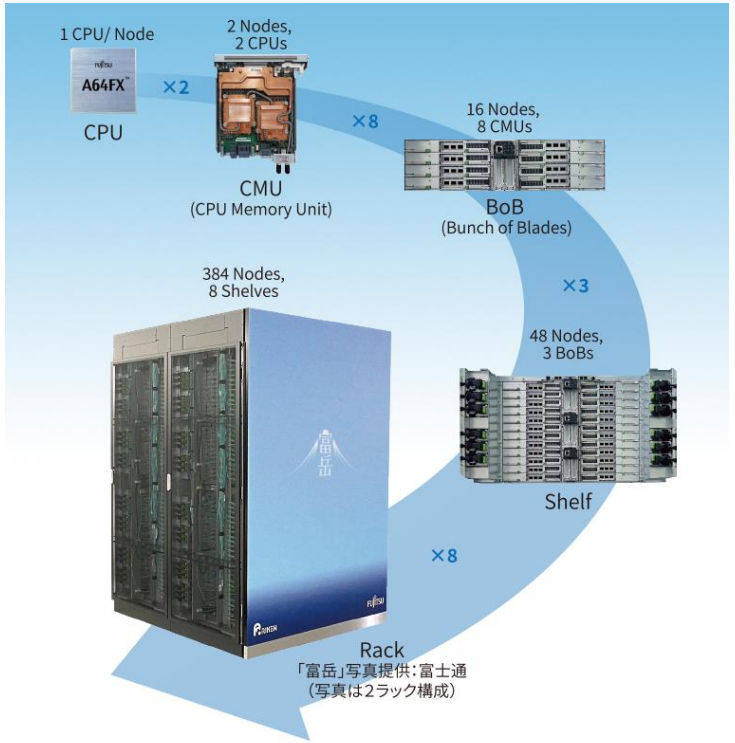
docker build

How to build your own container: Docker

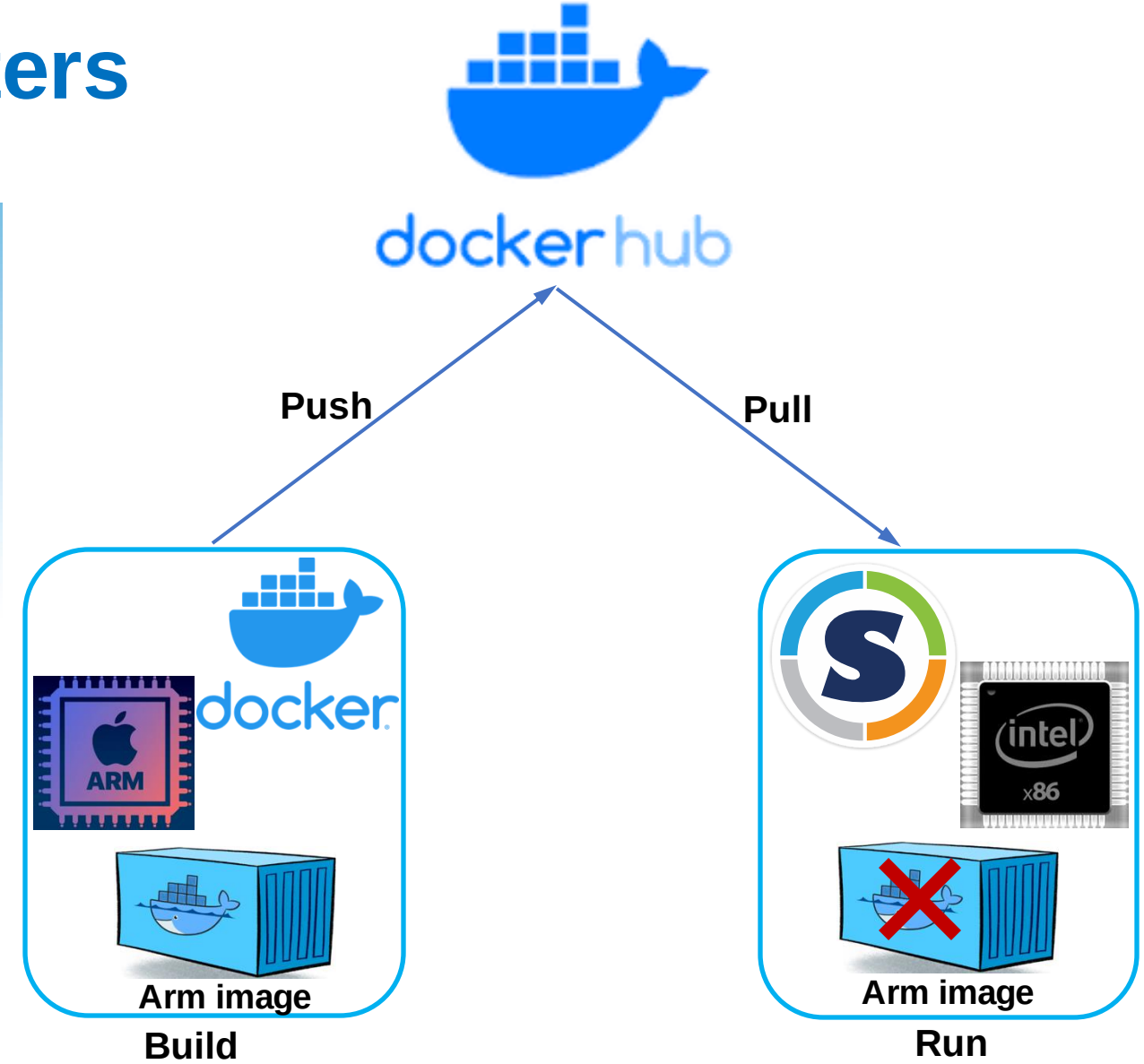


CPU Architecture Matters



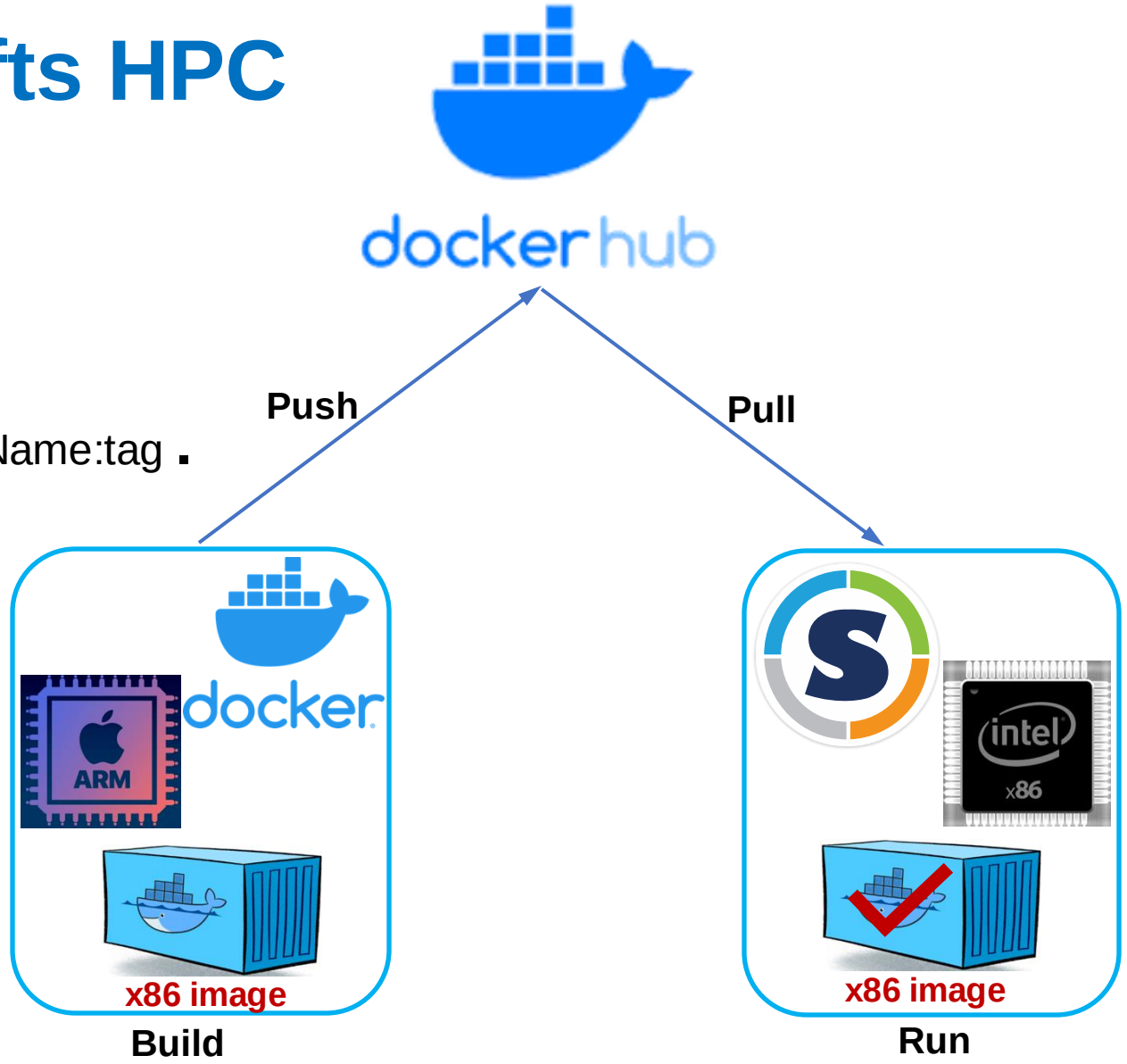


docker build -t UserName/AppName:tag .



Build containers for Tufts HPC using your Macbook

docker build **--platform linux/amd64** **-t** UserName/AppName:tag **.**



Pulling and Running Containerized HPC Applications

container-mod

container-mod

- container-mod streamlines the process of pulling containers from public registries and automatically generating ready-to-use modulefiles. It is a versatile tool designed for use by HPC system administrators and group managers to create and manage modules accessible to all HPC users or group members. Additionally, regular users can leverage container-mod to create personal modulefiles for their individual workflows, enhancing efficiency and reproducibility in HPC environments.

