EVIDEN
an atos business
science + computing

Linux Containers for HPC

2023 Update

Holger Gantikow

Juli 2023

tübix

Agenda

01. `whoami`

02. Intro

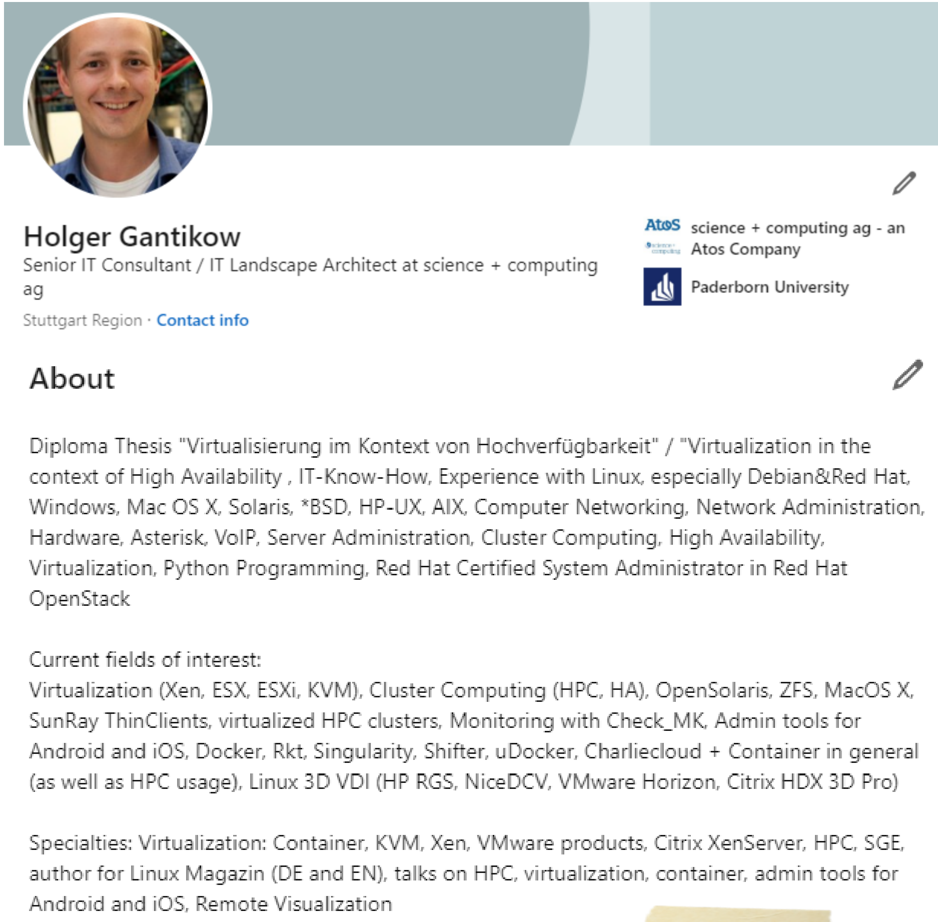
03. Container 101

04. Container Use Cases

05. Nimbix

06. Conclusion

01. whoami



Holger Gantikow
Senior IT Consultant / IT Landscape Architect at science + computing ag
Stuttgart Region · Contact info

science + computing ag - an Atos Company
Paderborn University

About

Diploma Thesis "Virtualisierung im Kontext von Hochverfügbarkeit" / "Virtualization in the context of High Availability , IT-Know-How, Experience with Linux, especially Debian&Red Hat, Windows, Mac OS X, Solaris, *BSD, HP-UX, AIX, Computer Networking, Network Administration, Hardware, Asterisk, VoIP, Server Administration, Cluster Computing, High Availability, Virtualization, Python Programming, Red Hat Certified System Administrator in Red Hat OpenStack

Current fields of interest:
Virtualization (Xen, ESX, ESXi, KVM), Cluster Computing (HPC, HA), OpenSolaris, ZFS, MacOS X, SunRay ThinClients, virtualized HPC clusters, Monitoring with Check_MK, Admin tools for Android and iOS, Docker, Rkt, Singularity, Shifter, uDocker, Charliecloud + Container in general (as well as HPC usage), Linux 3D VDI (HP RGS, NiceDCV, VMware Horizon, Citrix HDX 3D Pro)

Specialties: Virtualization: Container, KVM, Xen, VMware products, Citrix XenServer, HPC, SGE, author for Linux Magazin (DE and EN), talks on HPC, virtualization, container, admin tools for Android and iOS, Remote Visualization

Experience

science + computing ag - an Atos Company
14 yrs 9 mos

Senior IT Consultant / CAE IT Landscape Architect
Jul 2022 - Present · 1 yr
Baden-Württemberg, Germany

Senior Systems Engineer
Apr 2009 - Oct 2022 · 13 yrs 7 mos

System Engineer
2009 - Oct 2022 · 13 yrs 10 mos

Graduand
Oct 2008 - Mar 2009 · 6 mos

Education

Paderborn University
GOLD for Technology Leaders, Software Innovation Campus (SICP)
Sep 2022

Successfully completed the Atos Group Talent program "GOLD for Technology Leaders", with the Cambridge University's Institute for Manufacturing (IfM) ...see more

University of Cambridge
GOLD for Technology Leaders, Department of Engineering, Institute for Manufacturing (IfM)
Sep 2022

Successfully completed the Atos Group Talent program "GOLD for Technology Leaders", with the Cambridge University's Institute for Manufacturing (IfM) ...see more

Hochschule Furtwangen
Dipl. Inform. (FH), Coding, HPC, Clustering, Unix stuff :-)
2003 - 2009

Find me on Linkedin & Xing – feel free to reach out!

Atos
science + computing

Quick Facts

Focus on technical & scientific computing with 30 years of expertise

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

founded in 1989
as a university spin-off

science + computing

dedicated unit for
high-end & business
critical IT-Services

FY2020

€ 40.4M
external revenue

Atos

accompanied by Atos /
Bull Advanced
Computing Solutions

Tübingen

Berlin

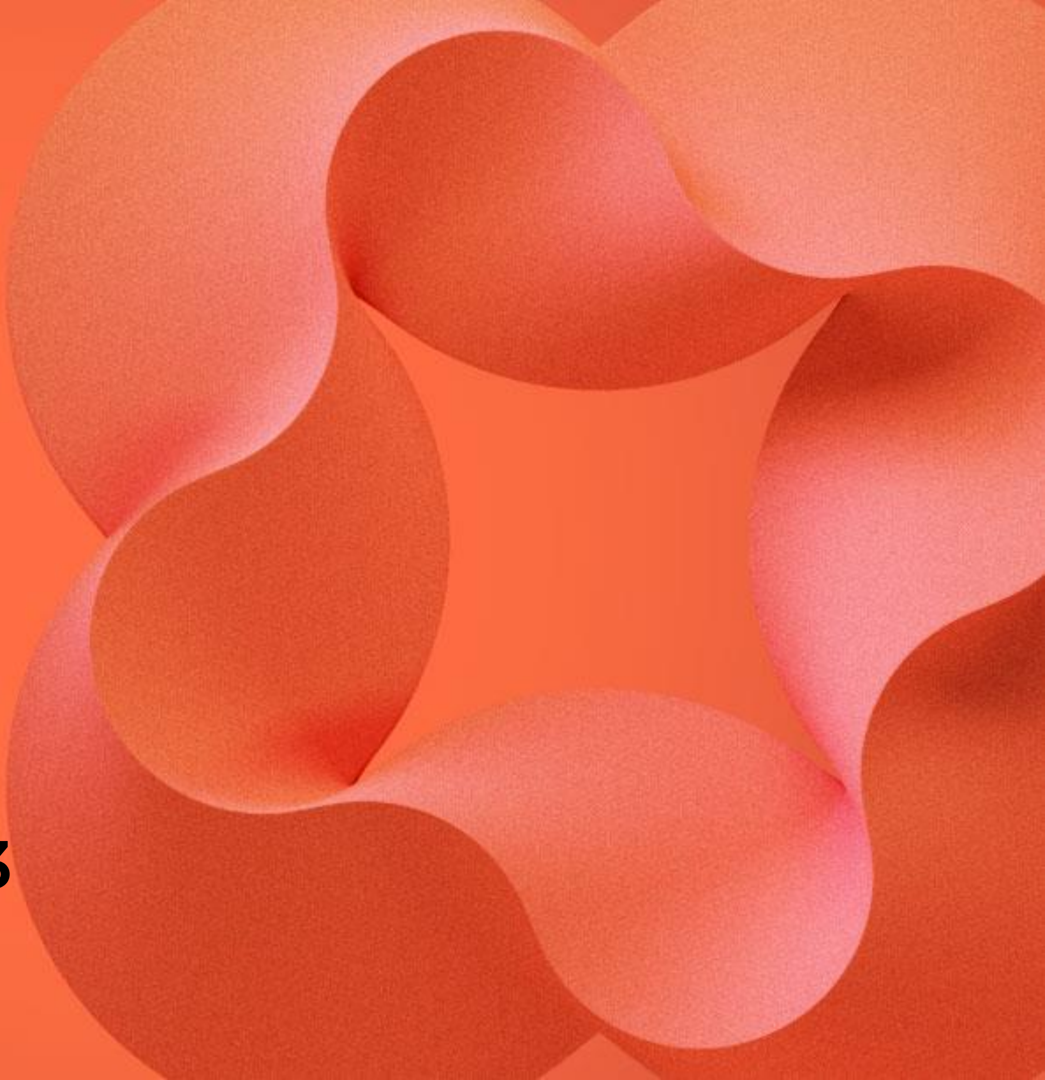
Munich

Düsseldorf

Timișoara

Brașov

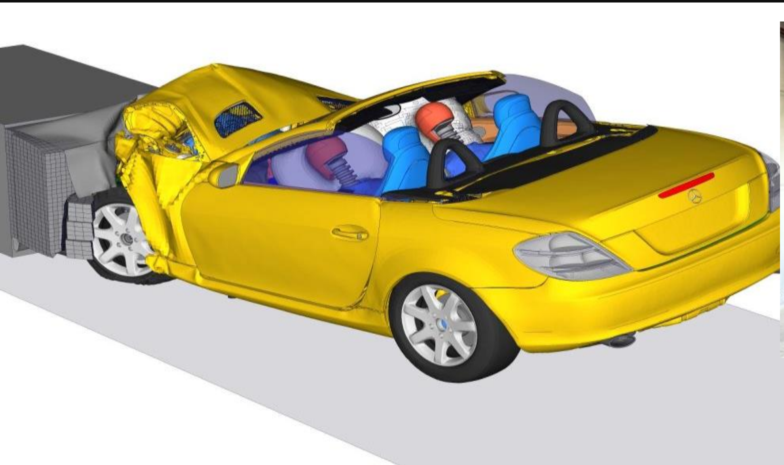
EVIDEN

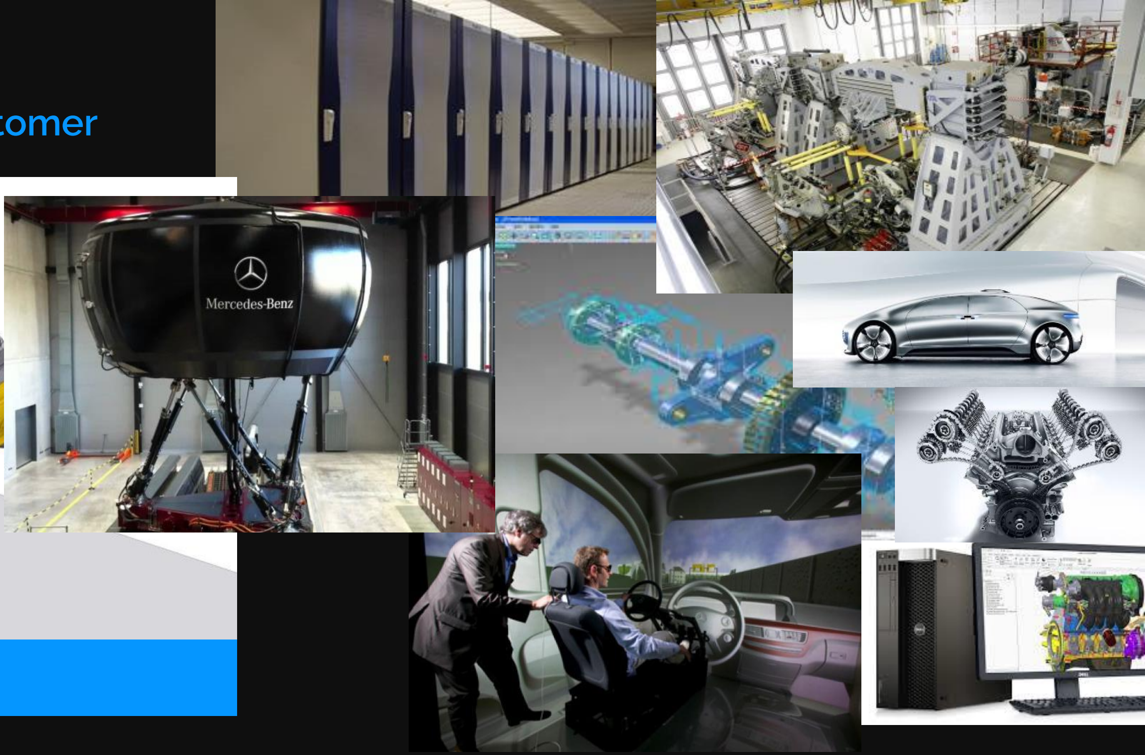


Starting Summer 2023

HPC Services

Operation and Projects close to the Customer





Our job and expertise

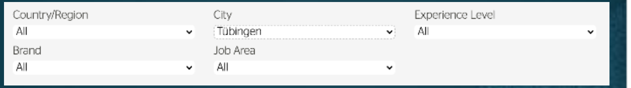
- *Operations of large scale heterogenous environments*
- *Consulting projects in challenging environments*
- *R&D + (Co-)Innovation projects*
- *Automation*
- *Close collaboration and customer-centric solutions*

- CAD / CAE / CAT, Virtual Reality
- Driver Assistance Systems / Autonomous Driving
- 3D accelerated Linux Engineering Work Places
- Virtualization, Ways to the Cloud, Hybrid HPC
- Data Management, Backup

Wir sind bei der science + computing immer wieder auf der Suche nach…

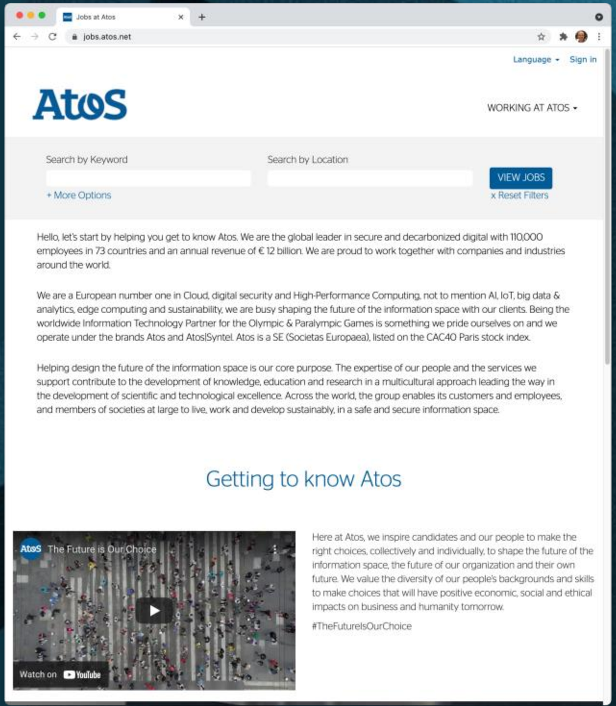
- *Systems Engineers CAE/Linux*
- *Linux / IT / HPC Consultants (CAE/Linux)*
- *Systems Engineers Cloud*
- *HPC Performance Engineers*
- *Database Engineers*
- *Software Engineers*
- *Praktika, Abschlussarbeiten, Werkstudent:innen…*

- Aktuelle Angebote unter **jobs.atos.net – Filter beachten**



Country/Region: All	City: Tübingen	Experience Level: All
Brand: All	Job Area: All	

- Gerne auch **initiativ** via Mail anfragen!
  - holger.gantikow@atos.net



Hello, let's start by helping you get to know Atos. We are the global leader in secure and decarbonized digital with 110,000 employees in 73 countries and an annual revenue of € 12 billion. We are proud to work together with companies and industries around the world.

We are a European number one in Cloud, digital security and High-Performance Computing, not to mention AI, IoT, big data & analytics, edge computing and sustainability, we are busy shaping the future of the information space with our clients. Being the worldwide Information Technology Partner for the Olympic & Paralympic Games is something we pride ourselves on and we operate under the brands Atos and Atos|Syntel. Atos is a SE (Societas Europaea), listed on the CAC40 Paris stock index.

Helping design the future of the information space is our core purpose. The expertise of our people and the services we support contribute to the development of knowledge, education and research in a multicultural approach leading the way in the development of scientific and technological excellence. Across the world, the group enables its customers and employees, and members of societies at large to live, work and develop sustainably, in a safe and secure information space.

Getting to know Atos

Here at Atos, we inspire candidates and our people to make the right choices, collectively and individually, to shape the future of the information space, the future of our organization and their own future. We value the diversity of our people's backgrounds and skills to make choices that will have positive economic, social and ethical impacts on business and humanity tomorrow.

