SUSE® CaaS Platform Roadmap

Andreas Jaeger
Senior Product Manager
aj@suse.com

Roger Klorese
Senior Product Manager
roger.klorese@suse.com
Agenda

IT Transformation and Containers

SUSE CaaS Platform Overview

SUSE CaaS Platform 4 Plan
Agenda

IT Transformation and Containers

SUSE CaaS Platform Overview

SUSE CaaS Platform 4 Plan
Changing Business Demands are Transforming IT

**IT Infrastructure**
- Datacenter
- Hosted / Managed
- Cloud

**Application Deployment**
- Physical Servers
- Virtual Servers
- Containers

**Application Architecture**
- Monolithic
- N-Tier
- Microservices

**Development Process**
- Waterfall
- Agile
- DevOps
Kubernetes is the container technology winner

Container orchestration and management

Orchestration
• Scheduling
• Service discovery

Performance and availability
• Scaling
• Load balancing
• Self-healing
• Monitoring

Maintenance
• Rollout
• Rollback

---

Kubernetes
SUSE Application Delivery Solutions

Accelerate development and deployment of containerized applications to speed application delivery and increase business agility.

- **SUSE CaaS Platform**
  - Container Management

- **SUSE Cloud Application Platform**
  - Platform as a Service

**Kubernetes for the Agile Enterprise**

*Best Kubernetes operator experience*

**Advanced Cloud Native Developer Experience**

*Proven productivity for Kubernetes*
SUSE CaaS Platform

Speed application delivery to improve business agility

SUSE CaaS Platform is Kubernetes-based container management solution used by application development and DevOps teams to deploy, manage, and scale container-based applications and services.
SUSE Software-defined Infrastructure and Application Delivery Approach

**Application Delivery**
- Container Management
  - SUSE CaaS Platform
- Platform as a Service
  - SLUSE Cloud Application Platform

**Software-Defined Infrastructure**
- Private Cloud / IaaS
  - SUSE OpenStack Cloud
- Compute
  - Virtual Machine & Containers
- Storage
  - SUSE Enterprise Storage
- Networking
  - SDN and NFV
- Multimodal Operating System
  - SUSE Linux Enterprise Server

**Physical Infrastructure:** Multi-platform Servers, Switches, Storage
Transforming IT with Containers: Popular Use Cases
How do containers help transform IT?

Accelerate application development and delivery

Build and deliver new cloud native applications

Ease application transformation
Accelerate Application Development and Delivery
Spend less time deploying, more time developing
Accelerate Application Development and Delivery

Collaborate more easily

Works on my machine

Works on “our” “your” machine
Accelerate Application Development and Delivery

Release more frequently

Dev
Release Frequency
We’ve cut our cycle times in half!

Ops
Operational Stability
Things are changing too fast!
Accelerate Application Development and Delivery

Release more frequently

Dev

...and our release artifacts are bulletproof!

Release Frequency

Ops

Great! Keep ‘em coming!

Operational Stability
Build and Deliver Cloud Native Applications

Instead of larger, monolithic services …
Build and Deliver Cloud Native Applications

... increase agility with microservices

smaller codebase + smaller teams + independent functionality = Fast IT
Build and Deliver Cloud Native Applications

Containers enable microservices model

Large numbers of small containers require efficiency of shared OS Kernel
Continuous delivery demands fast start and stop capability
Container Management Challenges
Challenges with deploying containers in production

Deploying containers **across disparate, distributed infrastructure**

Ensuring developers have the **freedom to innovate**

**Controlling** the complexity of extremely dense, fast changing environment
Deploying containers across disparate, distributed infrastructure

Services required to run containers:
- **Hosts** running containers
- **Network** that containers use to communicate with each other
- **Load balancers**
- **DNS**

Capabilities to:
- Automatically (re)schedule **containers** after a host fails
- Automatically check health of workloads
- Ensure new hosts and containers are created in response to increased load

A way to **scale services** by creating new containers from existing containers
Controlling the complexity of extremely dense, fast changing environments

Challenges:
• Complexity of the integration
• Container average lifespan: ~2.5 days (VMs lifespan: ~15 days)
  Increased number of tasks for managing and monitoring containers,
  Increased complexity
• Containers typically deployed over highly distributed environments

Organizations need an easier approach to orchestrate containers and manage the underlying infrastructure services for multi-container applications.
SUSE CaaS Platform

