SUSE CaaS Platform

OSAD Atix München, 16.10.2018

Ralf Dannert
Systems Engineer
raff.dannert@suse.com
Enterprises Want Container Workloads in Production

- **27%** Running Today
- **44%** Planning to Run Within 1 year
- **17%** Planning to Run Within 2 yrs or more

Cloud Adoption Trends Driving IT Transformation Research Report, Insight Avenue, 2017
1412 IT decision makers in companies with 250+ employees, across all sectors, interviewed in 2017 (55% VP / C-level / Director level, 45% Senior Manager level)
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
Deploying at Scale Requires Automation

Orchestration
• Scheduling
• Service discovery

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

Maintenance
• Rollout
• Rollback
Management Platforms Exist, but are Complex

Container management platforms must also be

- Composed
- Secured
- Hardened
- Supported
- Installed
- Operated
- Scaled
- Maintained
“Building a container stack from the ground up is not for everyone.”
SUSE CaaS Platform
Speed application delivery to improve business agility

SUSE CaaS Platform is a Kubernetes-based container management solution used by application development and DevOps teams to deploy, manage, and scale container-based applications and services.
3 Key Technology Components

SUSE CaaS Platform

Orchestration
- Kubernetes

OS for Microservices & Containers
- SUSE MicroOS

Configuration
- Container Engines, Salt
## SUSE CaaS Platform

A Key Component of the SUSE Approach to IT Transformation

<table>
<thead>
<tr>
<th>Infrastructure &amp; Lifecycle Management</th>
<th>Application Delivery</th>
<th>Software-Defined Infrastructure</th>
<th>Physical Infrastructure: Server, Switches, Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUSE Manager</td>
<td>Container Management</td>
<td>Private Cloud / IaaS</td>
<td>Compute Virtual Machine &amp; Container</td>
</tr>
<tr>
<td>SUSE OpenStack Cloud Monitoring</td>
<td>SUSE CaaS Platform</td>
<td>SUSE OpenStack Cloud</td>
<td>SUSE Enterprise Storage</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SDN and NFV</td>
<td>Operating System SUSE Linux Enterprise Server</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Platform as a Service</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUSE Cloud Application Platform</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
SUSE CaaS Platform

Key Features
SUSE CaaS Platform simplifies and extends Kubernetes Container management for the enterprise

Applications

- SUSE Linux container base images
SUSE CaaS Platform simplifies and extends Kubernetes Container management for the enterprise

Application Lifecycle Management

- packaging & deployment (Helm)
- Monitoring & mgt (kube dashboard, kubectl)
- Development tools pipeline integrations, SUSE Cloud Application Platform (CAP)
- CNCF community project integrations
SUSE CaaS Platform simplifies and extends Kubernetes
Container management for the enterprise

Kubernetes
• CNCF certified distribution
SUSE CaaS Platform simplifies and extends Kubernetes
Container management for the enterprise

Datacenter Integration
- Networking
- Storage classes
- Load balancer
- Devices (e.g. GPU)
- Cloud providers
- Identity provider
SUSE CaaS Platform simplifies and extends Kubernetes
Container management for the enterprise

Container Execution
- Container runtime
- Image registries
- Container OS
SUSE CaaS Platform Technology Overview
SUSE CaaS Platform Deployment - Setup Infrastructure

1. Install Admin node
   - MicroOS one step installation
   - Create AutoYaST profile
   - Set up Admin Dashboard

2. Connect to Admin Dashboard

3. Deploy Nodes
   - Uses AutoYaST profile

4. Configure Cluster
   - Set up Kubernetes, etcd, flannel,...
SUSE CaaS Platform

Kubernetes Cluster

Dashboar

Admin Node

k8s Master

k8s Worker

k8s Master

k8s Worker

k8s Worker

k8s Worker

...
SUSE CaaS Platform Deployment – Run Containers

1. Push code to git
2. Build container image(s)
3. Run app on Kubernetes cluster
4. Ready for production

Steps:
- Push code to git
- Build container image(s)
- Run app on Kubernetes cluster
- Ready for production

Components:
- CI
- Git
- SUSE Registry
- Private Registry
- Kubernetes
  - node
  - node
  - node
  - node
  - node
- HA Proxy
- Kubernetes Dashboard / kubectl
- Dev/DevOps
CaaS Platform Deployment Scenario – End User view

End user accesses application via Ingress Controller
SUSE CaaS Platform Transactional Updates

**Automatic**
Keep Kubernetes up-to-date without manual intervention

**Transparent**
Continue to run workloads while updating Kubernetes

**Atomic**
Eliminate failures due to partial updates

**Recoverable**
Roll back easily to any previous release
SUSE CaaS Platform Setup
SUSE CaaS Platform

SUSE CaaS Platform allows you to provision, manage, and scale container-based applications. It automates your tedious management tasks allowing you to focus on development and writing apps to meet business goals.