#TheFutureIsOurChoice

02. Intro

Customers' Innovation is driven by HPC

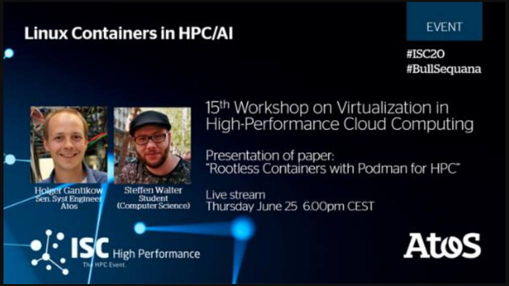
(High Performance Computing)

Customer‘s Hot Topics

CAE Cloudification, HPCaaS, Container

CAE Cloudification

- Design and implementation of hybrid HPC environments.
- Customization of technical workflows to overcome limited cloud capabilities
- Development of solutions for asynchronous data transfer
- Optimization of total cost of ownership / busting scenarios



Linux Containers in HPC/AI

EVENT

#ISC20
#BullSequana

Holger Gantikow
Sen. Syst Engineer
Atos

Steffen Walter
Student
(Computer Science)

15th Workshop on Virtualization in High-Performance Cloud Computing

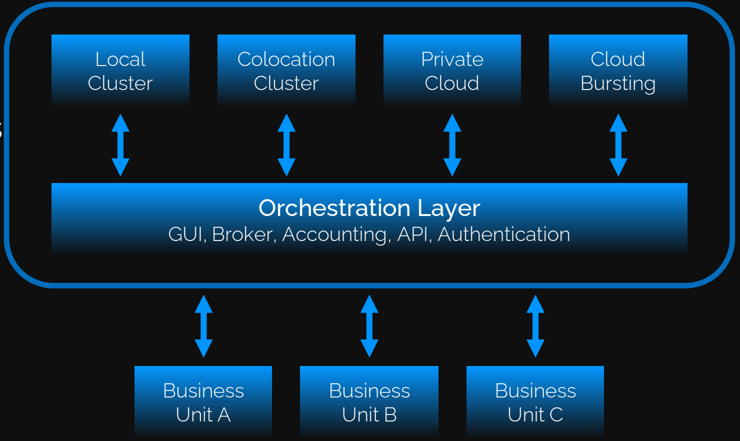
Presentation of paper:
"Rootless Containers with Podman for HPC"

Live stream
Thursday June 25 6.00pm CEST

ISC High Performance
The HPC Event.
Atos

HPC & CAE as a Service

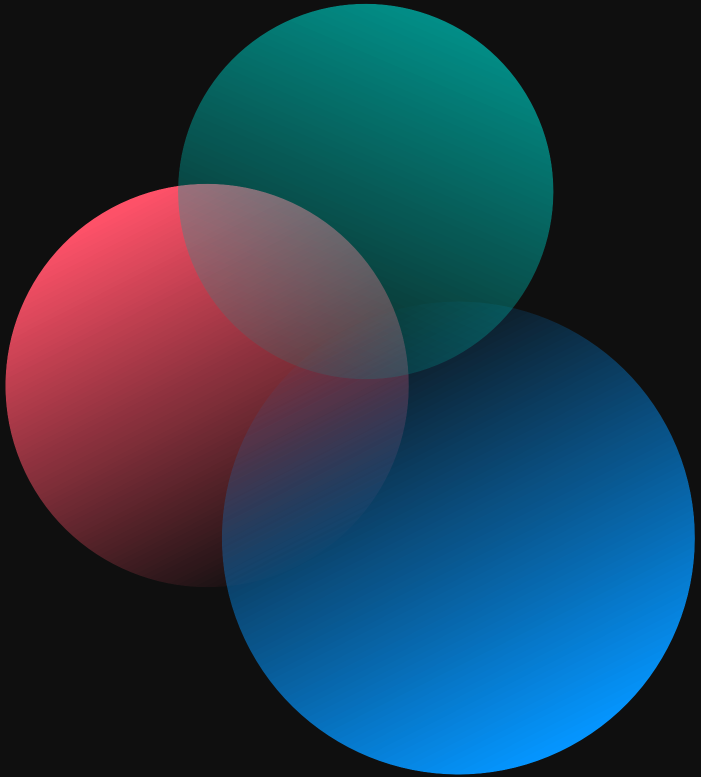
- Reduce project management and multi-provider management efforts
- Create private cloud CAE platforms



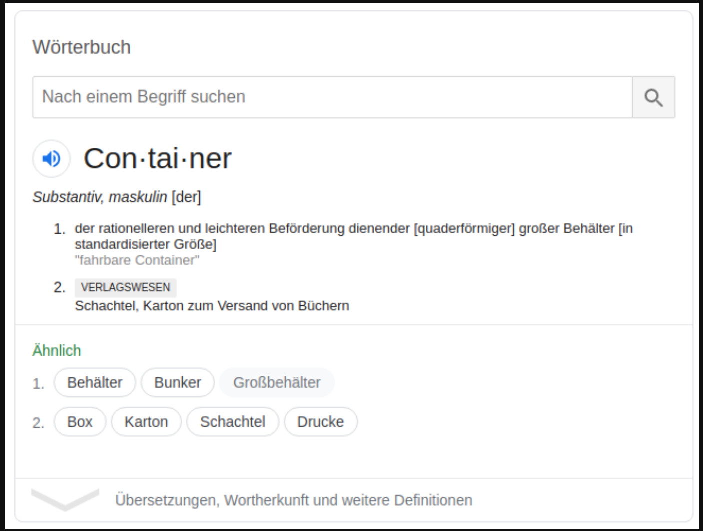
Containers for CAE/HPC

- Driver for many commercial use cases
  - Multi-site HPC Grids, K8s, Cloud HPC, Security, Standardization

03. Container 101



What is a container?



Wörterbuch

Nach einem Begriff suchen

Con·tai·ner

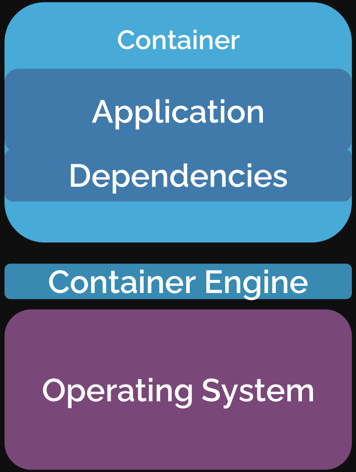
Substantiv, maskulin [der]

1. der rationelleren und leichteren Beförderung dienender [quaderförmiger] großer Behälter [in standardisierter Größe]
   "fahrbare Container"
2. VERLAGSWESEN
   Schachtel, Karton zum Versand von Büchern

Ähnlich

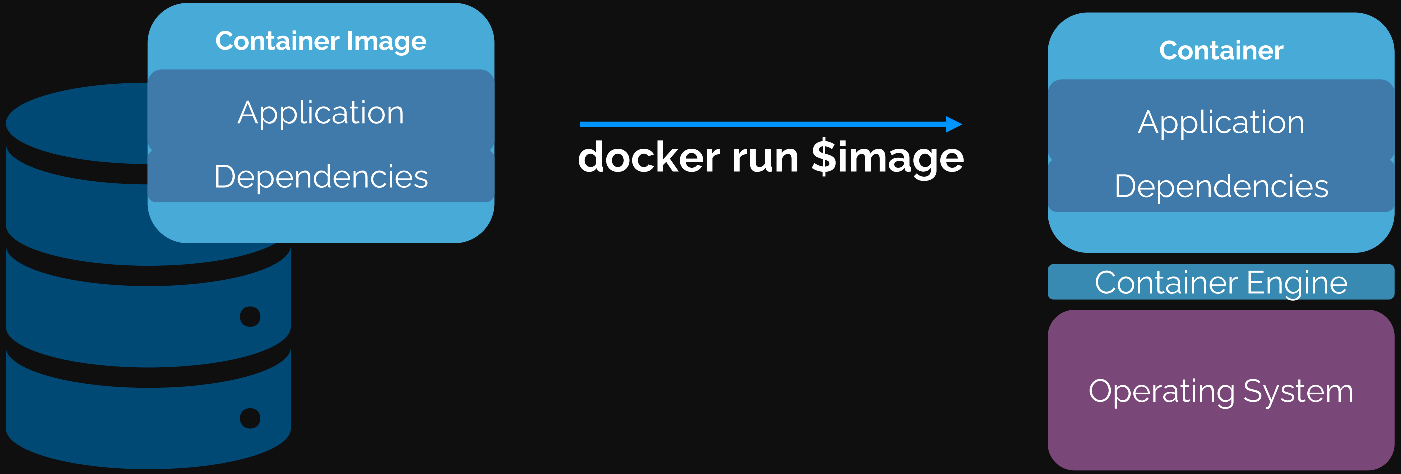
1. Behälter Bunker Großbehälter
2. Box Karton Schachtel Drucke

Übersetzungen, Wortherkunft und weitere Definitionen



- Standardized packaging unit to simplify handling - has revolutionized transport
- Same in IT: standardized unit of applications + dependencies.
- Often offers advantages in distribution and operation
- Requires a container engine/runtime + underlying operating system (usually Linux)
- Differentiation *container image* + *container*

Container Image vs Container

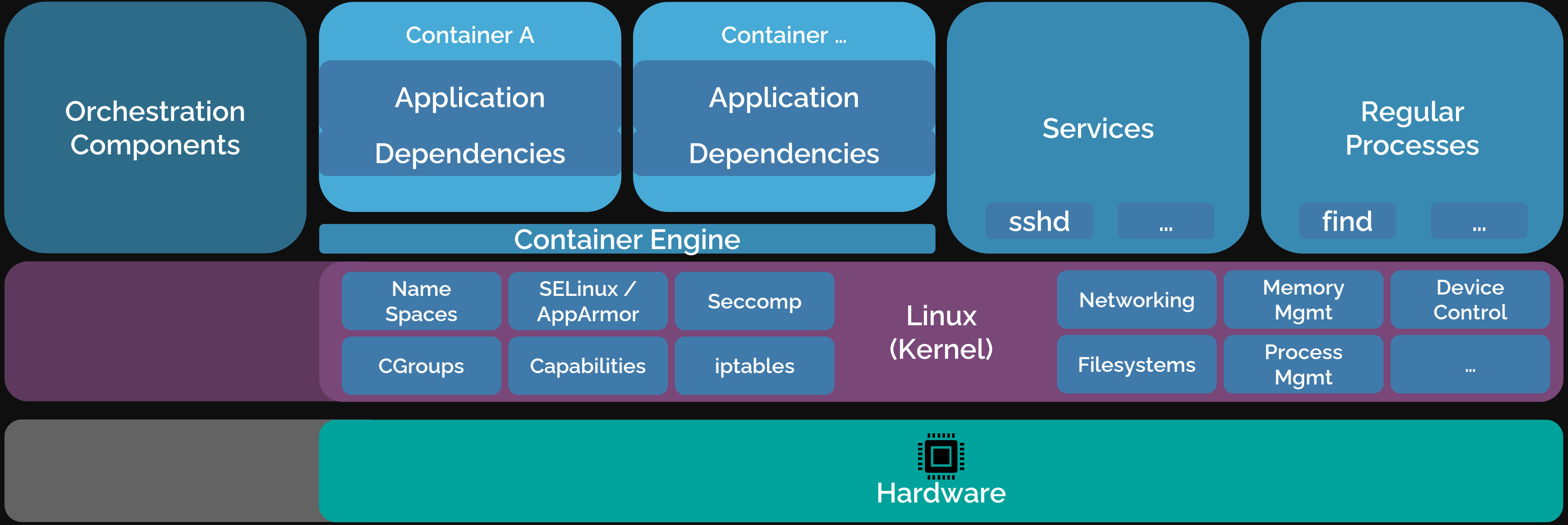


Notes

- **Containers Image** = Package containing *software, libraries and other dependencies* to run it.
- **Container** = Instantiated *container image.* Needs a *container engine + underlying OS*
- **Container Engine** often *aka Container Runtime = "Thing that starts the container"*
  - ***Engine**: Tooling around the runtime.* ***Runtime**: spawns the container*
- **Container Image Registry** = ~ *"App Store for Container Images"*
- Container images can be provided *locally* – or typically by using a ***container registry***

Container + Linux

Containers rely a lot on existing kernel infrastructure



- Containers require a *container runtime environment* to operate
- Containers use many (already existing) Linux kernel features
- Orchestration components are often added to the mix

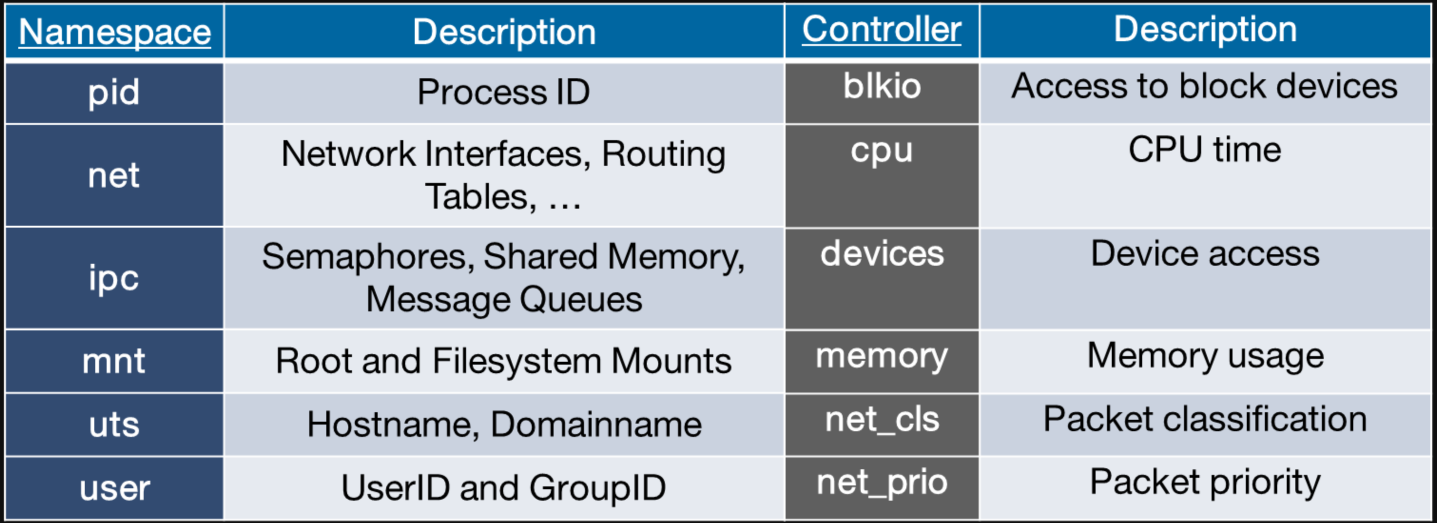
Container Base Technologies

Container = Namespaces + Control Groups

+ cgroup + time namespace

Both features of the Linux Kernel

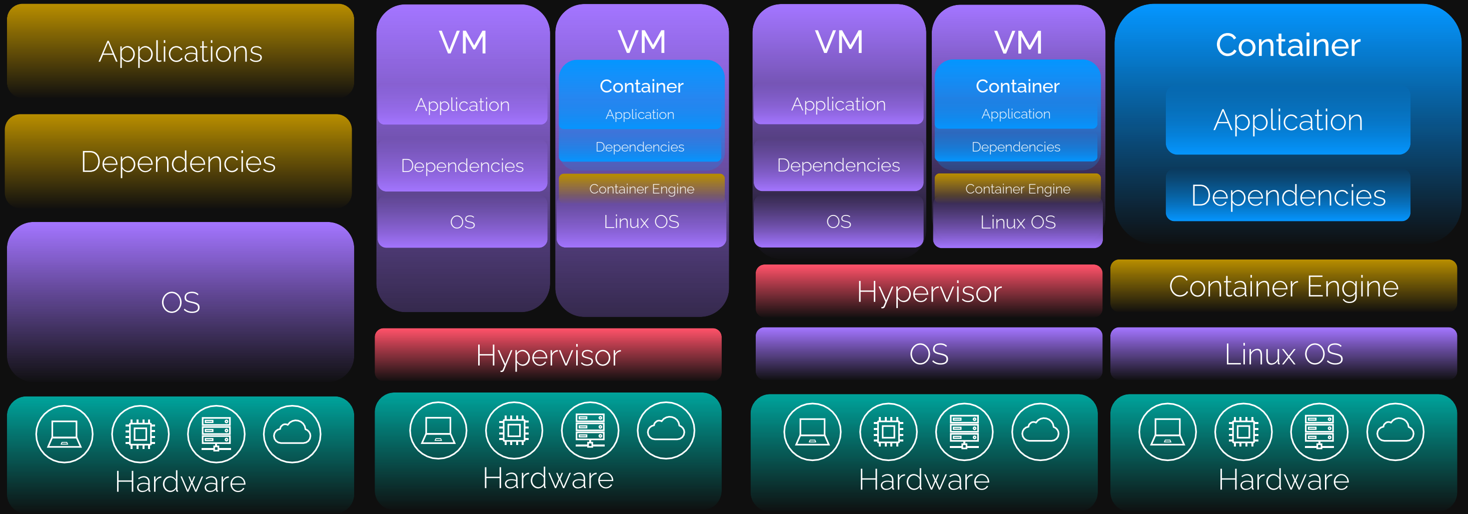
- Namespaces: Some sub systems ns-aware - isolated operation.
- Cgroups: Some resources limitable - prevention of excessive utilization



Namespace	Description	Controller	Description
pid	Process ID	blkio	Access to block devices
net	Network Interfaces, Routing Tables, …	cpu	CPU time
ipc	Semaphores, Shared Memory, Message Queues	devices	Device access
mnt	Root and Filesystem Mounts	memory	Memory usage
uts	Hostname, Domainname	net_cls	Packet classification
user	UserID and GroupID	net_prio	Packet priority

Virtualization with Containers

Stack: Bare Metal + Hypervisor + Containers



Bare Metal Operations

Type 1 hypervisor + OS with containers "Server"

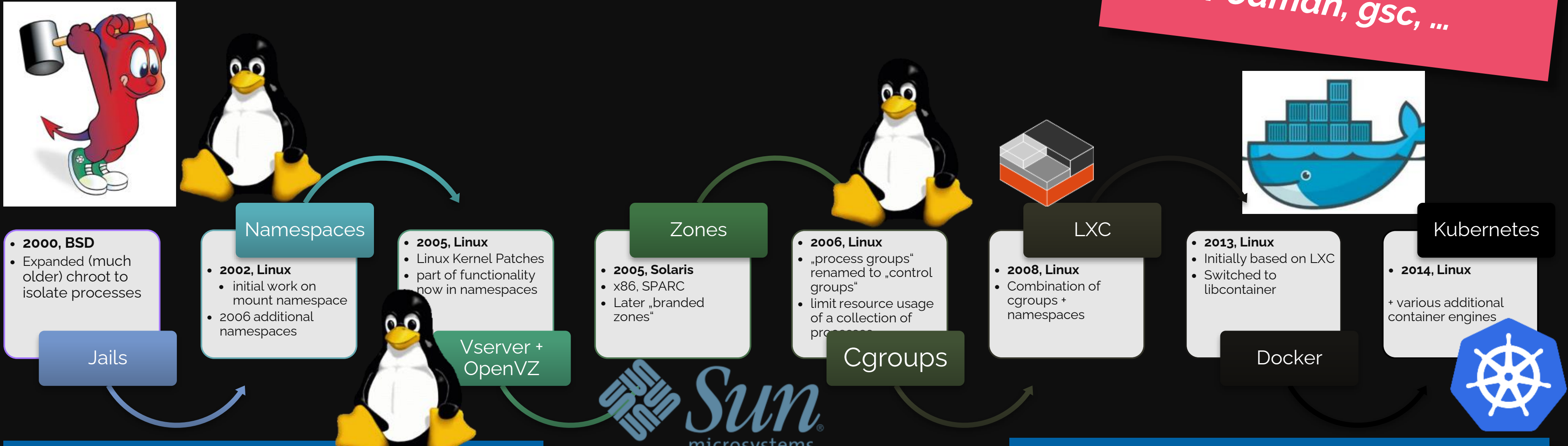
Type 2 hypervisor + OS with containers "Desktop""

Bare Metal OS + Containers

Evolution of OS-level Virtualization

Containers not just around since yesterday

+gVisor, SCONE, Kata, Podman, gsc, …



Jails
- **2000, BSD**
- Expanded (much older) chroot to isolate processes

Namespaces
- **2002, Linux**
  - initial work on mount namespace
- 2006 additional namespaces

Vserver + OpenVZ
- **2005, Linux**
- Linux Kernel Patches
- part of functionality now in namespaces

Zones
- **2005, Solaris**
- x86, SPARC
- Later „branded zones"

Cgroups
- **2006, Linux**
- „process groups" renamed to „control groups"
- limit resource usage of a collection of pro