A Kubernetes-based container management solution that enables IT and DevOps professionals to more easily and efficiently deploy and manage container-based applications and services. Use SUSE CaaS Platform to:

- **Accelerate application delivery** with Kubernetes, the industry’s leading container management platform
- **Simplify Kubernetes administration** with an exceptional platform operator experience
- **Maximize return on investment** with a flexible, no lock-in solution for today and tomorrow
Complete Solution Stack

SUSE Application Delivery

Applications
Application Lifecycle Management
Kubernetes
Datacenter Integration
Container Execution

Enterprise Hardening
Provides advanced sports modelling services to clients, including professional gamblers

Uses SUSE CaaS Platform to

| Cut operating costs while increasing flexibility and resource utilization | Increase availability of mission-critical services | Accelerate time-to-market for new features to stay ahead of competitors. | Minimize risk during migration and ongoing operations |
Uses SUSE CaaS Platform to

| Deploy production-ready Kubernetes clusters 75 percent faster | Improve resiliency, meet demand spikes, and keep operating costs low | Speed adoption of blockchain technology to days instead of months | Reach new customers with private cloud deployments that protect sensitive data and ease compliance |
“Datalounges chose SUSE CaaS Platform as our Kubernetes solution based on the very positive and productive relationship we have established with the company. SUSE CaaS Platform meets our key requirements for proven stability, relative ease of deployment, and compatibility with our OpenStack infrastructure. It has always been a pleasure to work with SUSE, and we look forward to growing our business with SUSE CaaS Platform.” - Florian Rommel, CTO and Co-founder of Datalounges
Ready TODAY!
SUSE CaaS Platform Partners
Where We Are Today:
SUSE CaaS Platform 3
SUSE CaaS Platform 3

- Optimize your cluster configuration
- More efficient and secure container image management
- Kubernetes version updates
Optimize Your Cluster Configuration

Improved integration of private and public cloud storage
- Kubernetes CPI
- Leverage public cloud IaaS resources from AWS, Azure, GCP
- Can be used to replace Magnum on OpenStack

Automatic deployment of Kubernetes software load balancer

Toolchain module for OS customization
- Build custom drivers (e.g. NVIDIA GPU)
- Troubleshoot system

Cluster reconfiguration accommodates change without downtime
- Add/remove master nodes
- Add/remove worker nodes (graceful and forced removal)
Container Image Management

Local registries improve security and performance

- Local mirrors of external registries

System-wide certificates ease access to internal resources

- Access any on-premise service that uses self-signed certificates (e.g. OpenStack endpoints, Ceph storage)
- Propagates certificates to all nodes in a cluster

Lightweight CRI-O container runtime (tech preview)

- Future alternative to open source Docker runtime (which is also included and supported)
- OCI-compliant
- Designed specifically for Kubernetes – lightweight, easy to manage/maintain
Don’t panic!

- New container engine available: CRI-O
- Tech preview in v3
- Open-source docker engine is default
Why CRI-O?

- Deliver a component that just does the job - and nothing more
- Docker open-source engine is not optimized for Kubernetes:
  \[ kubelet \rightarrow docker-shim \rightarrow dockerd \rightarrow containerd \rightarrow runC \]
- Designed with Kubernetes in mind:
  \[ kubelet \rightarrow crio \rightarrow runC \]
- Lightweight: offers better performance
- Easier to maintain and to debug
- OCI compliant: uses runC
Impact on customers

- **No** need to change container images
- **No** need to change the way to distribute images (pull from docker registries)
- **No** need to change Kubernetes manifest files
- The Container Runtime Interface is completely transparent to end-user
- However, debugging on a node is a bit different
Kubernetes Version Update

CaaS Platform 3.0 shipped with Kubernetes v1.9.8

- Apps Workloads API now stable/supported
- Facilitates orchestration (self-healing, scaling, updates, termination) of common types of workloads

CaaS Platform 3 includes Kubernetes v1.10

- Delivered as online update
- v1.10 introduces new alpha and beta features (nothing promoted to stable), and fixes
Looking Ahead:
SUSE CaaS Platform 4 Plan
SUSE CaaS Platform 3 Customer Requests

- Support larger clusters
- Track Kubernetes upstream releases more closely
- Enable non-interactive (scriptable) installation
- Support more flexible installation of OS and K8s layers
- Cluster and pod monitoring
SUSE CaaS Platform 4 Goals

