
NIMBIX
Supercomputing made super human

# 5 Years of Containerized HPC in the Cloud: Past, Present, and Future

Leo Reiter, CTO and Vice President of Software Engineering, Nimbix

NIMBIX
supercomputing made super human™

# What Are Containers?

Linux containers

- Abstraction – not actually “things”
- “Inter-modal” Method for packaging, distributing, and deploying applications and dependencies onto arbitrary infrastructure
  - VM’s, PC’s, servers, clusters, etc.
  - Consistency and integrity across platforms
  - Repeatable mechanisms
- VM’s useful in some cases, but no longer needed
  - VM’s are not compatible with HPC!

# “Demystified”…

Applicatio
n

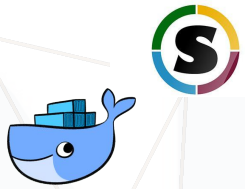

Containe
r

← Platform



Runtim
e

# Container Ecosystem: Explained



**Formats**

SINGULARITYHUB

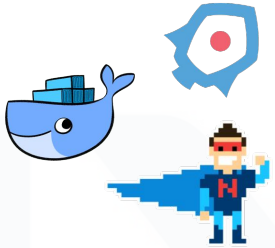
docker hub

NVIDIA

**Registries**

**Platforms**

OPEN CONTAINER INITIATIVE



**Runtimes**

# Container Solution: Complete

SINGULARITYHUB

docker hub

NVIDIA

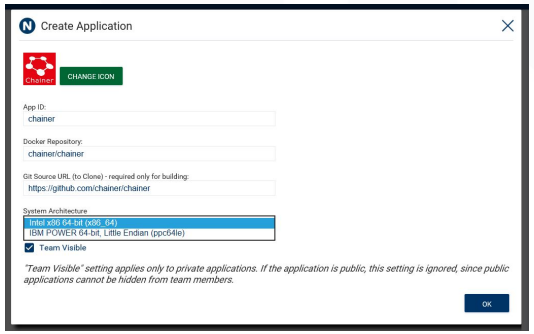
OPEN CONTAINER INITIATIVE

**Formats**

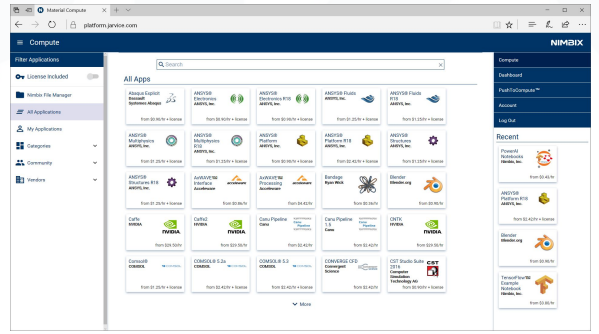
**Registries**

**Platforms**

**Runtimes**




**JARVICE**




NIMBIX
supercomputing made super human™

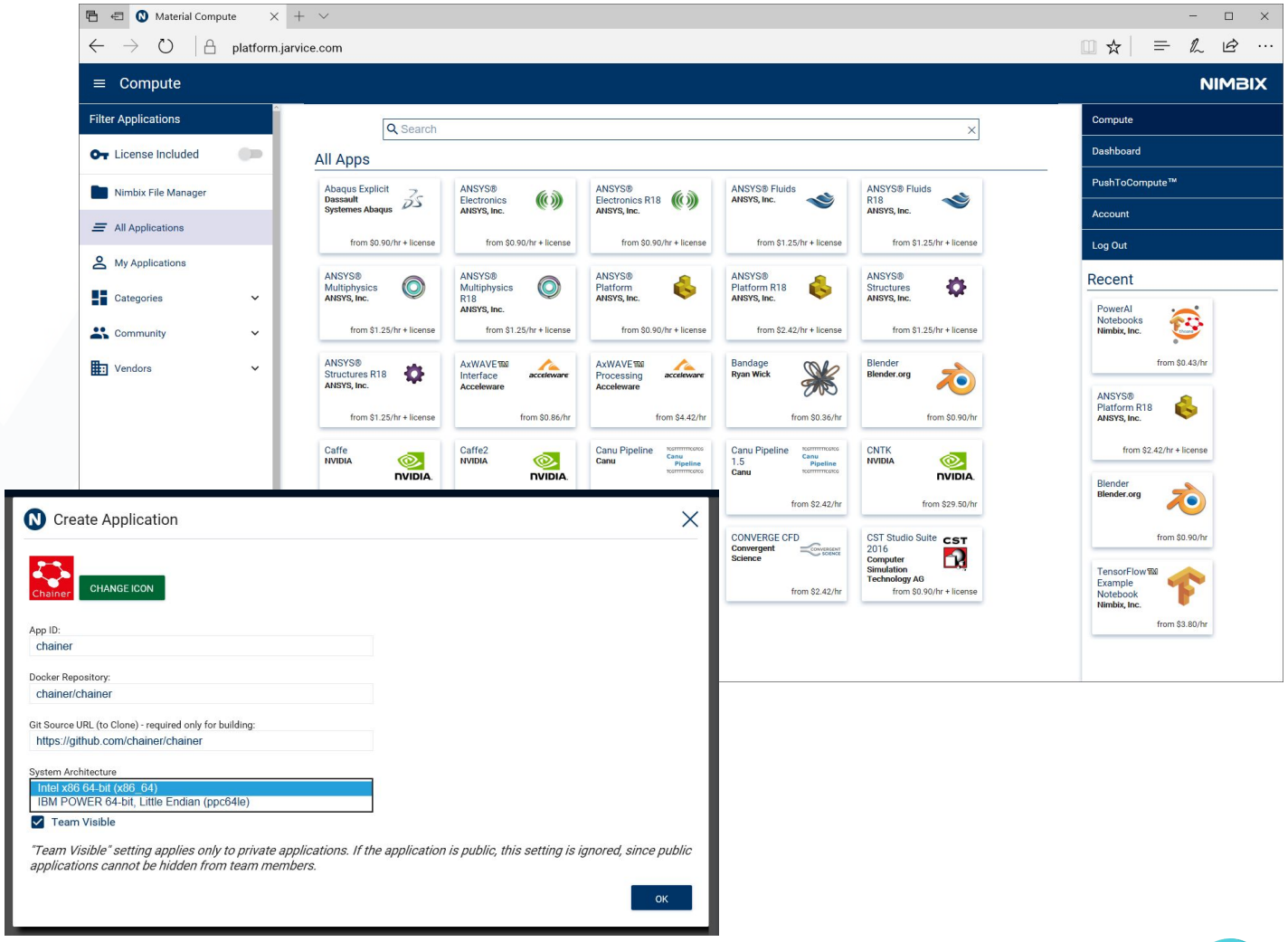
# JARVICE

A purpose built container-native cloud operating system and application delivery platform for accelerated and high performance computing.

Powers the Nimbix Cloud, now available on-premises.

Introduced containerized HPC and ML/DL, at scale to the world.

In production since 2013.




NIMBIX
supercomputing made super human™

# What is Containerized HPC?

NIMBIX
supercomputing made super human™

# What is Containerized HPC?

✓ Bare-metal – no hypervisors
- **HPC and virtualization don't mix!**

# What is Containerized HPC?

- ✓ Bare-metal – no hypervisors
  - **HPC and virtualization don't mix!**
- ✓ Large scale, no bottlenecks
  - Tightly coupled algorithm support for existing MPI and next-gen DDL