LXC
- **2008, Linux**
- Combination of cgroups + namespaces

Docker
- **2013, Linux**
- Initially based on LXC
- Switched to libcontainer

Kubernetes
- **2014, Linux**

+ various additional container engines

HPC Container Runtimes

2015 Charliecloud (Jun)
2015 Shifter (Aug)
2016 Singularity 1.0 (Apr)
2016 udocker (Jul)
2019 Sarus 1.0 (Nov)

Hypervisor-based virtualization

1999 VMware Workstation 1.0
2001 ESX 1.0 & GSX 1.0
2003 Xen 1st public release
2006 KVM (2.6.10)

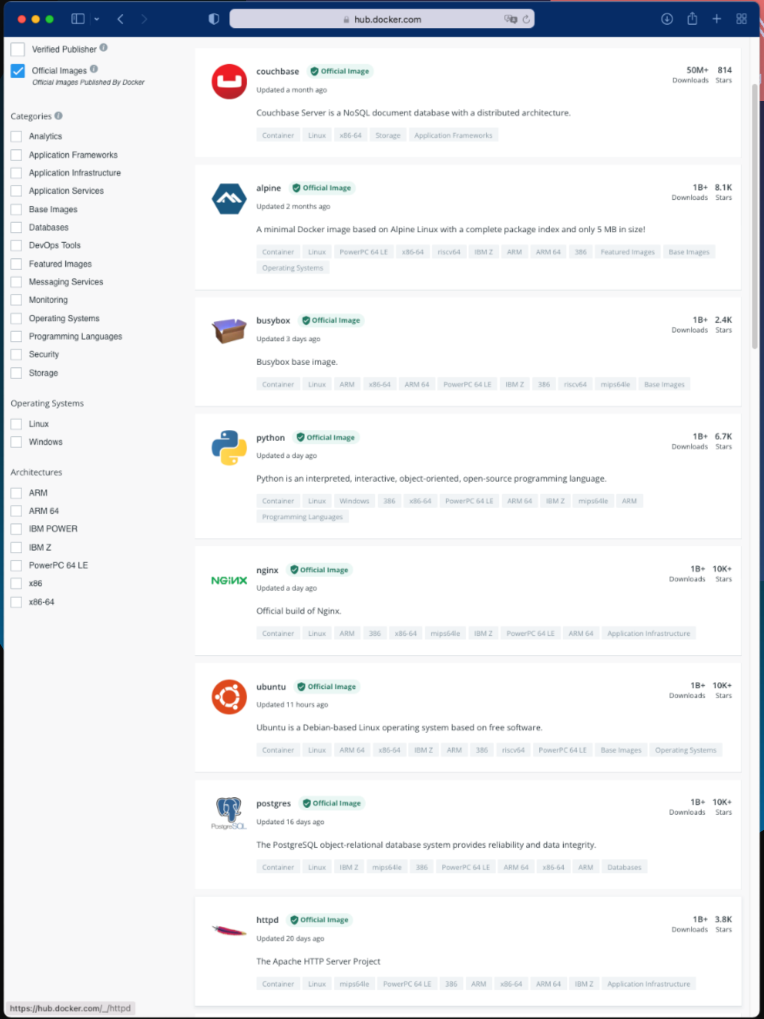
Container for Quick Solutions

From Zero to Hero

- Fast ramp-up of specialized applications
  - Databases, ...
  - Programming environments
  - Machine Learning Stack
  - IoT (sensor board example)
- Easy sharing
- No lengthy installation manuals
  - "Take my image and run this".
- Other environment: Ubuntu on RHEL Ubuntu, ...

GreenNose@HFU Sensor Board

Source: http://img.youtube.com/vi/KJRMjUzlHI8/0.jpg



☐ Verified Publisher
☑ Official Images
Official Images Published By Docker

Categories
☐ Analytics
☐ Application Frameworks
☐ Application Infrastructure
☐ Application Services
☐ Base Images
☐ Databases
☐ DevOps Tools
☐ Featured Images
☐ Messaging Services
☐ Monitoring
☐ Operating Systems
☐ Programming Languages
☐ Security
☐ Storage

Operating Systems
☐ Linux
☐ Windows

Architectures
☐ ARM
☐ ARM 64
☐ IBM POWER
☐ IBM Z
☐ PowerPC 64 LE
☐ x86
☐ x86-64

couchbase Official Image — 50M+ Downloads, 814 Stars
Updated a month ago
Couchbase Server is a NoSQL document database with a distributed architecture.
Container | Linux | x86-64 | Storage | Application Frameworks

alpine Official Image — 1B+ Downloads, 8.1K Stars
Updated 2 months ago
A minimal Docker image based on Alpine Linux with a complete package index and only 5 MB in size!
Container | Linux | PowerPC 64 LE | x86-64 | riscv64 | IBM Z | ARM | ARM 64 | 386 | Featured Images | Base Images | Operating Systems

busybox Official Image — 1B+ Downloads, 2.4K Stars
Updated 3 days ago
Busybox base image.
Container | Linux | ARM | x86-64 | ARM 64 | PowerPC 64 LE | IBM Z | 386 | riscv64 | mips64le | Base Images

python Official Image — 1B+ Downloads, 6.7K Stars
Updated a day ago
Python is an interpreted, interactive, object-oriented, open-source programming language.
Container | Linux | Windows | 386 | x86-64 | PowerPC 64 LE | ARM 64 | IBM Z | mips64le | ARM | Programming Languages

nginx Official Image — 1B+ Downloads, 10K+ Stars
Updated a day ago
Official build of Nginx.
Container | Linux | ARM | 386 | x86-64 | mips64le | IBM Z | PowerPC 64 LE | ARM 64 | Application Infrastructure

ubuntu Official Image — 1B+ Downloads, 10K+ Stars
Updated 11 hours ago
Ubuntu is a Debian-based Linux operating system based on free software.
Container | Linux | ARM 64 | x86-64 | IBM Z | ARM | 386 | riscv64 | PowerPC 64 LE | Base Images | Operating Systems

postgres Official Image — 1B+ Downloads, 10K+ Stars
Updated 16 days ago
The PostgreSQL object-relational database system provides reliability and data integrity.
Container | Linux | IBM Z | mips64le | 386 | PowerPC 64 LE | ARM 64 | x86-64 | ARM | Databases

httpd Official Image — 1B+ Downloads, 3.8K Stars
Updated 20 days ago
The Apache HTTP Server Project
Container | Linux | mips64le | PowerPC 64 LE | 386 | ARM | x86-64 | ARM 64 | IBM Z | Application Infrastructure

https://hub.docker.com/_/httpd

Source:
https://hub.docker.com/search?type=image&image_filter=official&operating_system=linux&architecture=amd64

Containers in Everyday IT

Hybrid Cloud, DevOps, HPC, …

Hybrid Cloud

- Simplified application deployment, scaling

DevOps - bringing developers (Dev) and operations team (Ops) together

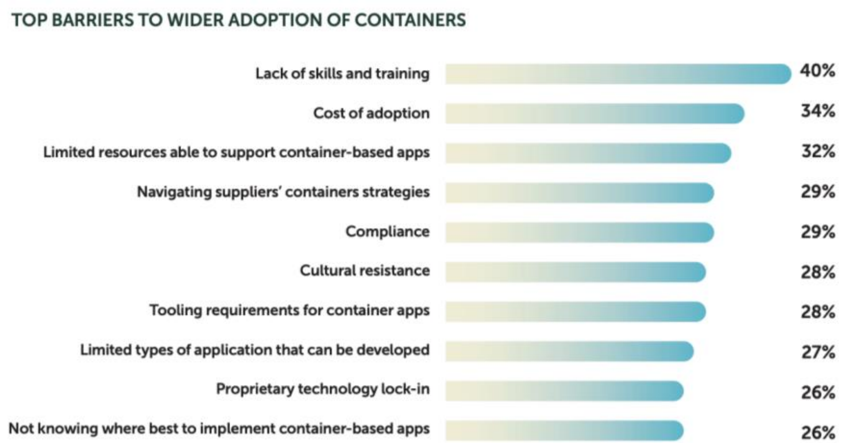
- Containers are great for streamlining DevOps workflows
  - Ops roll out applications that Dev has developed - doesn't always have to work out… - "Works on my machine"
  - Easier with containers - as application already has dependencies with in package
- Creation of images usually combined with CI/CD pipelines (Jenkins, …)
  - Automatic creation of images that are uploaded to a registry
- Consistency for development, test and production environments.
  - Everyone works with the same consistent container environment
- Simplify updates
  - Rollbacks to working state by redeploying an earlier working version

HPC - Containers are also gaining adoption in HPC, motivation:

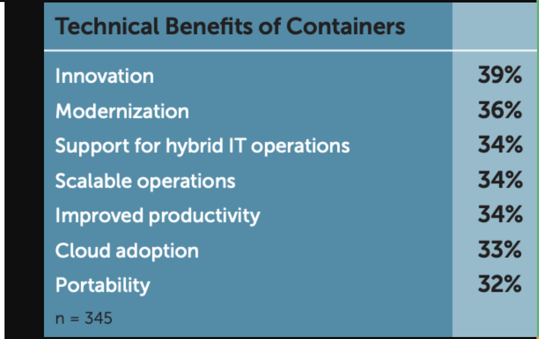
"portability" (grid-like environment), user-supplied software stack, low overhead, …

Further Reading

Container and Kubernetes, Market Dynamics Report, 2021

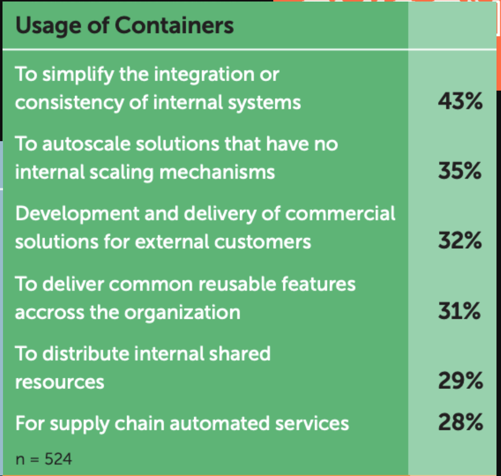


n = 574



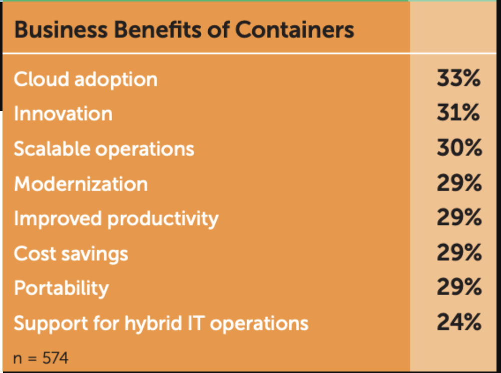
Technical Benefits of Containers	
Innovation	39%
Modernization	36%
Support for hybrid IT operations	34%
Scalable operations	34%
Improved productivity	34%
Cloud adoption	33%
Portability	32%

n = 345



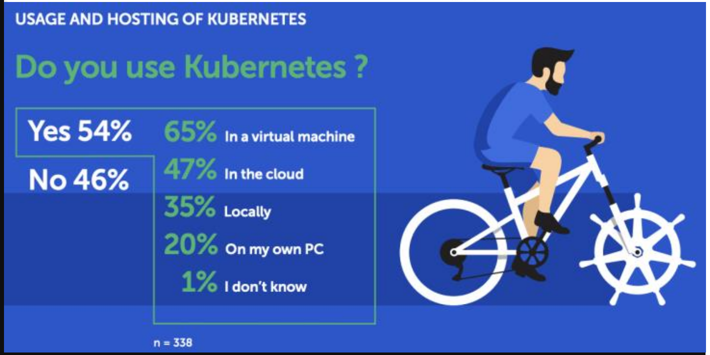
Usage of Containers	
To simplify the integration or consistency of internal systems	43%
To autoscale solutions that have no internal scaling mechanisms	35%
Development and delivery of commercial solutions for external customers	32%
To deliver common reusable features accross the organization	31%
To distribute internal shared resources	29%
For supply chain automated services	28%

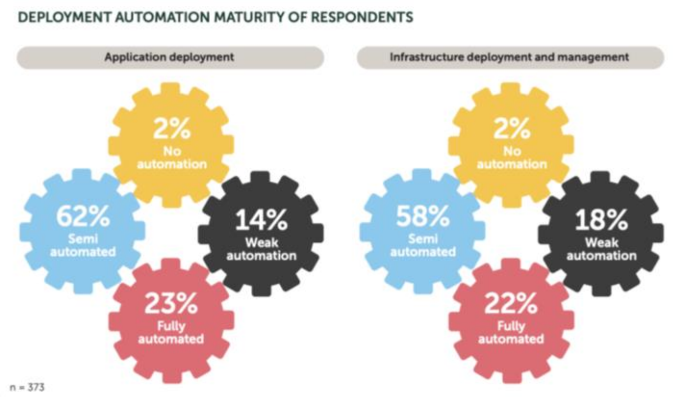
n = 524



Business Benefits of Containers	
Cloud adoption	33%
Innovation	31%
Scalable operations	30%
Modernization	29%
Improved productivity	29%
Cost savings	29%
Portability	29%
Support for hybrid IT operations	24%

n = 574





Sources: https://www.redhat.com/rhdc/managed-files/cl-containers-kubertnetes-market-dynamics-f29518-202106-en.pdf

Why scientists + the HPC community love Containers

And so should you!

Mobility / Portability

- Compute resources are flexible
  - Laptop
  - Workstation
  - HPC
  - Cloud
- Encapsulated SW stack

User-supplied applications

- Growing demand
- Helps with contradictory requirements
- “Works on my machine”
- Novel applications
  - Latest Ubuntu vs Enterprise Linux
- Legacy code
  - Fortran @CentOS5

Reproducibility

- Collaboration
- Passing on the SW environment
- Simplifies reproducibility
- Defined SW stack in container („immutability“)
- Standardization

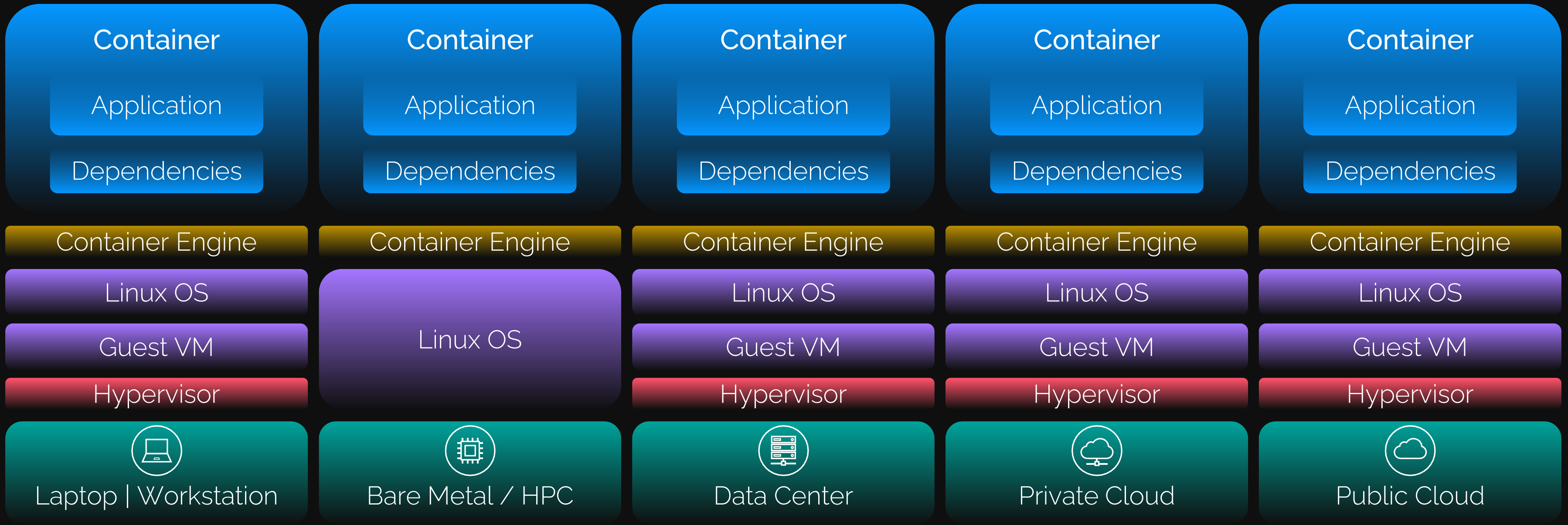
Performance

- Very low overhead
- Performance close to bare metal
- Lots of research on the subject, including our own

Container Technology

The promise of portability

- "Containers enable portability across platforms"
- Singularity talks about "Mobility of Compute"



Portability

Why is it harder in HPC?

- Portability depends on the application
- Results from host/environment dependencies

	Tensorflow	Tensorflow with GPU	Solver with GPU Support	Typical CAE Solver
Node	Single	Single	Single	Multi
Storage / Scratch	Volume	Volume	Shared	Shared
Devices required	-	GPU	GPU	Infiniband
Other				MPI & Slurm
Runtime	Enterprise	NVIDIA-docker	Enterprise/HPC	HPC

Dependencies and required effort

Container Engines

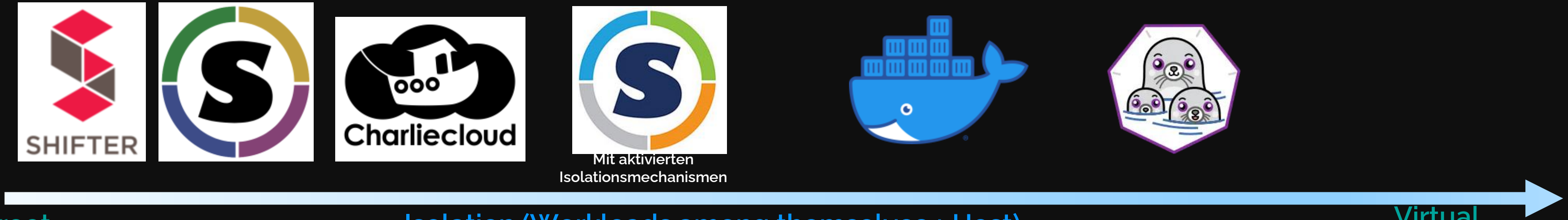
Not all containers are created equal

Even though everything is called a container, the degree of isolation varies

- Depending on origin (enterprise/HPC container engine) different degree of isolation mechanisms:
  "As much isolation as possible vs as little as necessary."
  - *chroot vs namespaces, Cgroups, Seccomp, AppArmor/SELinux, ...*
  - *Background: n containers / host vs 1 container / host*

- Isolation can often be activated for HPC engines or deactivated for enterprise engines.