Don't have an account? Create an account

Log In

Enter your email address

Enter your password

Log in

Remember me
Welcome! You have signed up successfully.

Initial CaaS Platform Configuration

Generic settings

Internal Dashboard FQDN/IP

prd-admin

Cluster services

- Install Tiller (Helm's server component)

Overlay network settings

Proxy settings

Next
Bootstrap your CaaS Platform

In order to complete the installation, it is necessary to bootstrap a few additional nodes, those will be the Kubernetes Master and Workers. This process leverages AutoYaST and is (almost) fully automated. In case you are not familiar with it, you can find more information about AutoYaST in the official documentation. The automatic installation gets invoked by adding `autoyast=http://prd-admin/autoyast` to the kernel parameter list. If you aren't under a PXE environment you can also use `netsetup= dhcp` kernel parameter for the network to be automatically configured using a reachable DHCP server. As installation media, you can use the very same image you bootstrapped the admin node with. A ready to use AutoYaST profile has already been generated for you during the bootstrap of the admin node. Bootstrap all the nodes you want to make part of this platform by adding the following boot parameter `autoyast=http://prd-admin/autoyast`.

Tips

You don't need to boot each node by hand. More information on how to embed an AutoYaST profile in your PXE environment is available here. Where `http://prd-admin/autoyast` is the real, generated path to the AutoYaST profile served by the dashboard.
A supported deployment of SUSE CaaS Platform requires a minimum of three nodes. Please select a minimum of three nodes.

Select nodes and roles

No nodes found

You must spawn your cluster before bootstrapping it. Your nodes will automatically appear once they are up and running, and they have been accepted on the table below.

Pending Nodes

Accepting nodes into the cluster might take a while. Be aware that it’s not possible to accept a new node while another node is being bootstrapped.

<table>
<thead>
<tr>
<th>ID</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1d86ca48bae412b0d0b8459e097b0ca</td>
<td>Accept Node</td>
</tr>
<tr>
<td>3b2411d3492466980c0d49bc08b2353</td>
<td>Accept Node</td>
</tr>
<tr>
<td>582974e01f8492b0d05a7e568f1addf</td>
<td>Accept Node</td>
</tr>
<tr>
<td>74b809a63d427d84f6e1499bd867</td>
<td>Accept Node</td>
</tr>
<tr>
<td>b2edd16ed2b98d808ec5726d87475d6</td>
<td>Accept Node</td>
</tr>
<tr>
<td>f33fa646f1426d8d1358be874444a</td>
<td>Accept Node</td>
</tr>
</tbody>
</table>
A supported deployment of SUSE CaaS Platform requires a minimum of three nodes. Please select a minimum of three nodes.

Select nodes and roles

No nodes found

You must spawn your cluster before bootstrapping it. Your nodes will automatically appear once they are up and running, and they have been accepted on the table below.

Pending Nodes

Accepting nodes into the cluster might take a while. Be aware that it’s not possible to accept a new node while another node is being bootstrapped.

<table>
<thead>
<tr>
<th>ID</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1d86cda48bac412bad0db8459e097dca</td>
<td>Acceptance in progress</td>
</tr>
<tr>
<td>3b2411d3c492469b9c0d49bc408b2353</td>
<td>Acceptance in progress</td>
</tr>
<tr>
<td>5829740e184922bd05a7e568f188def</td>
<td>Acceptance in progress</td>
</tr>
<tr>
<td>74b8c0f9a6d3427c846fcee1499bde67</td>
<td>Acceptance in progress</td>
</tr>
<tr>
<td>b2eddafeb0b84e0698e5726a87475d6</td>
<td>Acceptance in progress</td>
</tr>
<tr>
<td>f33fa6d46f1426d8d1358be814a44aa</td>
<td>Acceptance in progress</td>
</tr>
</tbody>
</table>
### Select nodes and roles

6 nodes found

After choosing the master and clicking "Next" all the other selected nodes will be set to the worker role.