  - **Low latency fabric support**

# What is Containerized HPC?

- ✓ Bare-metal – no hypervisors
  - **HPC and virtualization don't mix!**
- ✓ Large scale, no bottlenecks
  - Tightly coupled algorithm support for existing MPI and next-gen DDL
  - **Low latency fabric support**
- ✓ Accelerated and Heterogeneous
  - x86, POWER, GPUs, FPGAs, multi-core, "many-core" - **all of the above**

# What is Containerized HPC?

- ✓ Bare-metal – no hypervisors
  - **HPC and virtualization don't mix!**
- ✓ Large scale, no bottlenecks
  - Tightly coupled algorithm support for existing MPI and next-gen DDL
  - **Low latency fabric support**
- ✓ Accelerated and Heterogeneous
  - x86, POWER, GPUs, FPGAs, multi-core, "many-core" - **all of the above**
- ✓ Workflow-oriented
  - Not machines, not microservices – **just answers**

# 3 Containerized HPC Myths...

NIMBIX
supercomputing made super human™

# 3 Containerized HPC Myths…

Myth #1:
Containers are not bare-metal!

# 3 Containerized HPC Myths…

Myth #1:
Containers are not bare-metal!

**Yes they are.**

# 3 Containerized HPC Myths…

Myth #1:
Containers are not bare-metal!

**Yes they are.**
**In the Nimbix Cloud!**

# 3 Containerized HPC Myths…

Myth #2:
Containers are hard to make!

# 3 Containerized HPC Myths…

Myth #2:
Containers are hard to make!

**No they're not!**

# 3 Containerized HPC Myths…

## Myth #2: Containers are hard to make!

https://github.com/nimbix/app-hpctest/bb/master/Dockerfile

7 lines (4 sloc) | 191 Bytes

```
1  FROM jarvice/base-centos-torque:6.0.4
2
3  RUN yum -y install nano vim emacs man && yum clean all
4  COPY 01-openmpi-path.sh /etc/profile.d/01-openmpi-path.sh
5  COPY AppDef.json /etc/NAE/AppDef.json
6
```

http://singularity.lbl.gov/docs-recipes

```
Bootstrap: docker
From: ubuntu

%help
Help me. I'm in the container.

%setup
    touch ${SINGULARITY_ROOTFS}/tacos.txt
    touch avocados.txt
```

**No they're not!**

# 3 Containerized HPC Myths…

Myth #3:
My code is not container native!