- Approximate classification of the isolation on the basis of the available features*:



chroot

Isolation (Workloads among themselves + Host)
*(*simplified representation)*

Virtual
Machine

Container Engine Diversity

Quick Summary

Docker (2013)

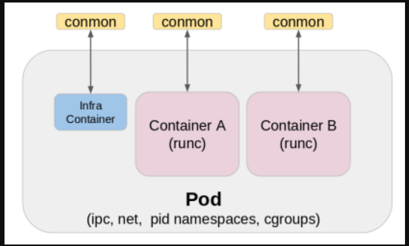
- Without a chance in HPC, fading in importance:
  - Dockershim in Kubernetes (aka "K8s drops Docker").
  - Sale of Enterprise to Mirantis, focus on Hub + Desktop

Singularity (2015)

- The most popular and widespread HPC container engine
- Additional own flat image format (SIF) - potentially more performant
- Singularity (Sylabs) vs Apptainer (CIQ / Linux Foundation)
- 11/22: Will encapsulate OCI images in SIF in the future

Podman (2017)

- Support by Red Hat
- Migration: "alias docker=podman" and "podman-compose"
- Rootless container, buildah for building images
- Supports pod concept -> simplified way to K8s from local tests
- 07/2022: Support SIF Images
- 11/2022 NERSC's podman-hpc



Container Engine Diversity

The lesser known HPC Container Engines

Shifter (2015)

- Docker images on Cray, focus on image workflow.
- *"Liked the container concept, but not the technology"*
- Maintenance only (due to NERSC's podman strategy)

Charliecloud (2015)"

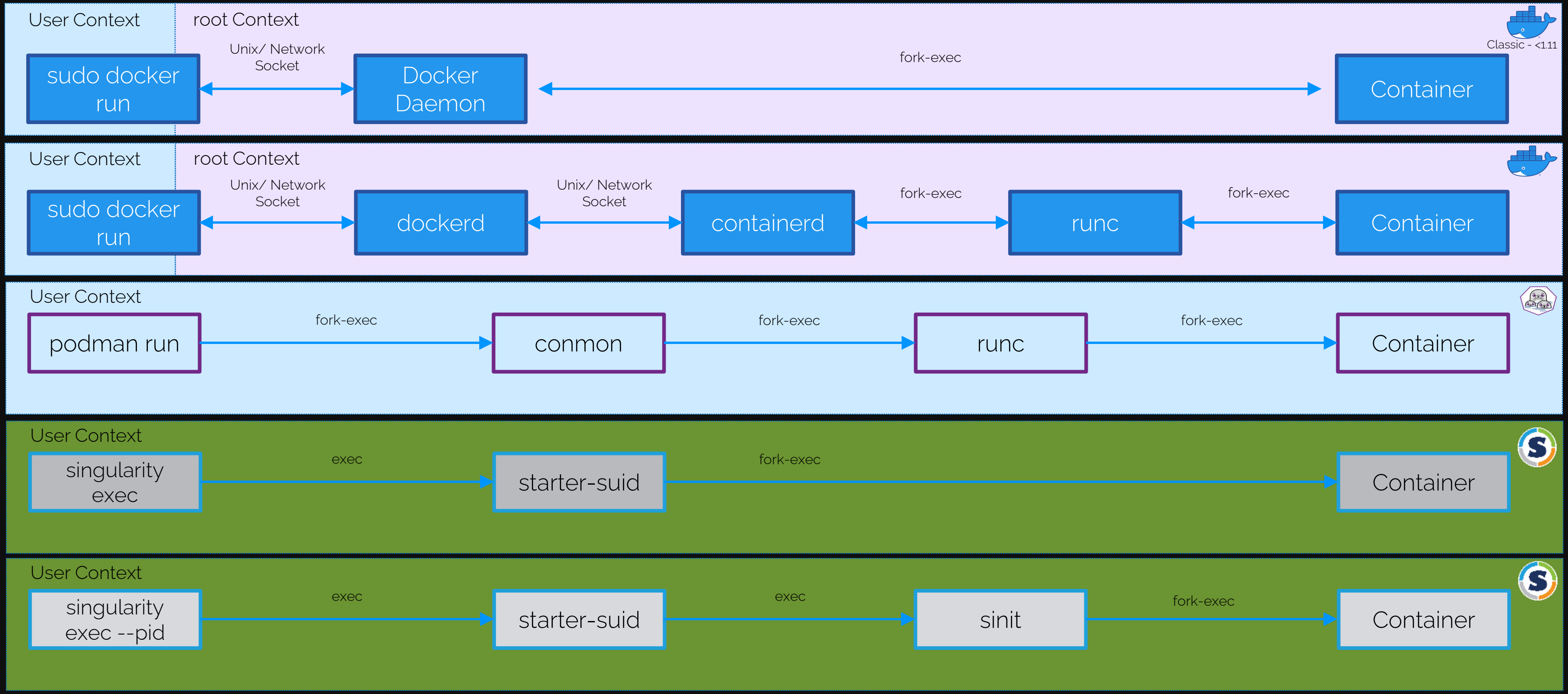
- Use of user namespaces. Compact (800 LOCs)
- Usable as Uboot (LRZ), but rough - own commands, image conversion

Sarus (2019)

- Developed at CSCS – focus: full OCI Support
- Providing of useful features via OCI Hooks (NVIDIA Container Toolkit, Native MPI, Native glibc, SSH, Slurm global sync hook,…)

Container Engines

From container engine call to running Container



Container Technologies

Standards

- Despite Docker's fading importance:
  - Co-created Open Container Initiative (OCI) *specifications image, runtime, distribution* will outlast

- All engines support *OCI images* - switching engines is relatively easy

- Many engines now rely on *runc* as runtime
  - Docker, Podman, Sarus
  - *runc* is the reference implementation of the OCI runtime-specification
  - Availability of *crun* as a replacement (reasons: Go vs C implementation, cgroups v2, performance)
  - Hooks support to extend core functionality defined in runtime spec

EVIDEN
an atos business
science + computing

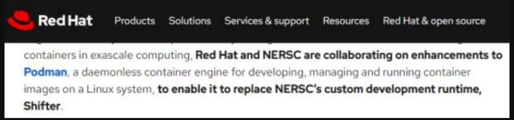
The Agony of Choice

Singularity vs. Podman vs. Others

- **Singularity is appropriate when**
  - The containerized workload is to be integrated into the environment as transparently as possible, including HPC-specific elements.
  - SIF image should not be stored in registry but in a share, directly executable, no "import"

- **Podman** is advisable when
  - The integration effort plays a subordinate role
  - Rootless limitations (work in progress) do not come into play
  - *Stronger isolation* is desired and feasible
  - A homogeneous setup for different workloads is desired + *single vendor strategy* if applicable

- Further Findings
  - Docker unfit for HPC
  - Sarus is still interesting, but low adoption and small developer group
  - Performance differences negligible
  - Future belongs to Podman – see also

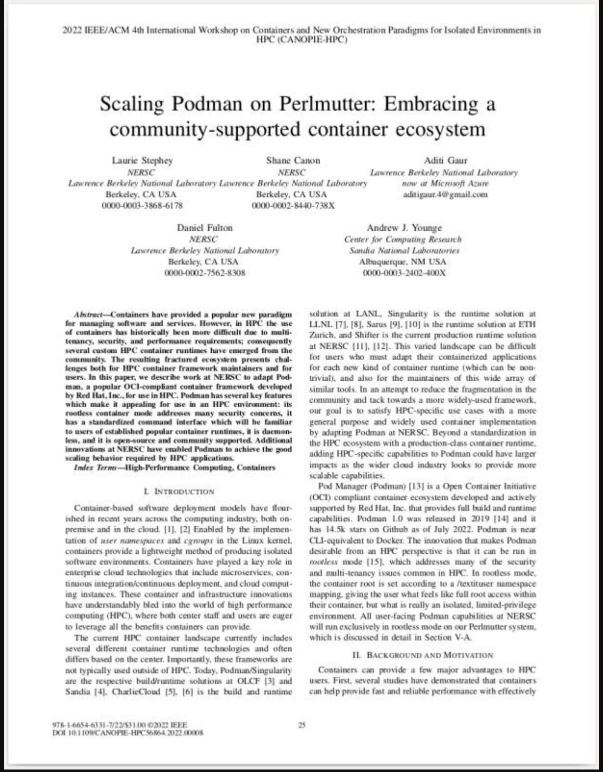


containers in exascale computing, **Red Hat and NERSC are collaborating on enhancements to Podman**, a daemonless container engine for developing, managing and running container images on a Linux system, **to enable it to replace NERSC's custom development runtime, Shifter**.

Source: https://www.redhat.com/en/about/press-releases/red-hat-joins-forces-us-department-energy-laboratories-bridge-gap-between-high-performance-computing-and-cloud-environments

Scaling Podman on Perlmutter

SC22 CANOPIE Container Workshop



2022 IEEE/ACM 4th International Workshop on Containers and New Orchestration Paradigms for Isolated Environments in HPC (CANOPIE-HPC)

Scaling Podman on Perlmutter: Embracing a community-supported container ecosystem

Laurie Stephey
NERSC
Lawrence Berkeley National Laboratory
Berkeley, CA USA
0000-0003-3868-6178

Shane Canon
NERSC
Lawrence Berkeley National Laboratory
Berkeley, CA USA
0000-0002-8440-738X

Aditi Gaur
Lawrence Berkeley National Laboratory
now at Microsoft Azure
aditigaur.4@gmail.com

Daniel Fulton
NERSC
Lawrence Berkeley National Laboratory
Berkeley, CA USA
0000-0002-7562-8308

Andrew J. Younge
Center for Computing Research
Sandia National Laboratories
Albuquerque, NM USA
0000-0003-2402-400X

Abstract—**Containers have provided a popular new paradigm for managing software and services. However, in HPC the use of containers has historically been more difficult due to multi-tenancy, security, and performance requirements; consequently several custom HPC container runtimes have emerged from the community. The resulting fractured ecosystem presents challenges both for HPC container framework maintainers and for users. In this paper, we describe work at NERSC to adapt Podman, a popular OCI-compliant container framework developed by Red Hat, Inc., for use in HPC. Podman has several key features which make it appealing for use in an HPC environment: its rootless container mode addresses many security concerns, it has a standardized command interface which will be familiar to users of established popular container runtimes, it is daemonless, and it is open-source and community supported. Additional innovations at NERSC have enabled Podman to achieve the good scaling behavior required by HPC applications.**

Index Terms—**High-Performance Computing, Containers**

I. Introduction

Container-based software deployment models have flourished in recent years across the computing industry, both on-premise and in the cloud. [1], [2] Enabled by the implementation of *user namespaces* and *cgroups* in the Linux kernel, containers provide a lightweight method of producing isolated software environments. Containers have played a key role in enterprise cloud technologies that include microservices, continuous integration/continuous deployment, and cloud computing instances. These container and infrastructure innovations have understandably bled into the world of high performance computing (HPC), where both center staff and users are eager to leverage all the benefits containers can provide.

The current HPC container landscape currently includes several different container runtime technologies and often differs based on the center. Importantly, these frameworks are not typically used outside of HPC. Today, Podman/Singularity are the respective build/runtime solutions at OLCF [3] and Sandia [4], CharlieCloud [5], [6] is the build and runtime solution at LANL, Singularity is the runtime solution at LLNL [7], [8], Sarus [9], [10] is the runtime solution at ETH Zurich, and Shifter is the current production runtime solution at NERSC [11], [12]. This varied landscape can be difficult for users who must adapt their containerized applications for each new kind of container runtime (which can be non-trivial), and also for the maintainers of this wide array of similar tools. In an attempt to reduce the fragmentation in the community and tack towards a more widely-used framework, our goal is to satisfy HPC-specific use cases with a more general purpose and widely used container implementation by adapting Podman at NERSC. Beyond a standardization in the HPC ecosystem with a production-class container runtime, adding HPC-specific capabilities to Podman could have larger impacts as the wider cloud industry looks to provide more scalable capabilities.

Pod Manager (Podman) [13] is a Open Container Initiative (OCI) compliant container ecosystem developed and actively supported by Red Hat, Inc. that provides full build and runtime capabilities. Podman 1.0 was released in 2019 [14] and it has 14.5k stars on Github as of July 2022. Podman is near CLI-equivalent to Docker. The innovation that makes Podman desirable from an HPC perspective is that it can be run in *rootless* mode [15], which addresses many of the security and multi-tenancy issues common in HPC. In rootless mode, the container root is set according to a /textituser namespace mapping, giving the user what feels like full root access within their container, but what is really an isolated, limited-privilege environment. All user-facing Podman capabilities at NERSC will run exclusively in rootless mode on our Perlmutter system, which is discussed in detail in Section V-A.

II. Background and Motivation

Containers can provide a few major advantages to HPC users. First, several studies have demonstrated that containers can help provide fast and reliable performance with effectively

978-1-6654-6331-7/22/$31.00 ©2022 IEEE
DOI 10.1109/CANOPIE-HPC56864.2022.00008

25

NERSC

- Very large scale HPC site (Perlmutter #5/#8 @Top500)
- uses Shifter as production runtime
- Goal: satisfy HPC-specific use cases with a more general purpose and widely used container implementation by adapting Podman
- Added HPC-specific capabilities in collab. with Red Hat

Why Podman?

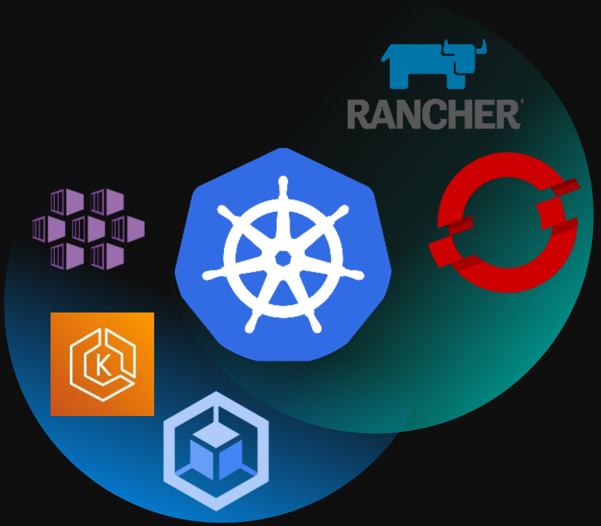
- large community, build capabilities with same tooling, fuse-overlay mount for writeable images, SIF image support, rootless containers, network isolation, OCI compliant

-> Growing interest in Podman in the HPC community

Kubernetes

Kubernetes (K8s)

If one says containers, he typically means Kubernetes



State-of-the-art Open-source system for automating deployment, scaling, and management of containerized applications - aka "Container Orchestration"

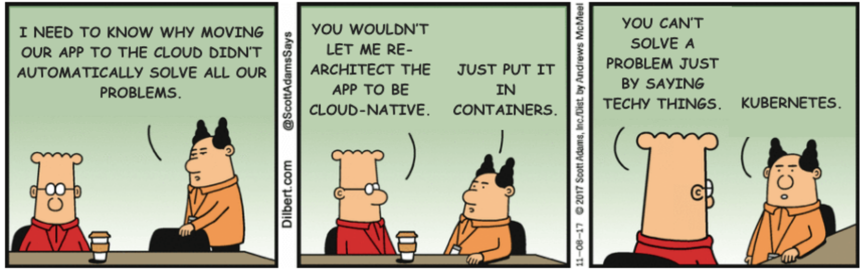
Commercial representatives

- Red Hat OpenShift, SUSE Rancher, …

Also available at the Hyperscaler of your choice

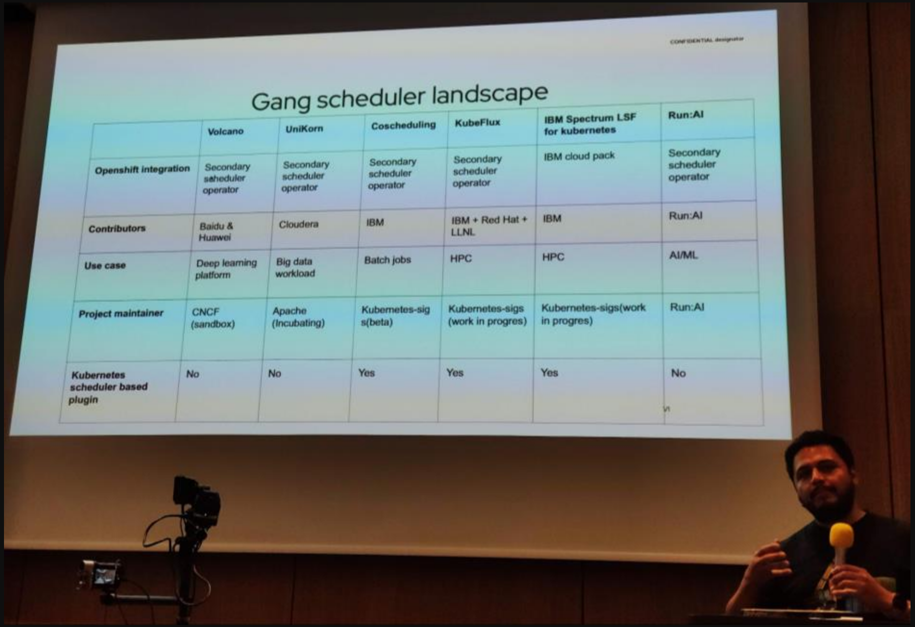
- AKS@Azure, EKS@AWS, GKE@GCP

However…


I NEED TO KNOW WHY MOVING OUR APP TO THE CLOUD DIDN'T AUTOMATICALLY SOLVE ALL OUR PROBLEMS.
@ScottAdamsSays
Dilbert.com
YOU WOULDN'T LET ME RE-ARCHITECT THE APP TO BE CLOUD-NATIVE.
JUST PUT IT IN CONTAINERS.
11-08-17 © 2017 Scott Adams, Inc./Dist. by Andrews McMeel
YOU CAN'T SOLVE A PROBLEM JUST BY SAYING TECHY THINGS.
KUBERNETES.

Kubernetes and HPC

Much harder than it sounds



CONFIDENTIAL designator

Gang scheduler landscape

	Volcano	UniKorn	Coscheduling	KubeFlux	IBM Spectrum LSF for kubernetes	Run:AI
Openshift integration	Secondary scheduler operator	Secondary scheduler operator	Secondary scheduler operator	Secondary scheduler operator	IBM cloud pack	Secondary scheduler operator
Contributors	Baidu & Huawei	Cloudera	IBM	IBM + Red Hat + LLNL	IBM	Run:AI
Use case	Deep learning platform	Big data workload	Batch jobs	HPC	HPC	AI/ML
Project maintainer	CNCF (sandbox)	Apache (Incubating)	Kubernetes-sigs(beta)	Kubernetes-sigs (work in progres)	Kubernetes-sigs(work in progres)	Run:AI
Kubernetes scheduler based plugin	No	No	Yes	Yes	Yes	No

Source: ISC 2022 – High Performance Container Workshop – Kubernetes + HPC - Eduardo Arango

Kubernetes was primarily designed with cloud native scalable services in mind

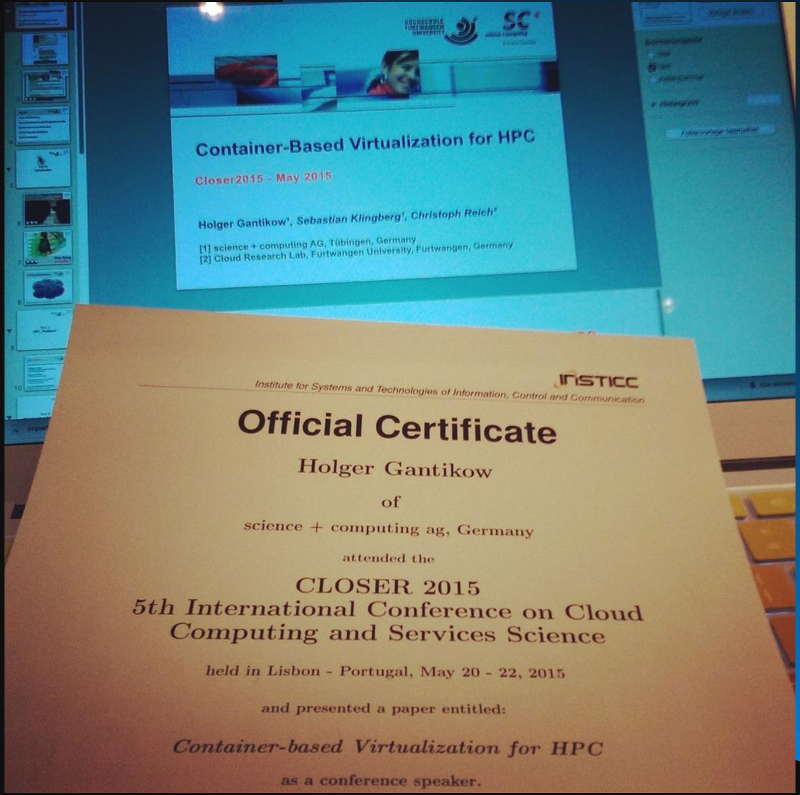
- Typical HPC applications behave differently
  - No dynamic auto-scaling
  - Use of MPI
  - Requirements for HPC interconnects (Infiniband, …)
  - …
- Ongoing efforts to make K8s *HPC-aware*
  - Variety of gang schedulers for HPC/Batch/AI+ML
  - MPI Operator
- Nimbix invested lots of effort to bring the two worlds together

04. Use Cases in HPC

Aka *We and Containers*

Containers?