<table>
<thead>
<tr>
<th>ID</th>
<th>Hostname</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>5829740e1f8f492db0a7e568f1866df</td>
<td>linux</td>
<td>Master Worker Unused</td>
</tr>
<tr>
<td>746bc019ed63d427cd46fcede149b8e87</td>
<td>linux</td>
<td>Master Worker Unused</td>
</tr>
<tr>
<td>b2eddafedeb084e0698ec5726a87475d6</td>
<td>linux</td>
<td>Master Worker Unused</td>
</tr>
<tr>
<td>f33fa8d6f41268d1358be81444aa</td>
<td>prd-node1.home.decanha-knight.net</td>
<td>Master Worker Unused</td>
</tr>
<tr>
<td>1d86c6a48b4c412bad0b8459e097badc</td>
<td>prd-master.home.decanha-knight.net</td>
<td>Master Worker Unused</td>
</tr>
<tr>
<td>3b2411d3c492f6b9ec0d499bc40b2353</td>
<td>linux</td>
<td>Master Worker Unused</td>
</tr>
</tbody>
</table>

**Pending Nodes**

You currently have no nodes to be accepted for bootstrapping.
Confirm bootstrap

Cluster specific settings

External Kubernetes API FQDN
prd-master.home.decanfa-knight.net

External Dashboard FQDN
prd-master.home.decanfa-knight.net
Before Salt Orchestration
SUSE CaaS Platform: Architectural View

Admin Node:
- Velum
- MariaDB
- Salt-api
- Salt master
- Salt minion
- etcd discovery

Kubernetes Master Node(s):
- Salt minion
- etcd
- flanneld
- API Server
- Scheduler
- Controller
- ContainerEngine

Kubernetes Worker Node(s):
- Salt minion
- etcd-proxy
- flanneld
- kubelet
- Kube-proxy
- ContainerEngine
- Container
## Cluster Status

### Summary

<table>
<thead>
<tr>
<th>Total nodes</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master nodes</td>
<td>1</td>
</tr>
<tr>
<td>New nodes</td>
<td>0</td>
</tr>
<tr>
<td>Updates</td>
<td>0</td>
</tr>
<tr>
<td>Manual</td>
<td>0</td>
</tr>
</tbody>
</table>

### Nodes

<table>
<thead>
<tr>
<th>Status</th>
<th>ID</th>
<th>Hostname</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️</td>
<td>5829740e1f8492db0a7e568f186ddf</td>
<td>linux</td>
<td>worker</td>
</tr>
<tr>
<td>✔️</td>
<td>74bb0f9a63d427c846facc1499bde87</td>
<td>linux</td>
<td>worker</td>
</tr>
<tr>
<td>✔️</td>
<td>b2edaf9c0b846698e5726a874756d</td>
<td>linux</td>
<td>worker</td>
</tr>
<tr>
<td>✔️</td>
<td>f33f0b1d61f1426d6135ebe8146449</td>
<td>prd-node1.home.decanha-knight.net</td>
<td>worker</td>
</tr>
<tr>
<td>✔️</td>
<td>1d86e9a8d0c412b0d0b8459097b6ca</td>
<td>prd-master.home.decanha-knight.net</td>
<td>master</td>
</tr>
<tr>
<td>✔️</td>
<td>3b2411d3c4924689bc0d49bc408b2353</td>
<td>linux</td>
<td>worker</td>
</tr>
</tbody>
</table>

### Pending Nodes

You currently have no nodes to be accepted for bootstrapping.
# SUSE® CaaS Platform

## Cluster Status

### Summary

<table>
<thead>
<tr>
<th>Total nodes</th>
<th>Updates</th>
<th>Manual</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Nodes

<table>
<thead>
<tr>
<th>Status</th>
<th>ID</th>
<th>Hostname</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔</td>
<td>5829740e1f8492b05a7e568f1868d6f</td>
<td>linux</td>
<td>worker</td>
</tr>
<tr>
<td>✔</td>
<td>74bb0f9a6d3427cb86fece1499bde87</td>
<td>linux</td>
<td>worker</td>
</tr>
<tr>
<td>✔</td>
<td>b2edafecbc84e686b8c5726a87475d6</td>
<td>linux</td>
<td>worker</td>
</tr>
<tr>
<td>✔</td>
<td>f33f8d4d41f4268d135ebe014a44aa</td>
<td>prd-node1.home.decana-knight.net</td>
<td>worker</td>
</tr>
<tr>
<td>✔</td>
<td>1d866a48b4ac412bad0b8459e097bdca</td>
<td>prd-master1.home.decana-knight.net</td>
<td>master</td>
</tr>
<tr>
<td>✔</td>
<td>3b2411d3c4924689bc0d49bc40b82353</td>
<td>linux</td>
<td>worker</td>
</tr>
</tbody>
</table>

### Pending Nodes

You currently have no nodes to be accepted for bootstrapping.

---

*SUSE® CaaS Platform 1.0.0 | © SUSE Linux 2018*
Log in to Your Account

username

password

Login
Download your kubeconfig file

You will see a download dialog that will allow you to download your kubeconfig file. Please, accept it and save it in a known location.

You can refer to it using kubectl by setting the KUBECONFIG environment variable, like KUBECONFIG=~/.Downloads/kubeconfig kubectl get nodes.