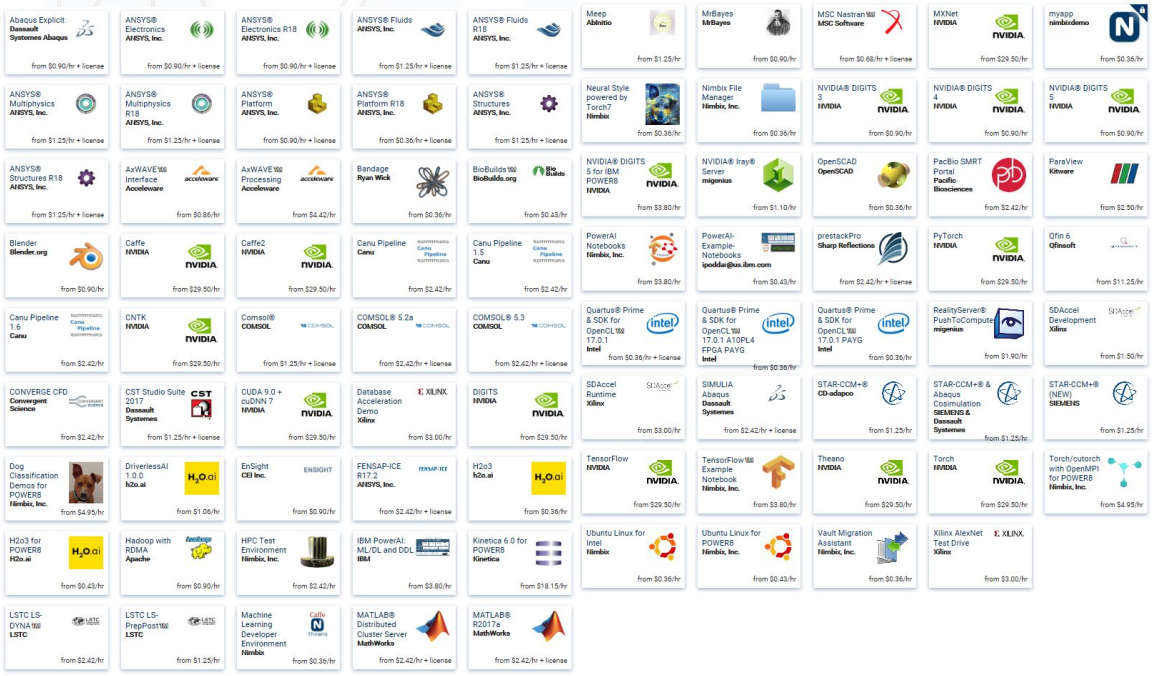
# 3 Containerized HPC Myths…

Myth #3:
My code is not container native!

**That's okay.**

# 3 Containerized HPC Myths…

## Myth #3:
## My code is not container




**That's okay.**
**Neither are most of these!**

NIMBIX
supercomputing made super human™

# How JARVICE Helps Get There

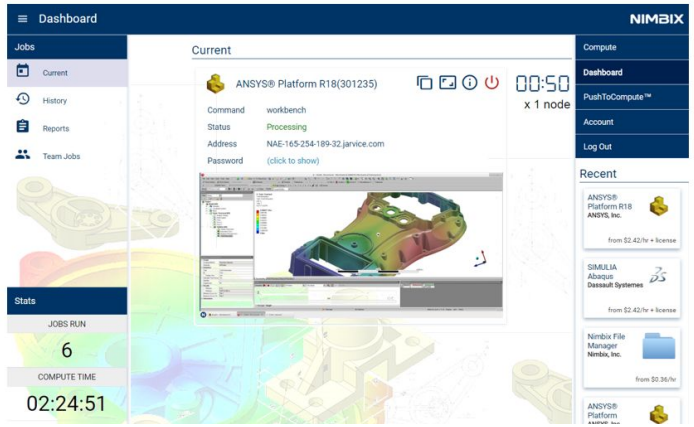
## Traditional App

- Turn-key HPC-ready environment with MPI, IB drivers, SSH, on either Ubuntu or CentOS (RHEL)
- Seamless access to state of the art acceleration
- Batch and interactive modes, including transparent 3D OpenGL hardware acceleration

```
~ % mpirun -machinefile /etc/JARVICE/nodes /usr/local/bin/solver ...
```