- Enable familiar OS management processes and tools
- Allow updates without cluster reboot (containerize components)
- Move to standard upstream installation framework
- Scale SUSE CaaS Platform cluster to thousands of nodes
- Remove admin node as overhead and single point of failure
- Enable network policies and support complex network designs
- Scriptable (non-interactive) updates
- More cloud-native approach to container delivery
- Enable monitoring
SUSE CaaS Platform 4 Architecture

- Installable on SUSE Linux Enterprise Server 15 SP1
  - Initially in traditional mode
- Replace custom salt-based installation framework with kubeadm
  - Contribute upstream to kubeadm
- New cluster bootstrap can run outside of cluster
  - Desktop, infrastructure server, etc. - dedicated node not needed
- Upgrade to Kubernetes 1.14 or 1.15
  - Update Kubernetes quarterly (target: upstream + 60 days)
SUSE CaaS Platform 4 Architecture (cont’d)

- Add cilium as supported network plugin
  - Only for new installations
  - Enables network policies and more complex network topology
- Deliver container images via SUSE Registry
  - No more “containers as RPMs”
- Release and support Prometheus and Grafana containers
  - Monitoring and visualization
  - Allow integration into customer logging strategies
    - rsyslog, Elastic stack, Splunk, etc.
SUSE CaaS Platform 4 Benefits

- Non-disruptive full stack update (Update Kubernetes without reboot)
  - If using CRI-O and SUSE Linux Enterprise as OS
- Prompt update to new SLES service packs
- Cluster node deployment via kubeadm will give:
  - Quicker release of upstream components
  - More flexible approach
## Major Technical Changes from 3.x to 4.0

<table>
<thead>
<tr>
<th></th>
<th>SUSE CaaS Platform 3</th>
<th>SUSE CaaS Platform 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS</td>
<td>MicroOS</td>
<td>SLES 15 SP1 (see note below)</td>
</tr>
<tr>
<td>Kubernetes</td>
<td>1.10</td>
<td>Latest (1.14 or 1.15)</td>
</tr>
<tr>
<td>Default Container Engine</td>
<td>Docker</td>
<td>Cri-o</td>
</tr>
<tr>
<td>Default Network Plugin</td>
<td>Flannel (flat network)</td>
<td>Cilium (network policies)</td>
</tr>
<tr>
<td>Cluster Operations</td>
<td>Velum and salt-master/minion</td>
<td>“caaspctl”, possibly Salt (details tbd)</td>
</tr>
<tr>
<td>UI</td>
<td>GUI, no scripting possible</td>
<td>CLI/API first (scripting), GUI secondary</td>
</tr>
<tr>
<td>K8s Install/Update</td>
<td>Own framework</td>
<td>upstream Kubeadm</td>
</tr>
<tr>
<td>Container Delivery</td>
<td>wrapped in RPMs</td>
<td>containers via SUSE registry</td>
</tr>
<tr>
<td>Overhead</td>
<td>Admin node</td>
<td>No admin node needed</td>
</tr>
<tr>
<td>Deliverable</td>
<td>ISOs, Virtual images, RPMs</td>
<td>RPMs, container images (and re-use SLES install ISOs and (tbd:) virtual images)</td>
</tr>
</tbody>
</table>

Note: Support for transactional updates in SLES and/or MicroOS will be delivered later
What stays the same?

- **Kubernetes experience!** It will get richer with version 4 due to newer Kubernetes but all investment in K8s learning and implementation will carry forward.
- **Workloads will run the same way** – change of network and container runtime is transparent
- **Role based access control**
- **External authentication (plugable into AD/LDAP etc)**
- **Helm as Kubernetes package manager**
- **Cluster wider updates of Kubernetes and OS**
- **User-supplied trusted-root certs**
- **Existing storage integration**
- **...**
Release model - quarterly updates

Sync with upstream Kubernetes (k8s) releases

<table>
<thead>
<tr>
<th>CaaS Platform</th>
<th>3.0</th>
<th>4.x</th>
</tr>
</thead>
<tbody>
<tr>
<td>K8s version</td>
<td>1.9</td>
<td>1.10</td>
</tr>
</tbody>
</table>

Incremental updates - cannot skip an update. All updates (Kubernetes and OS) have to be non-disruptive.
SUSE CaaS Platform Proposed Roadmap

Based on SUSE Linux Enterprise 12 SP 3 Kubernetes 1.9

**During 3.0 lifetime**
- LDAP Integration
- Customer certificates for user facing endpoints
- Airgapped install
- Nginx Ingress controller
- Documentation update
- Kubernetes 1.10 update