Not a new topic for us



Container-Based Virtualization for HPC

Closer2015 - May 2015

Holger Gantikow[1], *Sebastian Klingberg*[1], *Christoph Reich*[2]

[1] science + computing AG, Tübingen, Germany
[2] Cloud Research Lab, Furtwangen University, Furtwangen, Germany

INSTICC

Institute for Systems and Technologies of Information, Control and Communication

Official Certificate

Holger Gantikow

of

science + computing ag, Germany

attended the

CLOSER 2015
5th International Conference on Cloud Computing and Services Science

held in Lisbon - Portugal, May 20 - 22, 2015

and presented a paper entitled:

Container-based Virtualization for HPC

as a conference speaker.

- Publications : 7
- Final Theses: 7
- Interns: 4
- Focus: Security, HPC

Rootless containers with Podman for HPC (2020)

Based on *elapsed time to normal termination* reported by LS-DYNA

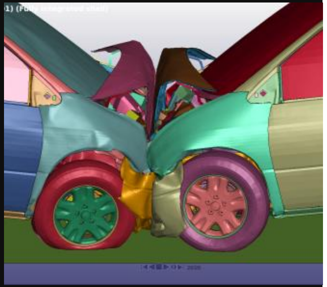
Holger Gantikow, Steffen Walter, and Christoph Reich

Rootless Containers with Podman for HPC

15th Workshop on Virtualization in High-Performance Cloud Computing (VHPC'20), The Internet, June 2020

All container runtimes introduce a certain amount of overhead (1,85% - 5,10%)

- Overhead **might be negligible** over benefits
  - 2/3 runtimes add <= 2,04% overhead
- Singularity causes **smallest overhead**
- Diff. Singularity vs Podman-crun **minimal**
  - 1,61%-1,62% for long runs
- Performance of **runc is noticeably (s)lower** compared to crun / Singularity



Gantikow, H., Walter, S., & Reich, C. (2020). Rootless Containers with Podman for HPC. In High Performance Computing (pp. 343–354). Springer International Publishing. http://dx.doi.org/10.1007/978-3-030-59851-8_23

Rootless Containers with Podman for HPC

Holger Gantikow[1,2](✉), Steffen Walter[1], and Christoph Reich[2]

[1] science+computing AG, Atos, Tübingen, Germany
gantikow@gmail.com, steffen.walter@atos.net
[2] Institute for Cloud Computing and IT Security, Furtwangen University, Furtwangen, Germany
{holger.gantikow,christoph.reich}@hs-furtwangen.de

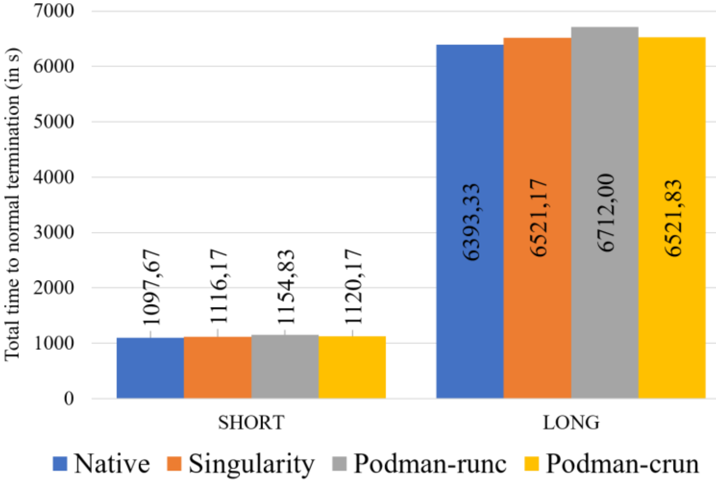
Abstract. Containers have become popular in HPC environments to improve the mobility of applications and the delivery of user-supplied code. In this paper we evaluate Podman, an enterprise container engine that supports rootless containers, in combination with runc and crun as container runtimes using a real-world workload with LS-DYNA, and the industry-standard benchmarks sysbench and STREAM. The results suggest that Podman with crun only introduces a similar low overhead as HPC-focused container technologies.

Keywords: Virtualization · Container · Podman · Singularity · runc · crun · Rootless containers · LS-DYNA · Benchmark · Performance analysis

1 Introduction

Over the last half decade, containers have become a valuable asset when it comes to running services and applications. Especially in enterprise environments, where containers have turned into an integral part of the development, deployment and operation of modern microservice architectures, they are well received. They also prove to be particularly suitable for HPC environments, where they improve the mobility of applications and the delivery of user-supplied code. HPC centers are increasingly confronted with these requirements, since users there are not only demanding software for traditional MPI-based simulations but also increasingly novel software stacks and workflows to support workloads from the data science domain. In addition, modern sites call for solutions that open up ways for simplified use of resources distributed across multiple locations and for obtaining supplementary resources from the cloud [18] at a progressive rate. Containers are a suitable component to support the demands for flexibility in a in many ways converging future of compute. They decouple applications and their dependencies from the underlying operating system and encapsulate them in an easily redistributable unit. Due to special requirements of HPC environments in the form of HPC-specific hardware and related libraries [2]

© Springer Nature Switzerland AG 2020
H. Jagode et al. (Eds.): ISC High Performance 2020 Workshops, LNCS 12321, pp. 343–354, 2020.
https://doi.org/10.1007/978-3-030-59851-8_23



	Native	Singularity	Podman-runc	Podman-crun
Short Run	1097,67 s	1116,17 s	1154,83 s	1120,17 s
Long Run	6393,33 s	**6521,17 s**	6712,00 s	**6521,83 s**
Overhead Short Run	-	*2,09%*	*5,36%*	*2,45%*
Overhead Long Run	-	*1,61%*	*4,58%*	*1,62%*
Mean Overhead	**-**	**1,85%**	**5,10%**	**2,04%**

Where are we coming from?

Developments for Enterprise HPC Users (focus on CAE/Automotive)*

- 2021 Summary:
  - Started to listen, interested, but container adoption rate far behind research sites
    - Mostly due to UDSS being no use-case – ops installs all the software…
  - Containers mostly seen with specific offerings (Nimbix, UberCloud, …)
- 2022 Summary:
  - Growing interest, recognized potential benefits of containers
  - Drivers: Multi-site HPC Grids, co-usage of K8s-based resources, cloud
    - + additional non-traditional workloads, specific interests
  - Growing interest in Podman for new installations (single vendor strategy?)
    - Singularity (at least for new installations) declining (Singularity vs Apptainer did not help)
- Since then:
  - Extended PoCs
  - Starting to think beyond "just put this in a container"
    - Lifecycle topics, run complex(er) workflows + SBOMs, security benetfits

science + computing

Burak Yenier

Holger Gantikow

High Performance Computing with Containers

Software Vendor Supported Images

HPCW21, June 2021

Trusted partner for your Digital Journey

science + computing

Software Vendor Supported Images

2022 Edition

Holger Gantikow
Burak Yenier
High Performance Container Workshop
ISC2022

© Atos - for internal use

science + computing

State of Container Adoption at Enterprise HPC Users*

It's a long way to the top

Holger Gantikow
CAE IT Landscape Architect

High Performance Container Workshop
ISC2023

25.04.2023

* slightly CAE-biased

For more details see ISC 2021/2022/2023 talks

(*) Traditional clusters with Slurm (or similar), no K8s, no AI/ML workloads

Reasons for low adoption for ISVs

And what to do about it?

science + computing

Software Vendor Supported Images
2022 Edition

Holger Gantikow
Burak Yenier
High Performance Container Workshop
ISC2022

© Atos - For internal use

- Lack of financial incentive
  - Customer needs our SW, will not pay more if containerized
- Don’t fix it, if it ain’t broken (enough)
  - Working software distribution model in place
  - Issues they had to solve predate containers – found “ways”
- Don’t have a common specification to satisfy all customers
  - Environment dependencies, workflows, …
- Increased effort + responsibilities, update frequency
  - As base layers would also be part of the shipped package

Recommendation to get started:

- Pick one workload / container image to start with
- Use Dockerfile to define the image, test various runtimes based on needs
- Solve for not one time build but also for long term maintenance
- Move to the next workload, at each step introduce one additional complexity
- Share your experience with your software vendor and others in the community

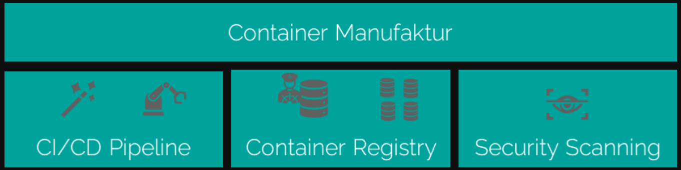
Container Infrastructure

POC in Multi-Site HPC Environment

Holger Gantikow
CAE IT Landscape Architect

POC Container Infrastructure

POC Automotive Customer



Container Manufaktur

CI/CD Pipeline

Container Registry

Security Scanning

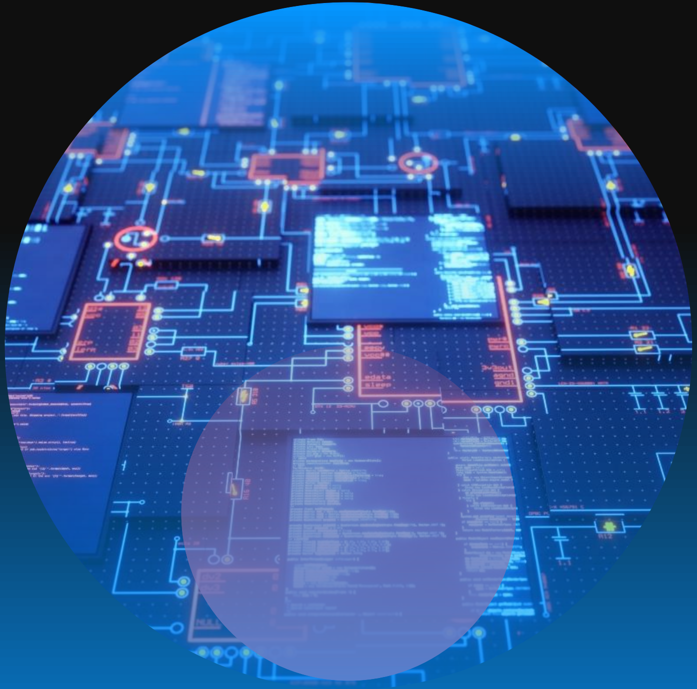
- $CUSTOMER uses a geographically distributed HPC environment with diverse resources.
  - Platforms: Onprem HPC, onprem "cloud native" environment, DataCenter in Scandinavia, Hyperscaler.

Focus of the project

- **Deployment of Container Engine** on different platforms (Singularity)
- Infrastructure for **building, deploying and maintaining images** for the different platforms
  - Jenkins CI/CD, Harbor Registry - **automated build + replication** to different locations
- **Improve application deployment** in the different environments
  - Image push vs SW share rsync (direct transfer image to Scandinavia 8min vs share sync overnight)
- Provision of images for typical CAE workloads for the different target environments
  - LS-Dyna, Abaqus, StarCCM, specific coupled application
- Enabling Bring your own Environment (BYOE) for power users (future outlook)
  - Encapsulation of self-developed SW / tools / workflows
- Starting points for future initiatives: Security scanning, software bill of material (SBOM)

Validation & Certification

Group-wide CAE Images



Holger Gantikow
CAE IT Landscape Architect

Validation + Certification Group-wide CAE Images

Supported by us

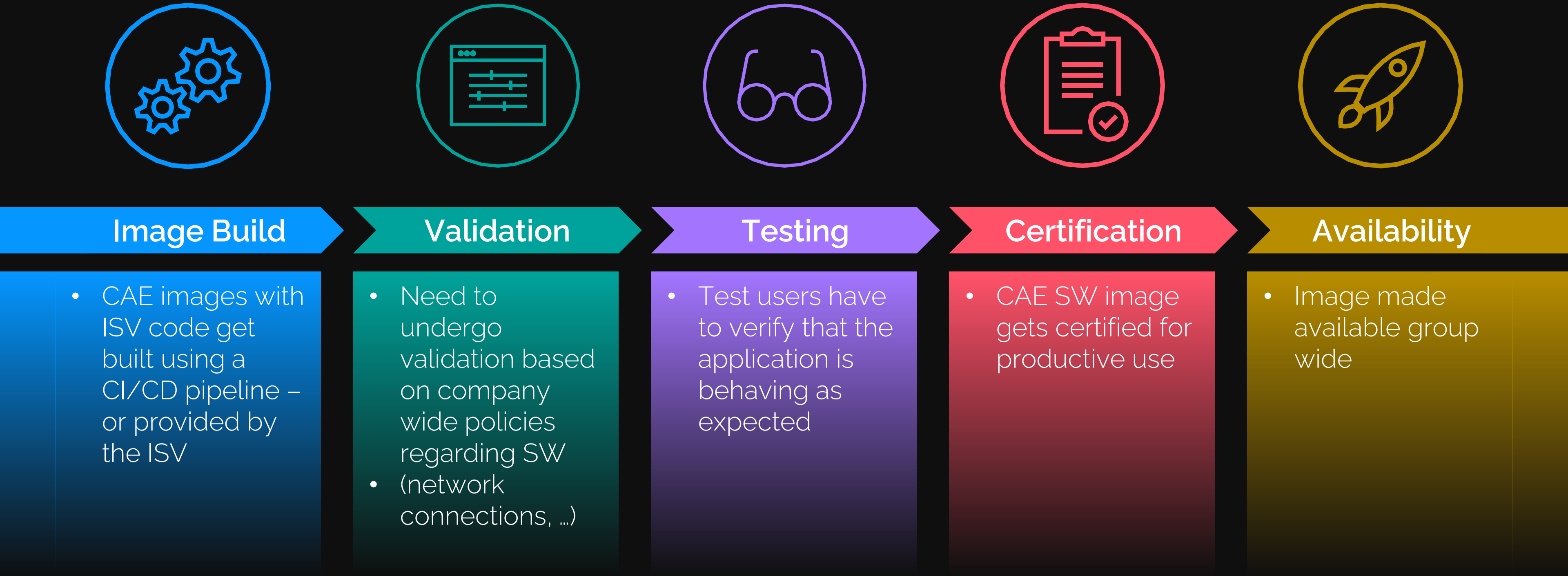


Image Build

- CAE images with ISV code get built using a CI/CD pipeline – or provided by the ISV

Validation

- Need to undergo validation based on company wide policies regarding SW
- (network connections, …)

Testing

- Test users have to verify that the application is behaving as expected

Certification

- CAE SW image gets certified for productive use

Availability

- Image made available group wide

Security Workshop

Holger Gantikow

Security in HPC with Containers

Online, December 2021

Trusted partner for your **Digital Journey**

Software Bill of Material

Aka “What is running on my cluster?”

Hard to keep track of software used on a large-scale system

- Lots of different applications, with numberless dependencies
- Especially hard when SW is provided beyond `rpm/apt/apk` (pip, jars, go modules, …)

Hard to answer questions like

- What software is outdated / has vulnerabilities?
- What software relies on a specific buggy library version that impacts the results?

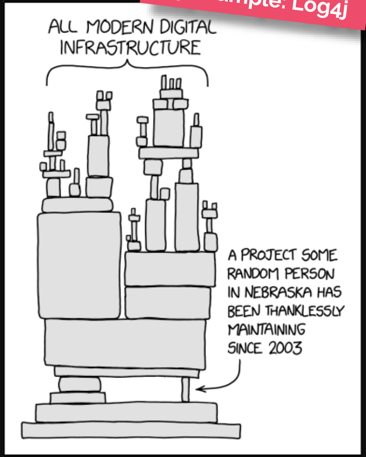
Gets much easier when relying on containers as sole source of software

- Software used = Host Software + Container image content

OSS software solutions to support this (examples later)

- Various package formats / SW sources, details like Maintainers, Licences, Checksums of files, …
- Should be integrated with image release process / registry ("Container App Store")

By example: Log4j


ALL MODERN DIGITAL INFRASTRUCTURE
A PROJECT SOME RANDOM PERSON IN NEBRASKA HAS BEEN THANKLESSLY MAINTAINING SINCE 2003

Source: https://xkcd.com/2347/

Software Solution – vulnerability static analysis

Standalone, Integrated in Image Registry

Rep: Clair, Syft / Grype, Trivy

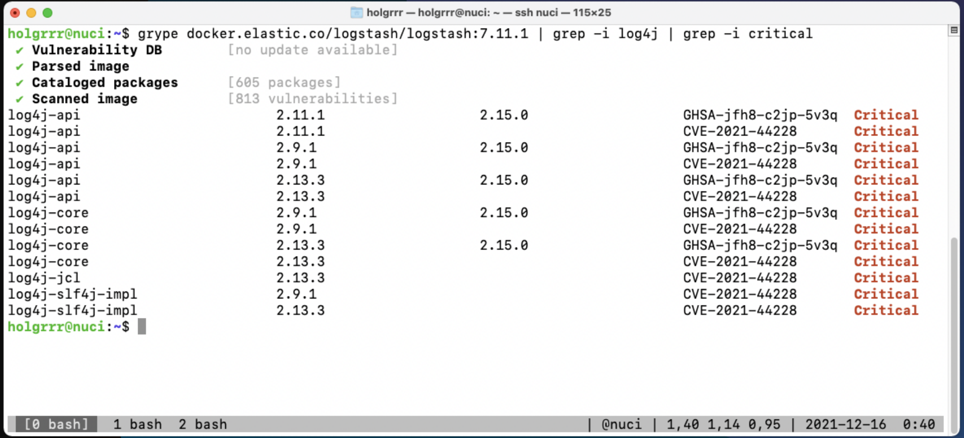
- Native support for OCI images
- Singularity Image Format mostly possible (with workaround)

- Mostly standalone scanners, often also integrable in container registry

- Work according to identical principle

Scanning simple, decision how to proceed complicated

- Deactivate image?
- Rebuild image?
- Use in strongly isolated environment?