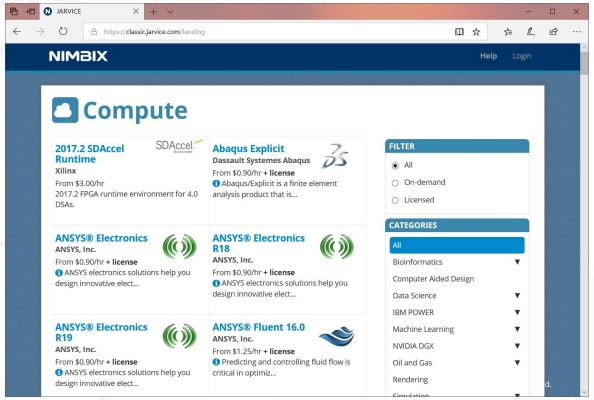
## Container-native App

- Fastest container runtime engine with the fastest starts
- Seamless access to state of the art acceleration
- Secure remote access with graphical desktop capabilities and SSH




NIMBIX
supercomputing made super human™

# Journey to Containerized HPC Cloud




**NACC**
2012

API-driven HPC SaaS

*Reconfigurable Cloud Computing*

**JARVICE 1**
2013

Containerized workflows

PaaS “v1”

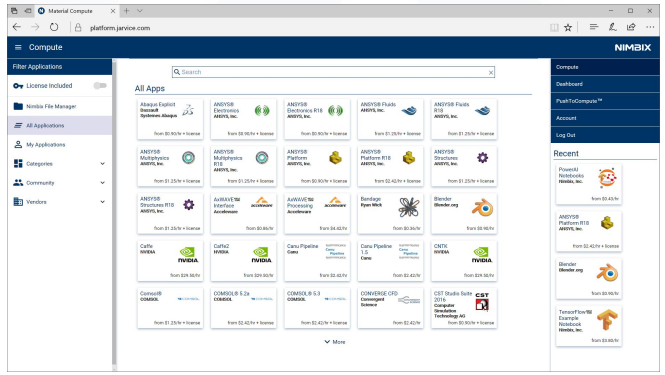
*Dynamic creation and execution of containerized applications in cloud computing*

**JARVICE 2**
2015

Extensible/scale-out SOA

Container-native

PaaS “v2”




**JARVICE 2.1**
2016

PushToCompute™

Docker and Singularity

PaaS “v3”

NIMBIX
supercomputing made super human™

# Stuff We Invented Along the Way

**Reconfigurable Cloud Computing**

- Application-defined workflows
- Processing API
- Bare-metal for cloud HPC
- Seamless programming/setup/teardown of accelerators including GPUs and FPGAs in the cloud

**Dynamic creation and execution of containerized applications in cloud computing**

- End-to-end orchestration and scheduling of workflows using Linux containers in cloud computing

# JARVICE 3 + Kubernetes

**JARVICE 3 brings end to end HPC application development, deployment, and scaling capabilities to Kubernetes!**

- Same benefits enjoyed in the Nimbix Public Cloud now available on single tenant and on premises clusters via the Kubernetes API
- HPC and distributed deep learning (DDL), side by side with Enterprise microservices and web apps
- Bare-metal or virtualized, or both
- Hundreds of commercial and open source turn-key workflow from Nimbix Public Cloud, ready to run

# How We Brought HPC to Kubernetes

- Tightly coupled workflows, capable of running MPI style applications in a turn-key manner
- Distributed container cache for very large HPC containers
  - Essential innovation of PushToCompute™
- Nimbix Public Cloud service catalog 100% compatibility
- x86 and IBM POWER are equal citizens and runtime selectable by application workflows


What About the Next 5 Years?

# What About the Next 5 Years?

- Very soon: Federated, exascale capabilities across multiple clouds under tenant control, with seamless end-user interface

# What About the Next 5 Years?

- Very soon: Federated, exascale capabilities across multiple clouds under tenant control, with seamless end-user interface
- Down the road: Data-driven workflows, for exploratory computing, including data-driven “pipelines” for scientific data processing

# What About the Next 5 Years?

- Very soon: Federated, exascale capabilities across multiple clouds under tenant control, with seamless end-user interface
- Down the road: Data-driven workflows, for exploratory computing, including data-driven “pipelines” for scientific data processing
- Not-too-distant future: ML/DL-driven platform optimization for cost and/or performance

# What About the Next 5 Years?

- Very soon: Federated, exascale capabilities across multiple clouds under tenant control, with seamless end-user interface
- Down the road: Data-driven workflows, for exploratory computing, including data-driven “pipelines” for scientific data processing
- Not-too-distant future: ML/DL-driven platform optimization for cost and/or performance
- Eventually: time travel, cold fusion, world peace

# Thank You

Questions? Comments? Thoughts?