**New Architecture:**
Kubeadm based
Command line first
More flexible

**Container Host OS**
Codebase: SUSE Linux Enterprise 15 SP1
Cri-o as default container engine
Monitoring (Prometheus)

**Orchestration**
Kubernetes 1.14 or 1.15
Network options (Cilium as first plugin)

Planned for 4.x releases
- Kubernetes version update
- GPU enablement
- CNI: Kuryr as Network plugin
- Istio
- Kata-container
- Support for MicroOS/SLES with transactional updates

Planned for 5.0 release
- Based on SUSE Linux Enterprise 15 SP2
- Kubernetes version updates
- ARM 64-bit support

Overall themes
- Continue to make Kubernetes easy to install, update, operate, and secure
- Multi-cluster, Multi-cloud
- Integration into customer environments (storage, networking)

* Information is forward looking and subject to change at any time.
And, Finally...
Some Related Content @ SUSECON 19

• Best Practices in Deploying SUSE CaaS Platform [TUT1131]
  • Tuesday @10:15, Wednesday @2:00
• Enabling Business Continuity with SUSE CaaS Platform [BOV1078]
  • Tuesday @2:00
• Getting Started with Kubernetes [HO1113]
  • Tuesday @2:00, Wednesday @2:00
• SUSE CaaS Platform Hands-On [HO1209]
  • Tuesday @4:30, Wednesday @2:00
• OpenStack and Kubernetes Get the Heat Treatment [HO1421]
  • Tuesday @10:15, Wednesday @10:00
• SUSE CaaS Platform vs SUSE Cloud Application Platform - Technical Tutorial [TUT1338]
  • Tuesday @10:15
• Simplifying AI Applications with Containers and K8s [TUT1151]
  • Wednesday @10:00
• Bringing container security to the next level using Kata containers [TUT1201]
  • Tuesday @4:30, Wednesday @3:15

• Visit our Kiosk in the Technology Showcase
• Join the Kubic project in the Developer Lounge
More Related Content @ SUSECON 19

- GitLab on SUSE CaaS Platform [HO1415]
  - Tuesday @10:15, Thursday @2:00
- SUSE CaaS Platform as a Service: Deploying on Demand CaaS Platform Clusters [TUT1308]
  - Tuesday @3:15, Wednesday @4:45
- SUSE Application Delivery Platform Future [FUT1429]
  - Wednesday @4:45, Thursday @3:15
- Tymlez Blockchain on SUSE CaaS Platform [BOV1313]
  - Tuesday @10:15
- SUSE CaaS Platform: Day 2 - Where Do We Go From Here? [TUT1076]
  - Tuesday @4:30
- Secure Modern Application Delivery with SUSE CaaS Platform [TUT1266]
  - Tuesday @4:30, Thursday @3:15
- Integrating Identity with LDAP for SUSE CaaS Platform [TUT1254]
  - Tuesday @10:15, Thursday @3:15
- Deploy and Scale Orchestrated Container Infrastructures on Your Own [TUT1166]
  - Tuesday @4:30

- Visit our Kiosk in the Technology Showcase
- Join the Kubic project in the Developer Lounge
Still More Related Content @ SUSECON 19

- DevOps Pipelines on Jenkins-X with SUSE CaaS Platform [DEV1118]
  - Wednesday @3:15, Friday @10:15
- Deploy a Modern Data Platform with SQL Server 2019 on the SUSE Platform [SPO1456]
  - Thursday @11:15
- openSUSE Kubic - exploring the containerized frontiers [DEV1111]
  - Tuesday @11:30, Thursday @3:15
- Geekos, Containers, and Clouds.. Oh my! [CAS1406]
  - Tuesday @4:30, Thursday @2:00
- Containerized OpenStack: Preview Inside [TUT1273]
  - Thursday @4:30
- Containers Today and Beyond [TUT1198]
  - Tuesday @10:15

Demo?
- Visit our Kiosk in the Technology Showcase
- Join the Kubic project in the Developer Lounge
Questions
Unpublished Work of SUSE LLC. All Rights Reserved.

This work is an unpublished work and contains confidential, proprietary and trade secret information of SUSE LLC. Access to this work is restricted to SUSE employees who have a need to know to perform tasks within the scope of their assignments. No part of this work may be practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of SUSE. Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

General Disclaimer

This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. SUSE makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for SUSE products remains at the sole discretion of SUSE. Further, SUSE reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All SUSE marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.