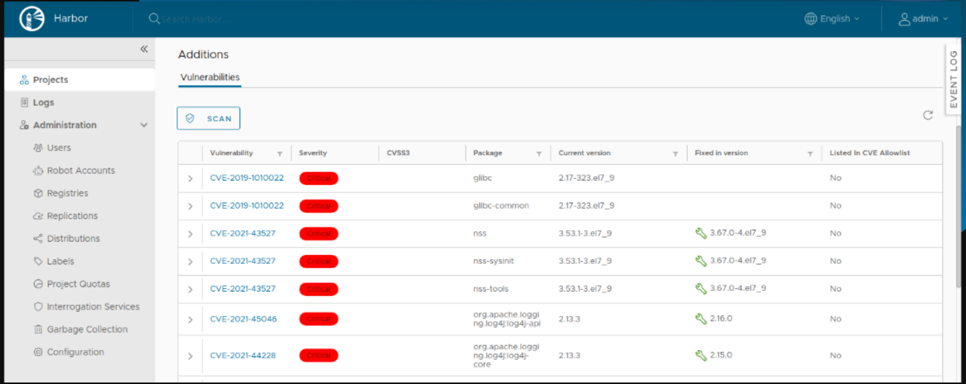


```
holgrrr@nuci:~$ grype docker.elastic.co/logstash/logstash:7.11.1 | grep -i log4j | grep -i critical
 ✔ Vulnerability DB          [no update available]
 ✔ Parsed image
 ✔ Cataloged packages        [605 packages]
 ✔ Scanned image             [813 vulnerabilities]
log4j-api                 2.11.1                2.15.0                  GHSA-jfh8-c2jp-5v3q  Critical
log4j-api                 2.11.1                                        CVE-2021-44228       Critical
log4j-api                 2.9.1                 2.15.0                  GHSA-jfh8-c2jp-5v3q  Critical
log4j-api                 2.9.1                                         CVE-2021-44228       Critical
log4j-api                 2.13.3                2.15.0                  GHSA-jfh8-c2jp-5v3q  Critical
log4j-api                 2.13.3                                        CVE-2021-44228       Critical
log4j-core                2.9.1                 2.15.0                  GHSA-jfh8-c2jp-5v3q  Critical
log4j-core                2.9.1                                         CVE-2021-44228       Critical
log4j-core                2.13.3                2.15.0                  GHSA-jfh8-c2jp-5v3q  Critical
log4j-core                2.13.3                                        CVE-2021-44228       Critical
log4j-jcl                 2.13.3                                        CVE-2021-44228       Critical
log4j-slf4j-impl          2.9.1                                         CVE-2021-44228       Critical
log4j-slf4j-impl          2.13.3                                        CVE-2021-44228       Critical
holgrrr@nuci:~$ 






 [0 bash]  1 bash  2 bash                                             | @nuci | 1,40 1,14 0,95 | 2021-12-16  0:40
```



Additions

Vulnerabilities

	Vulnerability	Severity	CVSS3	Package	Current version	Fixed in version	Listed In CVE Allowlist
>	CVE-2019-1010022	Critical		glibc	2.17-323.el7_9		No
>	CVE-2019-1010022	Critical		glibc-common	2.17-323.el7_9		No
>	CVE-2021-43527	Critical		nss	3.53.1-3.el7_9	3.67.0-4.el7_9	No
>	CVE-2021-43527	Critical		nss-sysinit	3.53.1-3.el7_9	3.67.0-4.el7_9	No
>	CVE-2021-43527	Critical		nss-tools	3.53.1-3.el7_9	3.67.0-4.el7_9	No
>	CVE-2021-45046	Critical		org.apache.logging.log4j:log4j-api	2.13.3	2.16.0	No
>	CVE-2021-44228	Critical		org.apache.logging.log4j:log4j-core	2.13.3	2.15.0	No

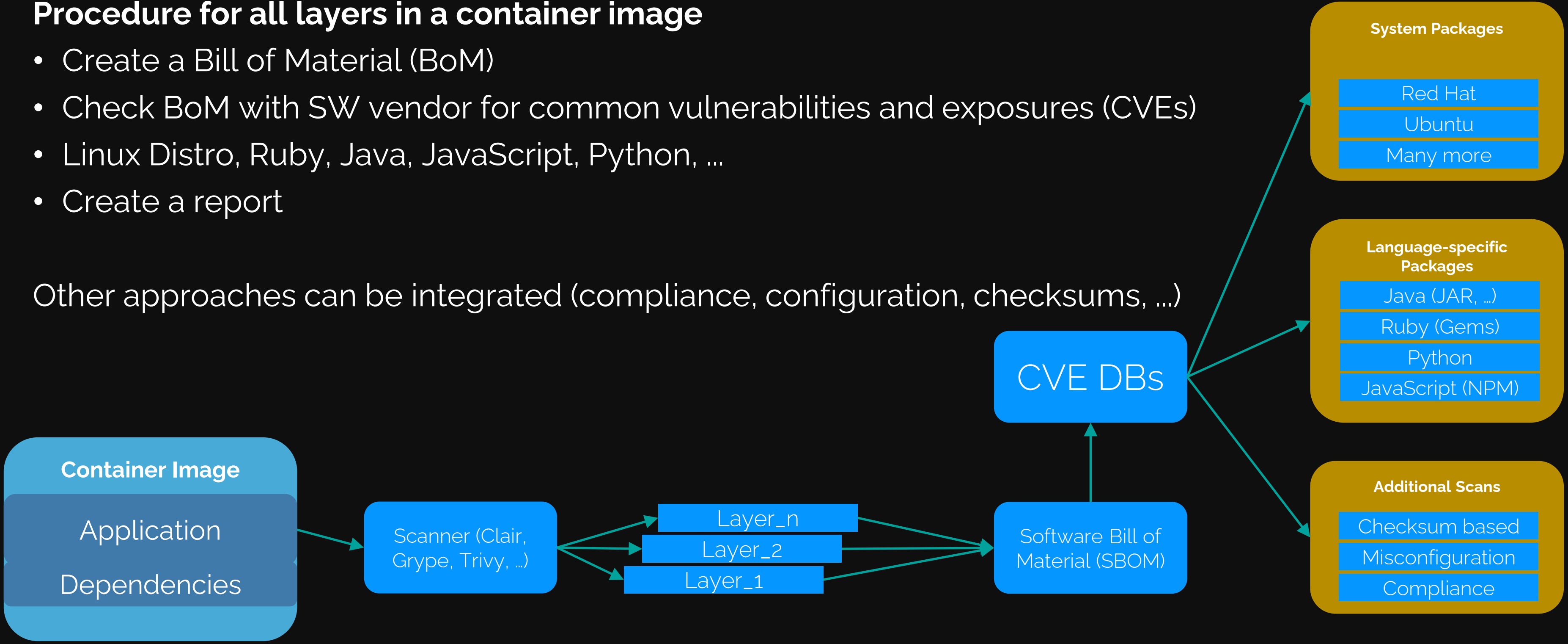
How an image scanner works

Divide et impera – divide & conquer

Procedure for all layers in a container image

- Create a Bill of Material (BoM)
- Check BoM with SW vendor for common vulnerabilities and exposures (CVEs)
- Linux Distro, Ruby, Java, JavaScript, Python, …
- Create a report

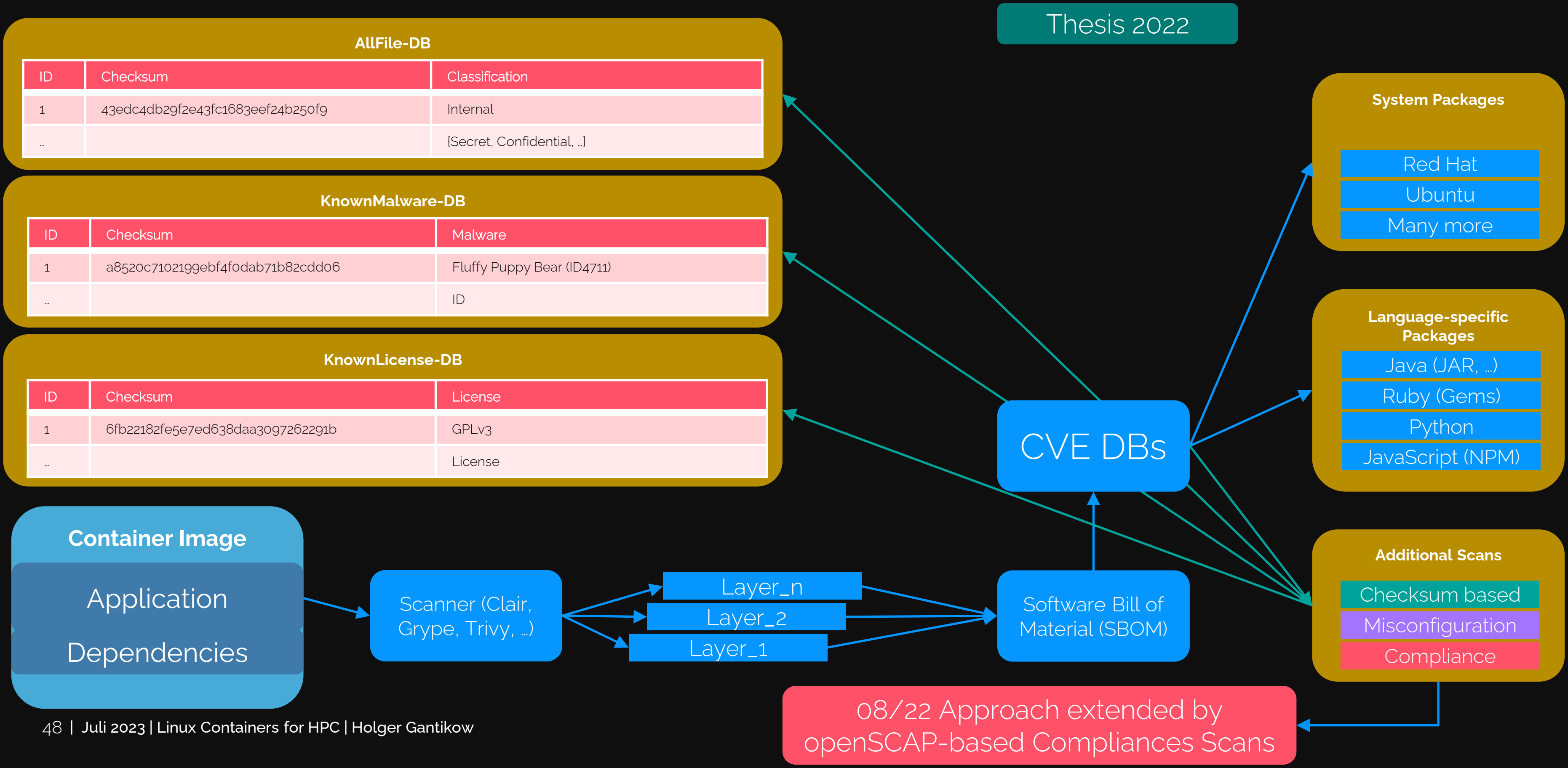
Other approaches can be integrated (compliance, configuration, checksums, …)



EVIDEN
an atos business
science + computing

Checksum approach: Connection to Data Classification

Policy: *Images Containing Files with Classification > Public -> only allowed onPrem*



Thesis 2022

AllFile-DB

ID	Checksum	Classification
1	43edc4db29f2e43fc1683eef24b250f9	Internal
…		{Secret, Confidential, …}

KnownMalware-DB

ID	Checksum	Malware
1	a8520c7102199ebf4f0dab71b82cdd06	Fluffy Puppy Bear (ID4711)
…		ID

KnownLicense-DB

ID	Checksum	License
1	6fb22182fe5e7ed638daa3097262291b	GPLv3
…		License

08/22 Approach extended by openSCAP-based Compliances Scans

Last Student Final Theses

2022/2023

CONTAINER PLUMBING DAYS 2023

https://containerplumbing.org/

Orchestrierung von HPC Container Workloads
Oliver Jessen

Atos

Orchestrating modern workloads

- Usability of HPC, AI/ML, Data Analytics workloads with K8s
- Evaluation of different gang schedulers

Sicherheit von containerisierten HPC Workloads
Cedric Casper

Atos

Security of containerized workloads

- Untrusted code in security-aware environment
- Seccomp, sidecar containers, …
- Integration with Nimbix (?)

Container-based Confidential Computing
Artur Reser (extern)

Atos

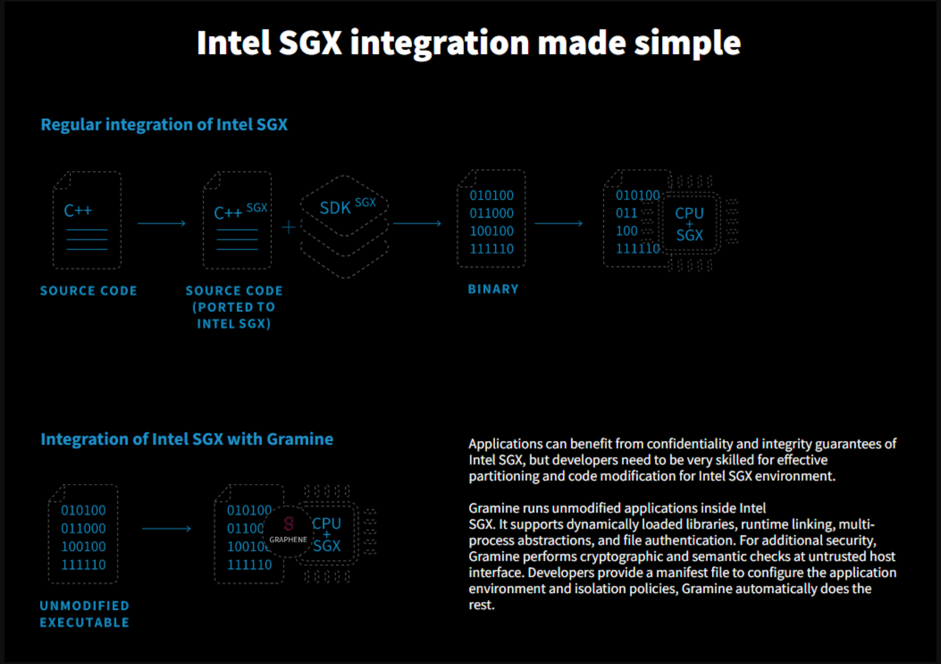
Container-based confidential computing

- Intel SGX, Trusted Execution Environments
- Unmodified code running in secure enclases
- Gramine, Gramine Shielded Containers

Gramine && gramine Shielded Containers

Few more details

- https://gramineproject.io/
- https://gramine.readthedocs.io/projects/gsc/en/latest/



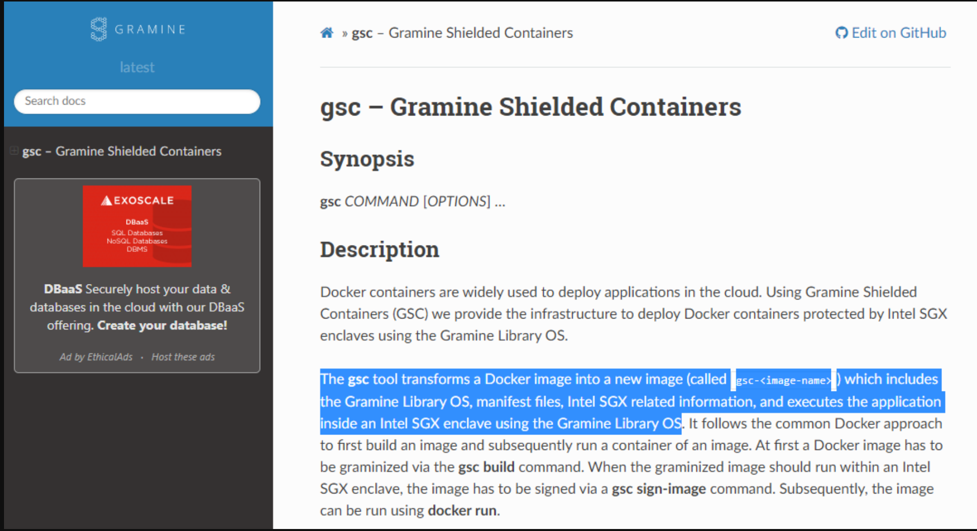
Intel SGX integration made simple

Regular integration of Intel SGX

Integration of Intel SGX with Gramine

Applications can benefit from confidentiality and integrity guarantees of Intel SGX, but developers need to be very skilled for effective partitioning and code modification for Intel SGX environment.

Gramine runs unmodified applications inside Intel SGX. It supports dynamically loaded libraries, runtime linking, multi-process abstractions, and file authentication. For additional security, Gramine performs cryptographic and semantic checks at untrusted host interface. Developers provide a manifest file to configure the application environment and isolation policies, Gramine automatically does the rest.



GRAMINE
latest
Search docs
gsc – Gramine Shielded Containers
» gsc – Gramine Shielded Containers
Edit on GitHub

gsc – Gramine Shielded Containers

Synopsis

gsc *COMMAND [OPTIONS] …*

Description

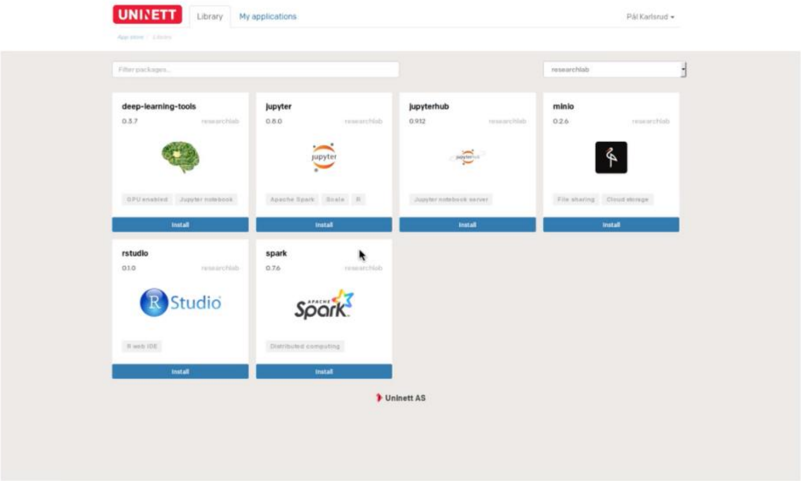
Docker containers are widely used to deploy applications in the cloud. Using Gramine Shielded Containers (GSC) we provide the infrastructure to deploy Docker containers protected by Intel SGX enclaves using the Gramine Library OS.

The **gsc** tool transforms a Docker image into a new image (called `gsc-<image-name>`) which includes the Gramine Library OS, manifest files, Intel SGX related information, and executes the application inside an Intel SGX enclave using the Gramine Library OS. It follows the common Docker approach to first build an image and subsequently run a container of an image. At first a Docker image has to be graminized via the **gsc build** command. When the graminized image should run within an Intel SGX enclave, the image has to be signed via a **gsc sign-image** command. Subsequently, the image can be run using **docker run**.