https://github.com/TuftsRT/container-mod

Pulling a container and generating its module

Pull the image

```
$ module load container-mod
$ container-mod pipe URI
```

Load the module

```
$ module load use.own
$ module load myapp/x.xx.x
```

Example1: fastp

```
[yzhang85@login-prod-02 ~]$ module load container-mod
[yzhang85@login-prod-02 ~]$ container-mod pipe -p docker://staphb/fastp:0.24.0
No profile specified. Running in personal mode!
Processing URI: docker://staphb/fastp:0.24.0 with the subcommand: pull
INFO:    Converting OCI blobs to SIF format
INFO:    Starting build...
Getting image source signatures
Copying blob 6414378b6477 skipped: already exists
Copying blob 863c05946a95 done
Copying blob 00b70a144829 done
Copying blob d566c8e1bb83 done
Copying config 88a3b79e69 done
Writing manifest to image destination
Storing signatures
2025/01/10 10:43:07  info unpack layer: sha256:6414378b647780fee8fd903ddb9541d134a1947ce092d08bdeb23a54cb3684ac
2025/01/10 10:43:08  info unpack layer: sha256:863c05946a95d85b7b6d14ba496ea3ddfcb0392c37ba6d350954b13803136ee4
2025/01/10 10:43:08  info unpack layer: sha256:00b70a14482901d10fb235282318e8fab6b87b5fe772e13c60bc9552fac5130c
2025/01/10 10:43:08  info unpack layer: sha256:d566c8e1bb83becfe86505a4d5174f20e652508f3562bd36ea008e6434435153
INFO:    Creating SIF file...
Processing URI: docker://staphb/fastp:0.24.0 with the subcommand: module
Remember to edit '/cluster/home/yzhang85/privatemodules/fastp/0.24.0.lua' stub (look for TODO: labels!)

Processing URI: docker://staphb/fastp:0.24.0 with the subcommand: exec
Generating executables provided by fastp version 0.24.0...
Successfully generated: fastp
+-------------------------------------------------------------------------------+
| To use this module, load the following modules:                               |
|                                                                               |
|     module load use.own                                                       |
|     module load fastp/0.24.0                                                  |
|                                                                               |
| The modulefile is located at: /cluster/home/yzhang85/privatemodules/fastp/0.24.0.lua |
+-------------------------------------------------------------------------------+
[yzhang85@login-prod-02 ~]$ module load use.own
[yzhang85@login-prod-02 ~]$ module load fastp/0.24.0
[yzhang85@login-prod-02 ~]$ fastp -h
option needs value: --html
usage: /usr/local/bin/fastp [options] ...
```

Containers supporting Jupyter

```
$ module load container-mod
$ container-mod pipe -j URI
```

or

```
$ container-mod pipe --jupyter URI
```

Requirements

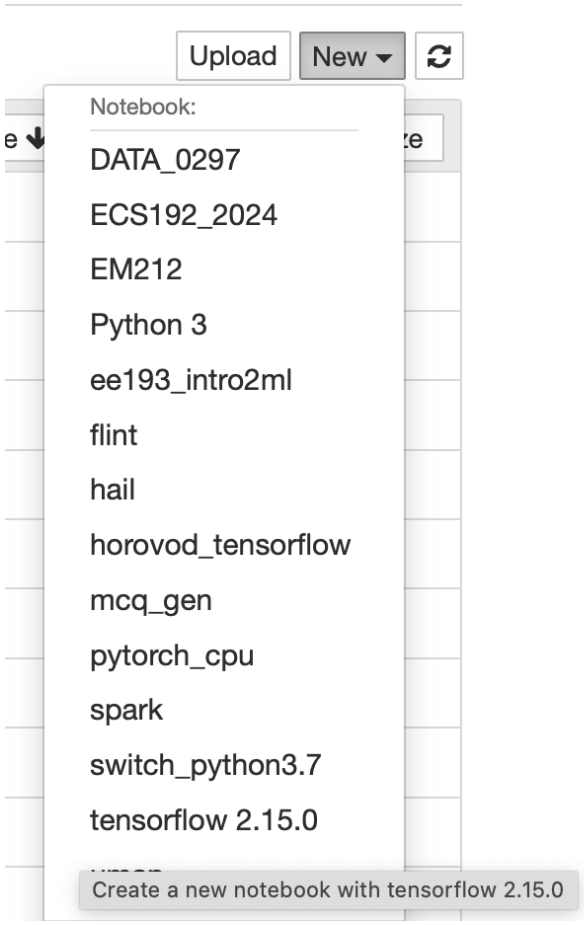
ipykernel is installed in the container

With **-j** or **--jupyter** option, the script will generate a **kernel.json** into **$HOME/.local/share/jupyter/kernels/XX/**