Self Service Portals

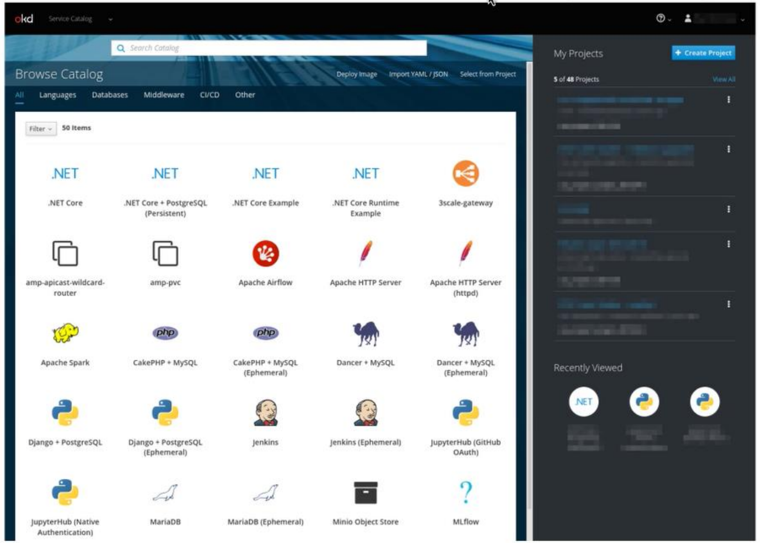
Based on the plans for LUMI-K

Containers in LUMI - LUMI-K (future)



LUMI
Introduction to containers on LUMI
LUMI User Support Team (LUST)
Several slides kindly contributed by Orian Louant, LUMI User Support Team

Containers in LUMI - LUMI-K (future)



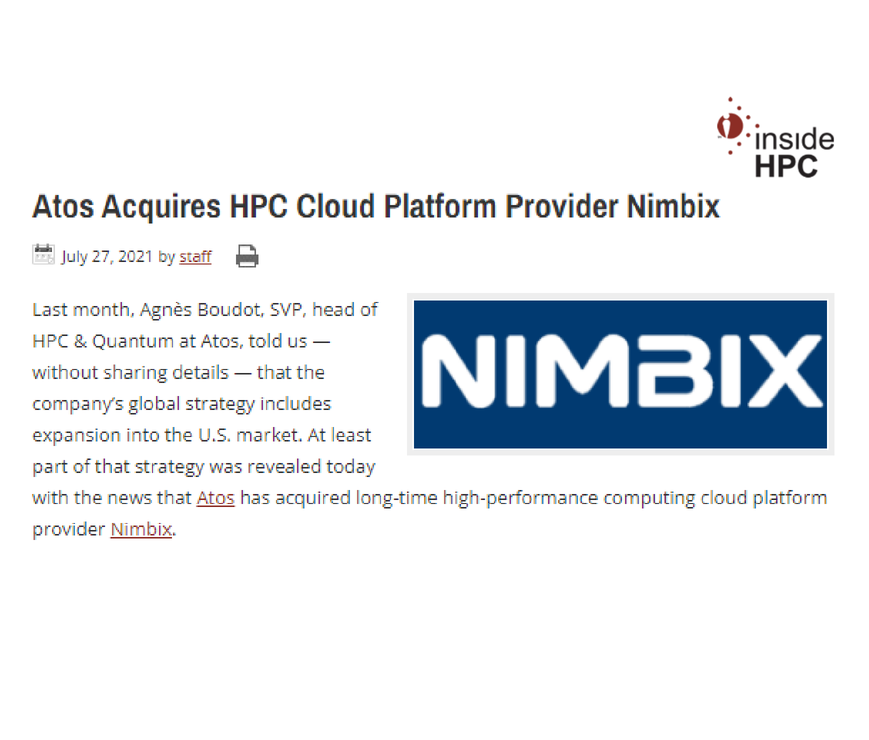
- LUMI: #3 Top500 (06/2022)
  - Future: "Cloud partition based on Kubernetes (LUMI-K)"
  - K8s Cluster running service containers
- Could also be implemented with Nimbix

05. Nimbix

Nimbix Supercomputing Suite

Atos Nimbix

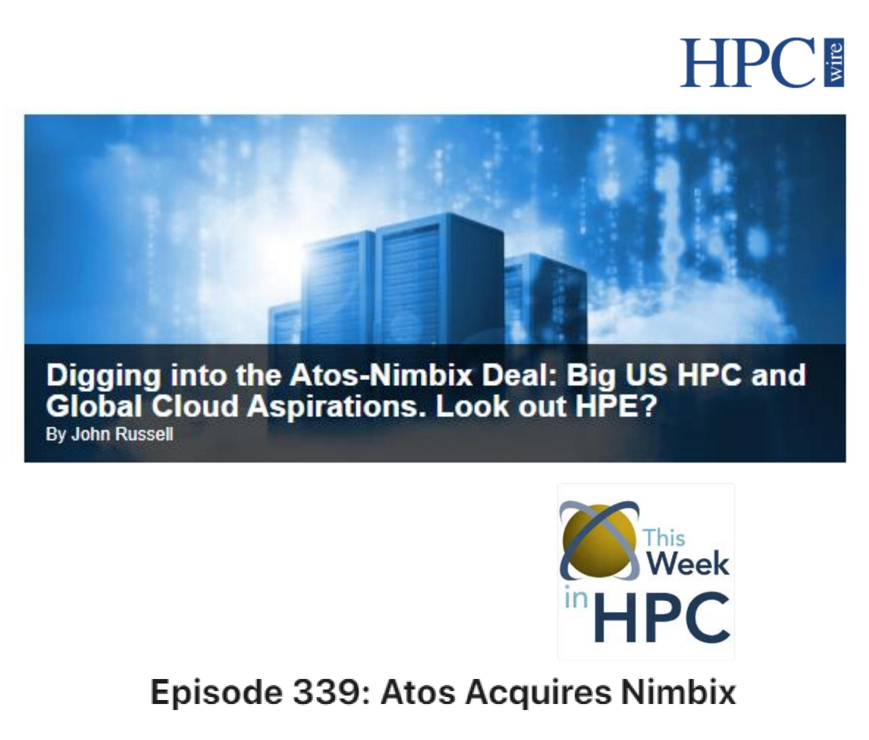
Leading-Edge CAE as a Service



Atos Acquires HPC Cloud Platform Provider Nimbix

July 27, 2021 by staff

Last month, Agnès Boudot, SVP, head of HPC & Quantum at Atos, told us — without sharing details — that the company's global strategy includes expansion into the U.S. market. At least part of that strategy was revealed today with the news that Atos has acquired long-time high-performance computing cloud platform provider Nimbix.



Digging into the Atos-Nimbix Deal: Big US HPC and Global Cloud Aspirations. Look out HPE?

By John Russell

Episode 339: Atos Acquires Nimbix

Nimbix JARVICE™

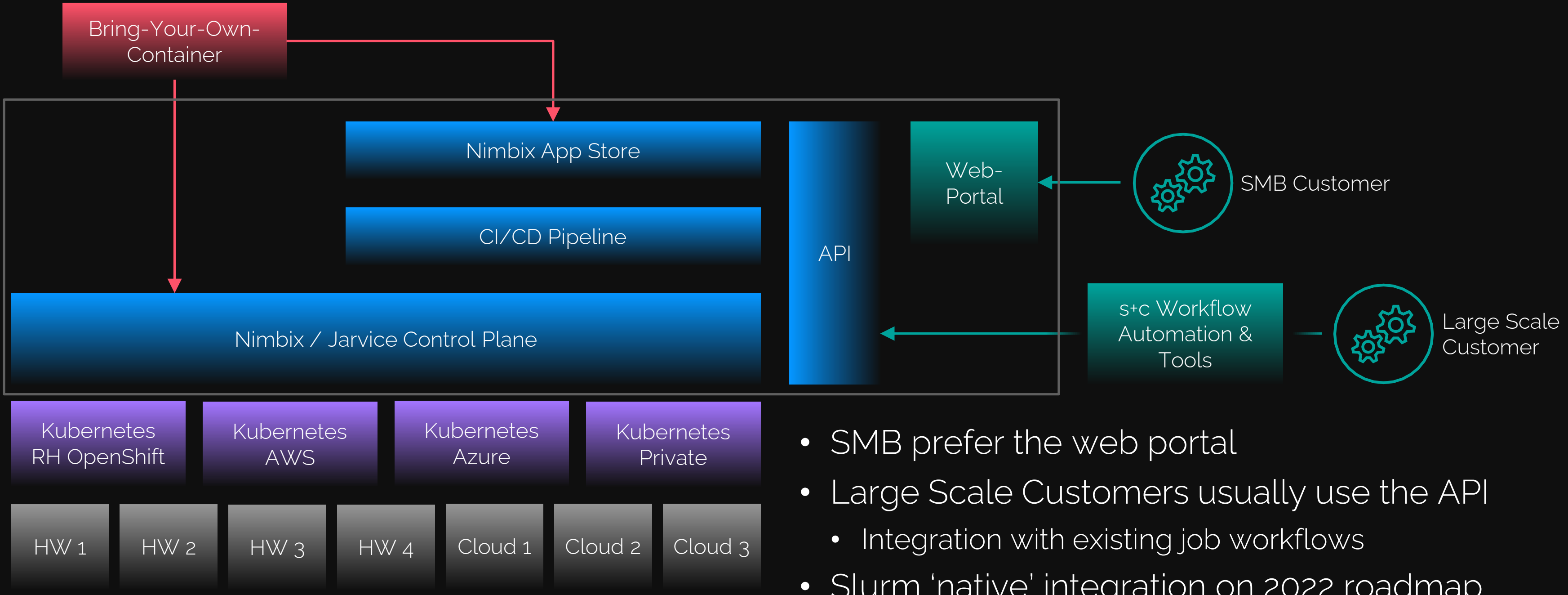
is a container-based solution for multi-site, multi-cloud HPC workloads with an integrated App marketplace and ready to use click-to-run workflows of typically used simulation applications and community AI tools.

Nimbix™

HPC in Kubernetes – simplified architecture

- Elastic: Pay-as-you-go
- Dedicated: Bare Metal as-a-Service
- Federated: Unifies all compute zones

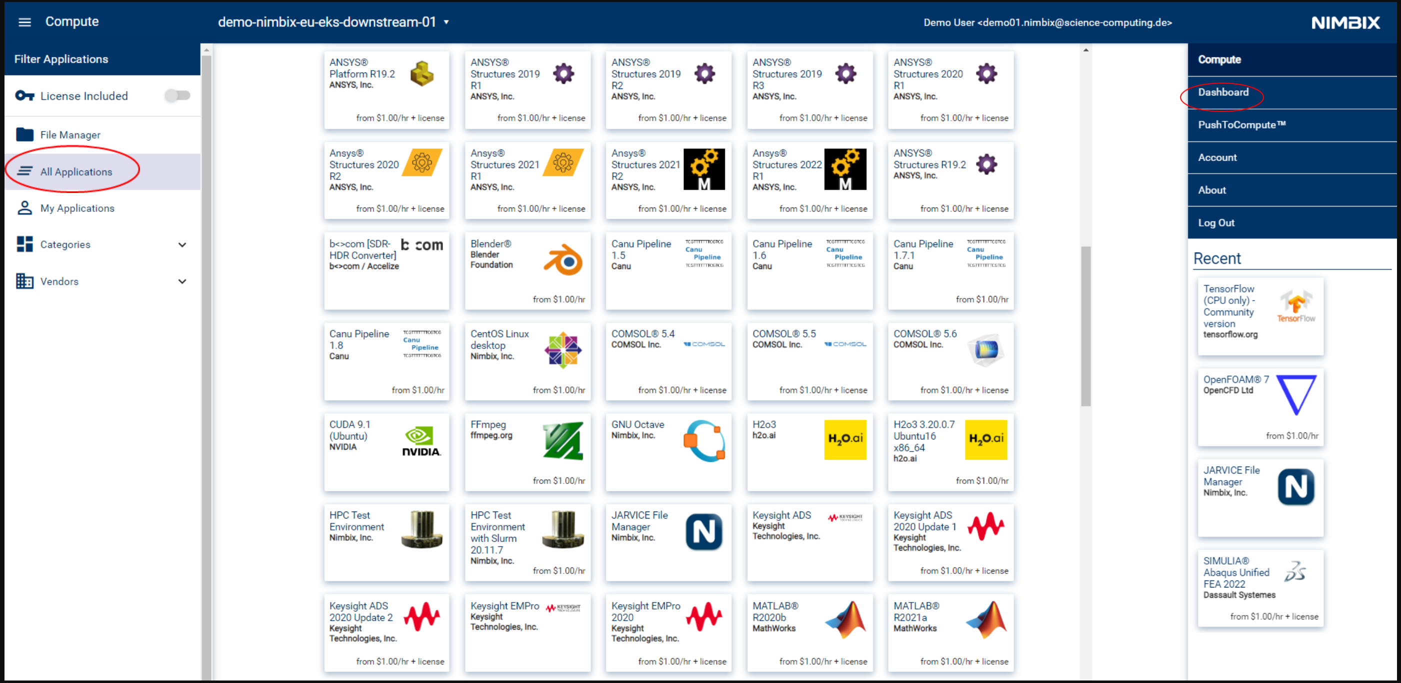
Nimbix enables HPC with containers on Kubernetes based infrastructure



- SMB prefer the web portal
- Large Scale Customers usually use the API
  - Integration with existing job workflows
- Slurm 'native' integration on 2022 roadmap

HyperHub

Ready to run commercial + popular community applications



ANSYS® Platform R19.2
ANSYS, Inc.
from $1.00/hr + license

ANSYS® Structures 2019 R1
ANSYS, Inc.
from $1.00/hr + license

ANSYS® Structures 2019 R2
ANSYS, Inc.
from $1.00/hr + license

ANSYS® Structures 2019 R3
ANSYS, Inc.
from $1.00/hr + license

ANSYS® Structures 2020 R1
ANSYS, Inc.
from $1.00/hr + license

Ansys® Structures 2020 R2
ANSYS, Inc.
from $1.00/hr + license

Ansys® Structures 2021 R1
ANSYS, Inc.
from $1.00/hr + license

Ansys® Structures 2021 R2
ANSYS, Inc.
from $1.00/hr + license

Ansys® Structures 2022 R1
ANSYS, Inc.
from $1.00/hr + license

ANSYS® Structures R19.2
ANSYS, Inc.
from $1.00/hr + license

b<>com [SDR-HDR Converter]
b<>com / Accelize

Blender®
Blender Foundation
from $1.00/hr

Canu Pipeline 1.5
Canu

Canu Pipeline 1.6
Canu

Canu Pipeline 1.7.1
Canu
from $1.00/hr

Canu Pipeline 1.8
Canu
from $1.00/hr

CentOS Linux desktop
Nimbix, Inc.
from $1.00/hr

COMSOL® 5.4
COMSOL Inc.
from $1.00/hr + license

COMSOL® 5.5
COMSOL Inc.
from $1.00/hr + license

COMSOL® 5.6
COMSOL Inc.
from $1.00/hr + license

CUDA 9.1 (Ubuntu)
NVIDIA

FFmpeg
ffmpeg.org
from $1.00/hr

GNU Octave
Nimbix, Inc.

H2o3
h2o.ai

H2o3 3.20.0.7 Ubuntu16 x86_64
h2o.ai
from $1.00/hr

HPC Test Environment
Nimbix, Inc.

HPC Test Environment with Slurm 20.11.7
Nimbix, Inc.
from $1.00/hr

JARVICE File Manager
Nimbix, Inc.

Keysight ADS
Keysight Technologies, Inc.

Keysight ADS 2020 Update 1
Keysight Technologies, Inc.
from $1.00/hr + license

Keysight ADS 2020 Update 2
Keysight Technologies, Inc.
from $1.00/hr + license

Keysight EMPro
Keysight Technologies, Inc.

Keysight EMPro 2020
Keysight Technologies, Inc.
from $1.00/hr + license

MATLAB® R2020b
MathWorks
from $1.00/hr + license

MATLAB® R2021a
MathWorks
from $1.00/hr + license

Recent

TensorFlow (CPU only) - Community version
tensorflow.org

OpenFOAM® 7
OpenCFD Ltd
from $1.00/hr

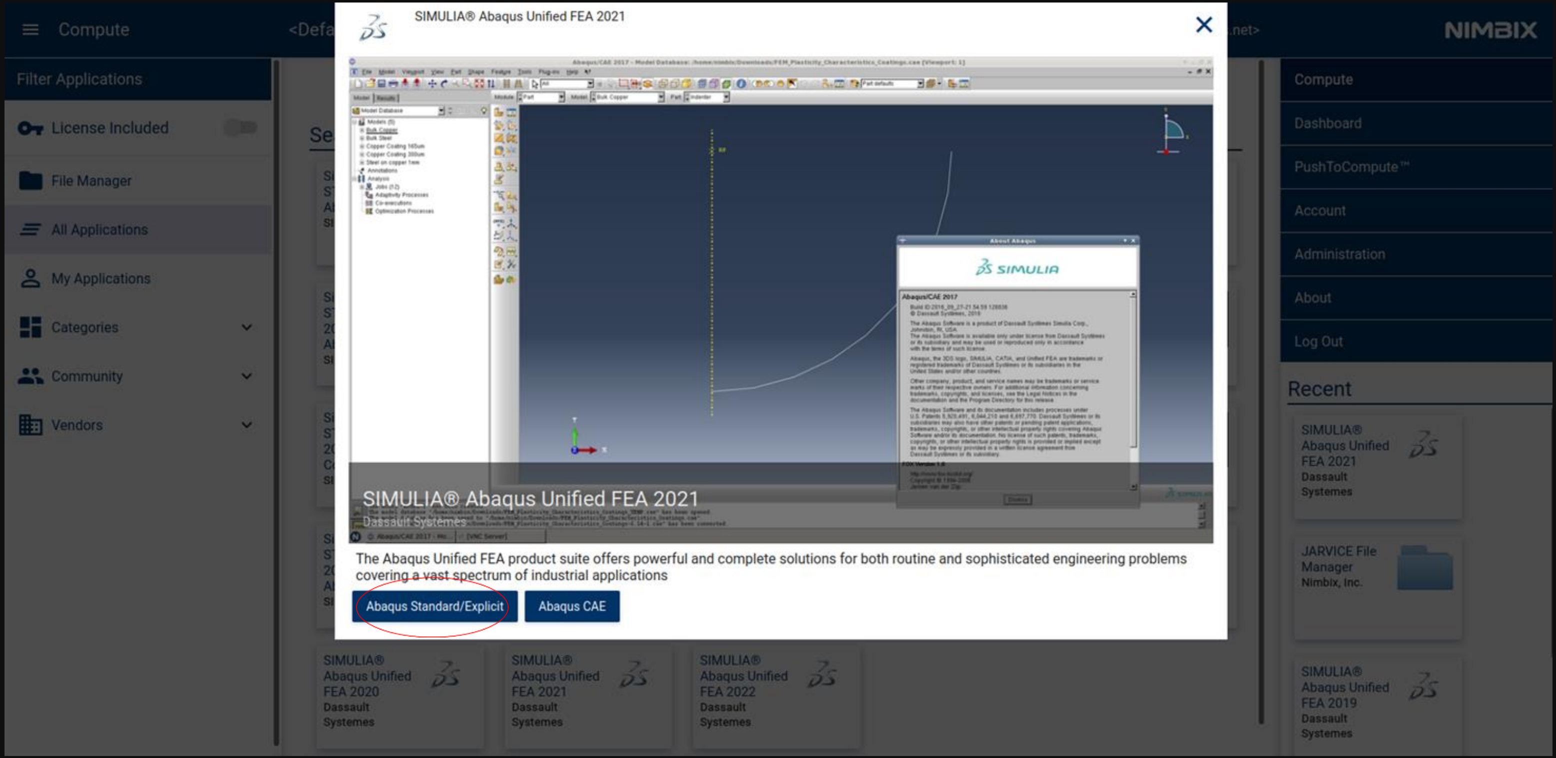
JARVICE File Manager
Nimbix, Inc.

SIMULIA® Abaqus Unified FEA 2022
Dassault Systemes
from $1.00/hr + license

- Point-and-click workflows, Support for GPU, IB, EFA, Extendable

Abaqus

Let‘s start a job!



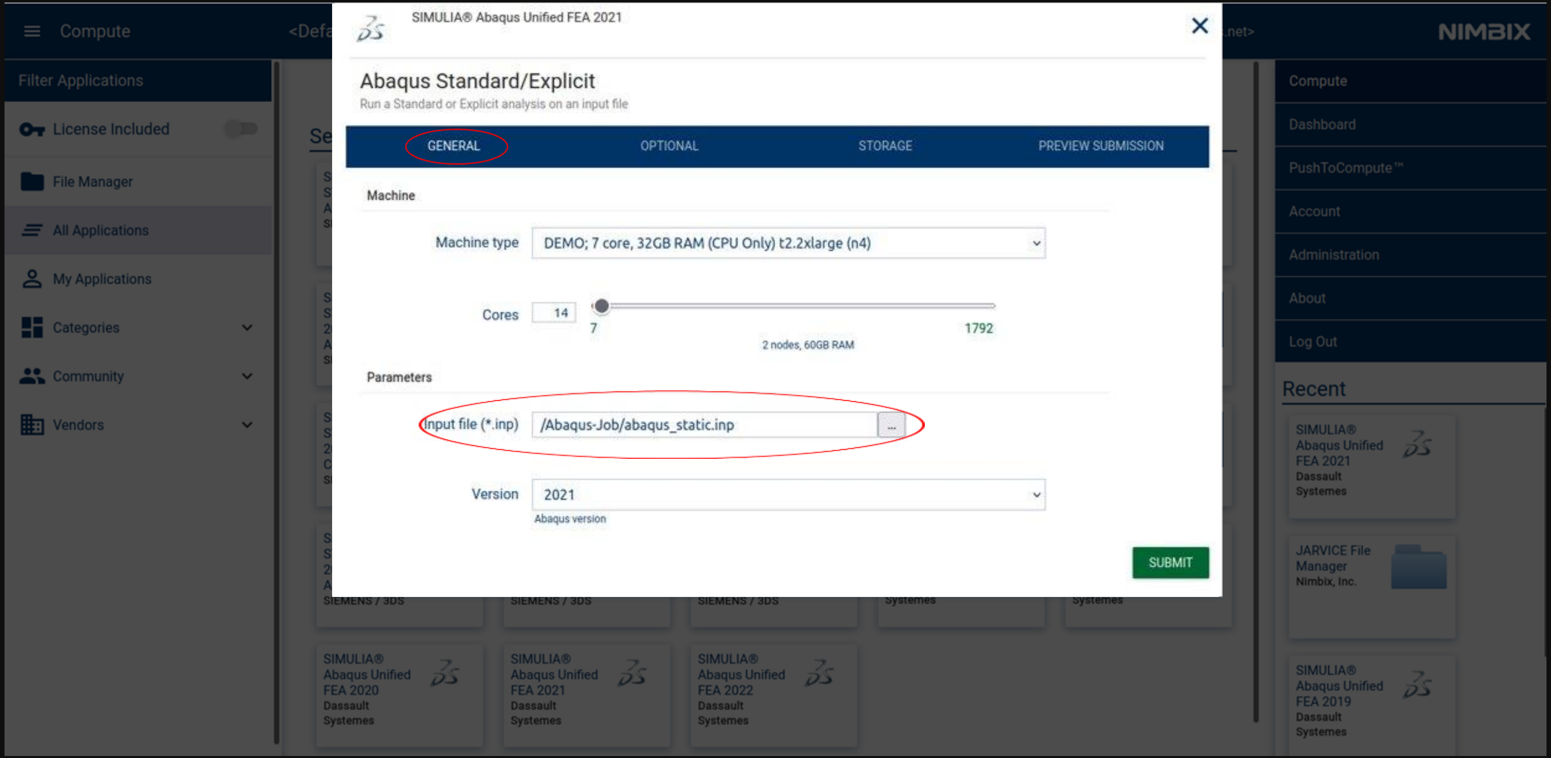
SIMULIA® Abaqus Unified FEA 2021

SIMULIA® Abaqus Unified FEA 2021
Dassault Systèmes

The Abaqus Unified FEA product suite offers powerful and complete solutions for both routine and sophisticated engineering problems covering a vast spectrum of industrial applications

Abaqus

Insert input data, Resource selection – will be *scaled up on demand*



SIMULIA® Abaqus Unified FEA 2021

Abaqus Standard/Explicit

Run a Standard or Explicit analysis on an input file

GENERAL
OPTIONAL
STORAGE
PREVIEW SUBMISSION

Machine

Machine type: DEMO; 7 core, 32GB RAM (CPU Only) t2.2xlarge (n4)

Cores: 14 (7 – 1792), 2 nodes, 60GB RAM

Parameters

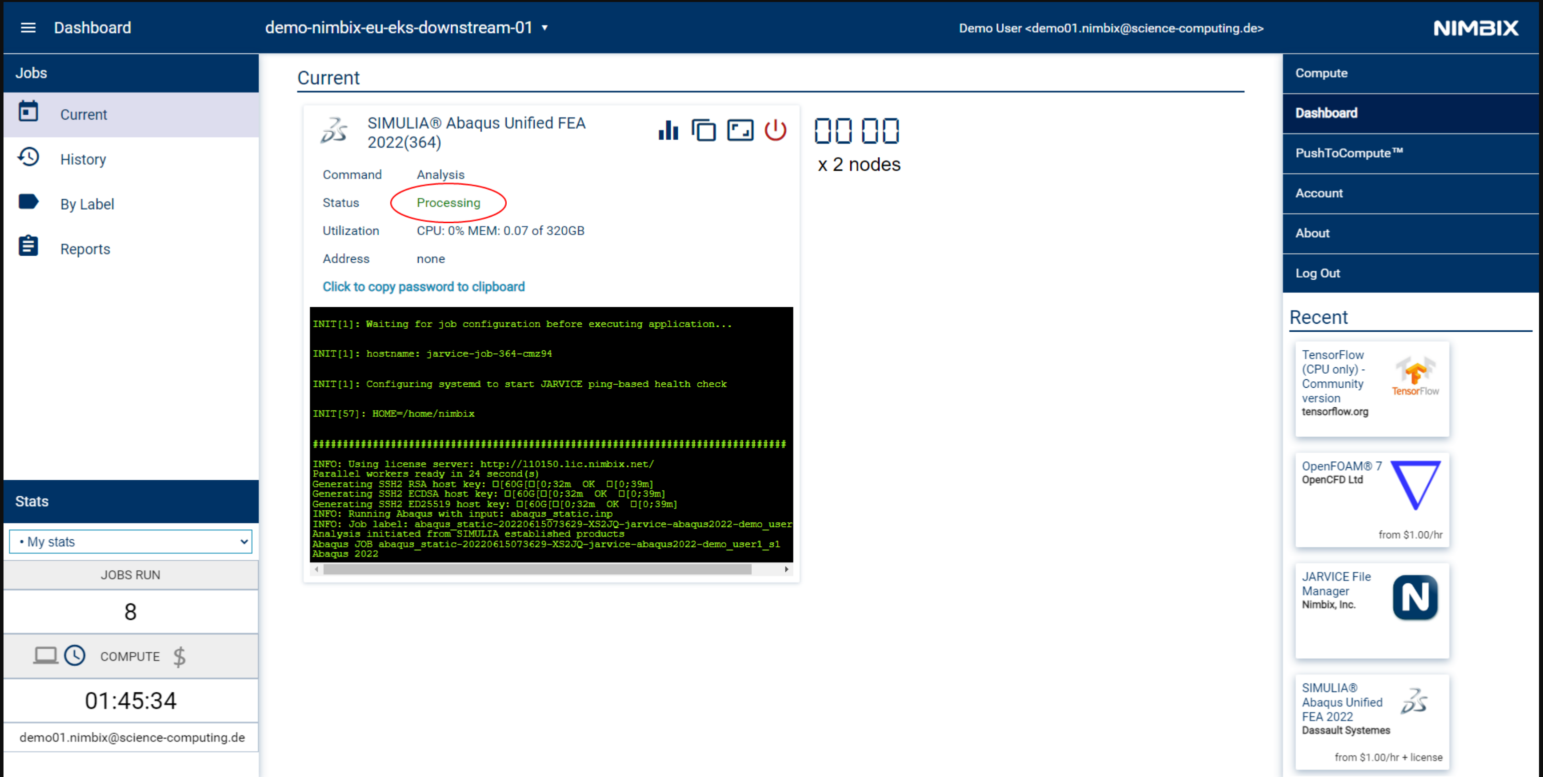
Input file (*.inp): /Abaqus-Job/abaqus_static.inp

Version: 2021

Abaqus version

SUBMIT
Compute
Filter Applications
License Included
File Manager
All Applications
My Applications
Categories
Community
Vendors
NIMBIX
Dashboard
PushToCompute™
Account
Administration
About
Log Out
Recent
SIMULIA® Abaqus Unified FEA 2021 Dassault Systemes
JARVICE File Manager Nimbix, Inc.
SIMULIA® Abaqus Unified FEA 2019 Dassault Systemes
SIMULIA® Abaqus Unified FEA 2020 Dassault Systemes
SIMULIA® Abaqus Unified FEA 2021 Dassault Systemes
SIMULIA® Abaqus Unified FEA 2022 Dassault Systemes
SIEMENS / 3DS

Abaqus
Job starts *processing*



SIMULIA® Abaqus Unified FEA 2022(364)

00:00
x 2 nodes

Command	Analysis
Status	Processing
Utilization	CPU: 0% MEM: 0.07 of 320GB
Address	none

Click to copy password to clipboard

```
INIT[1]: Waiting for job configuration before executing application...

INIT[1]: hostname: jarvice-job-364-cmz94

INIT[1]: Configuring systemd to start JARVICE ping-based health check

INIT[57]: HOME=/home/nimbix

################################################################################
INFO: Using license server: http://110150.lic.nimbix.net/
Parallel workers ready in 24 second(s)
Generating SSH2 RSA host key: □[60G[□[0;32m  OK  □[0;39m]
Generating SSH2 ECDSA host key: □[60G[□[0;32m  OK  □[0;39m]
Generating SSH2 ED25519 host key: □[60G[□[0;32m  OK  □[0;39m]
INFO: Running Abaqus with input: abaqus_static.inp
INFO: Job label: abaqus_static-20220615073629-XS2JQ-jarvice-abaqus2022-demo_user
Analysis initiated from SIMULIA established products
Abaqus JOB abaqus_static-20220615073629-XS2JQ-jarvice-abaqus2022-demo_user1_s1
Abaqus 2022
```

Nimbix Supercomputing Suite

Global Elastic HPC & Supercomputing as-a-Service

3 Supercomputing "as-a-Service" Consumption Models

Elastic

Pay-as-you-go, on-demand, secure and scalable supercomputing through a single user interface.

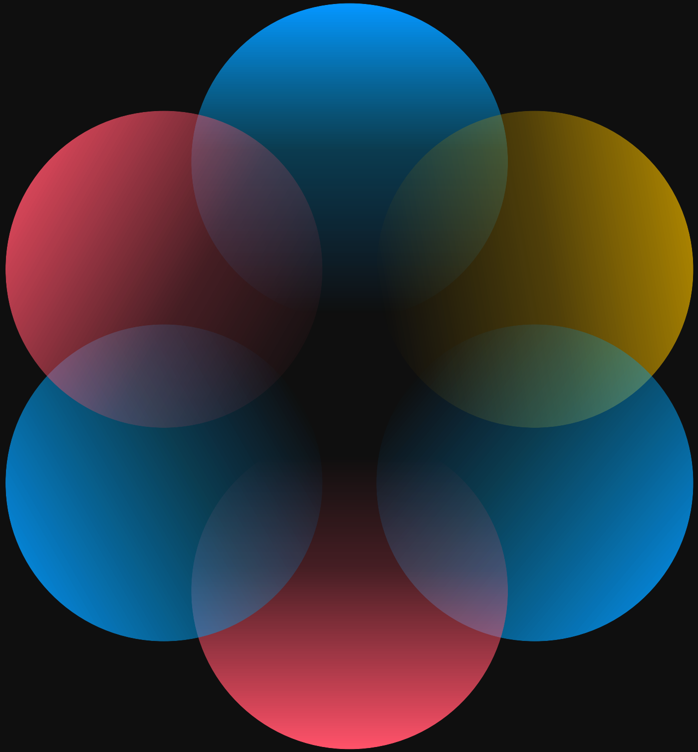
Dedicated

Leverage powerful dedicated HPC servers as "Bare Metal as-a-Service" for the best of infrastructure and on-demand scalability, convenience, and agility.

Federated

Federated Supercomputing-as-a-Service offers a unified service console to manage all compute zones and regions in a public or private HPC, AI, and supercomputing federation

06. Conclusion



Containers from the big player perspective (CAE focus)

Certification + Standardization, Off-premises Resources, SBOM, Security as drivers

Certification + Standardization

- Build based on a build file
- Defined frozen stack for use in the future as OS-base evolves
- Application + dependencies library setup as one package
- Defined stack for a project (standardized + documented)
- Possibility to share the image
- Possibility to define deviation in additional layer

Off-premises Resources

- Decoupling containers from host OS - RHEL vs SLES vs Ubuntu*
- Use of resources with different SW configuration (library setup)*
- Deployment of applications in environments under foreign administrative sovereignty (easier handling with containers)

Software Bill of Material

- Topic from last year
- What's running on my cluster?
- Which applications are affected by CVE XYZ?
- What dependencies does application X have?
- Which setup do I actually use to compute project Y?

Security Aspects

- Streamline some processes regarding new applications
- Beyond: improve status quo of HPC security with containers, fields of interest:
- Signed Images
- Seccomp Profiles
- Security Monitoring
- Behavioral Monitoring
- -> Mostly „future use"

For smaller players, driver is often "running legacy code"

Current challenges in adoption

ISVs, Registries at Scale, „FUD“ + Future Topics

Still no ISV images

- Short version: ISVs…
- have established mechanisms to distribute applications
- don't feel enough pressure?
- are not container-aware? Containers might be used internally though
- fear shift in responsibility? – base layers part of the distributed artifact (CVEs, …) – more frequent releases needed

Registries at Scale

- Best practices for registry setup – especially in distributed environments
- Mirroring / caching / pull through
- How many registries per cluster

„FUD“*, MPI, SUB*-Mapping

- Some operators like to disable user namespaces support
- Which breaks usability of some container runtimes…
- MPI compatibilities – injecting host MPI typically not an option
- Subuid/gid-Mapping problematic
- UID in the Kernel should equal the actual UID, license server…

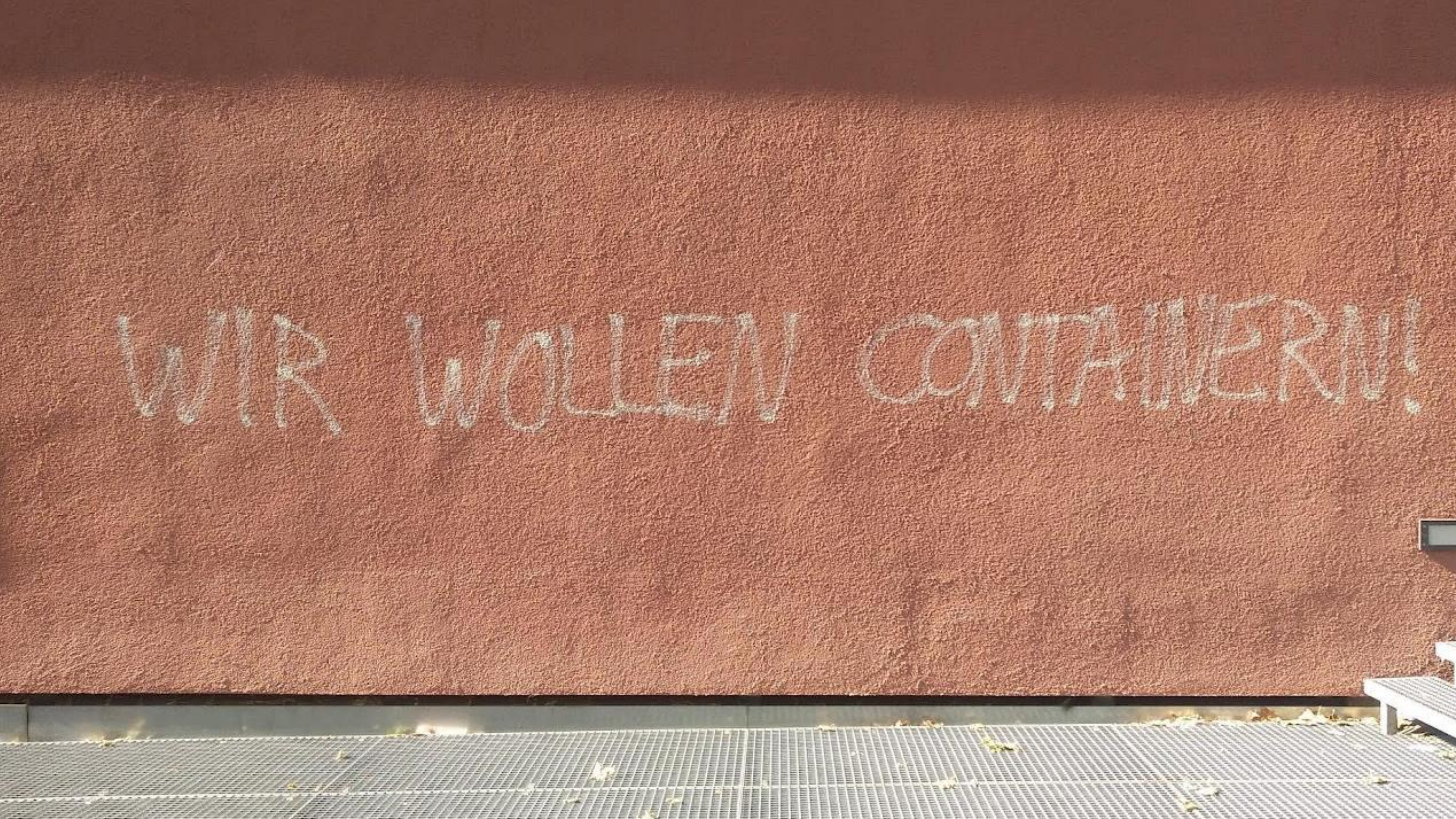
Potential Challenges + Future Topics

- Automated (re)builds
- Reproducibility in builds
- Image signing
- Fat vs multiple small images - Dealing with site specifics + Complex workflows involving multiple applications / tools
- Performance-optimized images – less of a topic, as image build is around ISV provided code
- Moving towards K8s

Conclusion

Container all the things?

- Containers have become established in the last few years and will remain so
  - Some customers are already thinking about containerizing 100% of CAE applications (Desktop+HPC)
  - Much (OCI specs) now standardized, HPC engines can run OCI images, Podman learns SIF
  - Performance overhead minimal
  - No longer only vehicle for historical SW on new OS, also standard in the domain of AI/ML
- Paving the way for better use of resources in distributed environments
  - The less host dependencies the more portable (MPI, Infiniband, Omnipath, ...)
- Currently practically no container images through the ISVs - self-build or image-aaS
- Reasonable approach: mapping in CI/CD pipelines and corresponding infrastructure
  - Enables automated build per target environment
- Containers provide opportunity to improve security of HPC environment
  - What is running in the environment? What of it has security vulnerabilities? What SW is being used at all?
- Nimbix approach highly interesting for very many customers
  - Especially when cloud resources or Kubernetes setups are to be connected.


WIR WOLLEN CONTAINERN!

EVIDEN
an atos business
science + computing

Thank you!

For more information please contact:

Holger Gantikow
holger.gantikow@atos.net

Atos