Example2: tensorflow

container-mod pipe **--jupyter** docker://tuftsttsrt/tensorflow:2.15.0

```
+-----------------------------------------------------------------------------------------+
| To use this module, load the following modules:                                         |
|                                                                                         |
|     module load use.own                                                                 |
|     module load tensorflow/2.15.0                                                       |
|                                                                                         |
| The modulefile is located at: /cluster/home/yzhang85/privatemodules/tensorflow/2.15.0.lua |
+-----------------------------------------------------------------------------------------+
Generating Jupyter kernel for tensorflow version 2.15.0
Jupyter kernel created: tensorflow-2.15.0
+-----------------------------------------------------------------------------+
| You can now launch Jupyter notebook/lab and select the kernel:              |
|                                                                             |
|    tensorflow 2.15.0                                                        |
|                                                                             |
| If you'd like to edit the kernel, you can find it at:                       |
|                                                                             |
|    /cluster/home/yzhang85/.local/share/jupyter/kernels/tensorflow-2.15.0    |
|                                                                             |
+-----------------------------------------------------------------------------+
```



TensorFlow

ondemand.pax.tufts.edu

jupyter Untitled3 Last Checkpoint: 11/16/2023 (autosaved)

Logout

File Edit View Insert Cell Kernel Help

Trusted | tensorflow 2.15.0

Run | Code

In [1]:
```
import tensorflow as tf

# Check if TensorFlow detects a GPU
print("Num GPUs Available: ", len(tf.config.list_physical_devices('GPU')))

if tf.config.list_physical_devices('GPU'):
    print("TensorFlow is using GPU(s):")
    for gpu in tf.config.list_physical_devices('GPU'):
        print(" -", gpu)
else:
    print("TensorFlow is not using GPU. Check your installation and configuration.")

# A simple computation to test the GPU
a = tf.constant([1.0, 2.0, 3.0], shape=[3, 1], name='a')
b = tf.constant([1.0, 2.0, 3.0], shape=[1, 3], name='b')
c = tf.matmul(a, b)

print("Result of matrix multiplication using TensorFlow:\n", c)
```

```
2025-01-10 15:52:35.538739: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slight
ly different numerical results due to floating-point round-off errors from different computation orders. To turn th
em off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2025-01-10 15:52:36.815169: E external/local_xla/xla/stream_executor/cuda/cuda_dnn.cc:9261] Unable to register cuDN
N factory: Attempting to register factory for plugin cuDNN when one has already been registered
2025-01-10 15:52:36.815200: E external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:607] Unable to register cuFFT
factory: Attempting to register factory for plugin cuFFT when one has already been registered
2025-01-10 15:52:36.908383: E external/local_xla/xla/stream_executor/cuda/cuda_blas.cc:1515] Unable to register cuB
LAS factory: Attempting to register factory for plugin cuBLAS when one has already been registered
2025-01-10 15:52:37.059869: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimize
d to use available CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 AVX512F AVX512_VNNI FMA, in other operations, rebuild TensorFlow with th
e appropriate compiler flags.
```

```
Num GPUs Available:  1
TensorFlow is using GPU(s):
 - PhysicalDevice(name='/physical_device:GPU:0', device_type='GPU')
Result of matrix multiplication using TensorFlow:
 tf.Tensor(
[[1. 2. 3.]
 [2. 4. 6.]
 [3. 6. 9.]], shape=(3, 3), dtype=float32)
```

```
2025-01-10 15:52:43.097276: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1929] Created device /job:localhost/
replica:0/task:0/device:GPU:0 with 38375 MB memory:  -> device: 0, name: NVIDIA A100-PCIE-40GB, pci bus id: 0000:a
f:00.0, compute capability: 8.0
```

Summary

- Always use the latest version of singularity modules
- To ensure that you're not wasting your time building your own containers, it's recommended to check if there are any publicly available containers that can serve your purpose.
- If you plan to use GPU containers, ensure compatibility between the CUDA version that was used to build the target application inside container and our GPU's CUDA driver version.
- When running containers that require GPU, make sure to include the **--nv** flag.
- Remember to use **ncdu**, a helpful tool, to regularly check and clean up $**HOME/.singularity** directories.

Thank you

Ticket: tts-research@tufts.edu
Email: yzhang85@tufts.edu
Consultation: https://go.tufts.edu/yucheng
Slides & Hands-on