You can also save it to your home in `~/.kube/config`, `kubectl` will automatically read this file without the need to specify the KUBECONFIG environment variable.

You can navigate to the dashboard now, once you have downloaded your kubeconfig file.
SUSE CaaS Platform Futures
**SUSE CaaS Platform Release History**

<table>
<thead>
<tr>
<th>v1</th>
<th>v2</th>
<th>v2.x</th>
<th>v3</th>
<th>v3+</th>
</tr>
</thead>
<tbody>
<tr>
<td>2017</td>
<td>2018</td>
<td>2019</td>
<td>2020</td>
<td></td>
</tr>
</tbody>
</table>

### Container Host OS (MicroOS)
- **Codebase**: SUSE Linux Enterprise 12 SP2
- Designed for containers and optimized for large deployments
- Transactional updates
- Cluster dashboard for deployment and update

#### Orchestration
- Kubernetes v1.5: Complete solution for container-based workloads: deploy, scale, manage
- Cmdline capabilities
- Docker open source project version 1.12.6

#### Ecosystem
- Private registry
- Persistent storage: local, NFS, SUSE Enterprise Storage

---

### Container Host OS (MicroOS)
- **Codebase**: SUSE Linux Enterprise 12 SP3
- Tested with 50 nodes
- Multi-master cluster set up, tested with 100 nodes
- Integration with public cloud (Amazon, Azure, Google)***
- Enhance administration dashboard

#### Orchestration
- Kubernetes 1.7
- Add Kubernetes DNS module (kube-dns)
- Docker open source project version 1.12.6

#### Ecosystem
- Helm for installing containerized applications
- Enable SUSE CAP on top of SUSE CaaS Platform

---

*** Item delivered post-GA
* Information is forward looking and subject to change at any time.
SUSE CaaS Platform Roadmap

**v3**
- Container Host OS
  - Codebase: SUSE Linux Enterprise 12 SP3
  - Tested with 200 nodes
  - Enhance administration dashboard
  - Container engine alternative (cri-o) (tech preview)
  - Toolchain module
- Orchestration
  - Kubernetes 1.9; Docker 17.03
  - Loadbalancer integration (software)
  - Network options (CNI using flannel)
  - Cloud integration for Storage and network (CPI)
- Ecosystem
  - Private registry in offline mode
  - Trusted container images
  - Documentation rework, best practices
  - SUSE Container Certification Program
- Themes
  - Continue to make K8s easy to install, update, and operate
  - Multi-cloud, multi-cluster
  - Integration into customer environments (storage, networking)

**v4**
- Container Host OS
  - Codebase: SUSE Linux Enterprise 15 SP1
  - Container isolated via virtualization
  - Additional HW architectures
- Orchestration
  - Kubernetes version update
  - IPv6
- Ecosystem
  - Full Management of CaaS Platform within SUSE Manager
- Themes
  - Tools for containerized work loads

**v5**
- Container Host OS
  - Codebase: SUSE Linux Enterprise 15
  - More flexible set up
  - Enhance administration dashboard
  - Further scalability
  - Fold dedicate admin node into master cluster
  - CLI for administration dashboard
- Orchestration
  - Kubernetes version update
  - Network options (further CNI plugins like Kuryr or Calico)
- Ecosystem
  - Additional container images for workloads
  - Improved processes for maintenance

**v6**
- Container Host OS
  - Codebase: SUSE Linux Enterprise 15 SP1
  - Container isolated via virtualization
  - Additional HW architectures
- Orchestration
  - Kubernetes version update
  - IP6v
- Ecosystem
  - Full Management of CaaS Platform within SUSE Manager
- Themes
  - Tools for containerized work loads

**v7+**
- Container Host OS
  - Codebase: SUSE Linux Enterprise 15 SP1+
- Orchestration
  - Kubernetes version update
- Themes
  - Tools for containerized work loads

* Information is forward looking and subject to change at any time.
What’s New in SUSE CaaS Platform 3?
SUSE CaaS Platform 3
What’s new?

Optimize your cluster configuration

- Improved integration of private and public cloud storage
- Automatic deployment of Kubernetes software load balancer
- Toolchain module for MicroOS customization
- Cluster re-configuration (single/multi-master)

More efficient and secure container image management

- Local registries improve security and performance
- System-wide certificates
- Lightweight Cri-O container runtime (Tech Preview)

Kubernetes version update (1.9)

- Apps Workloads API facilitates orchestration of common workloads
Kubernetes CPI

- Take advantage of the IaaS where Kubernetes is running
- Leverage storage, Load Balancer as a Service ...
- Works with OpenStack, Azure, AWS, GCE
- CaaS Platform v3 can be used to replace Magnum on OpenStack
SUSE MicroOS Toolchain module

• Include tools to debug a system and build drivers
• Module not enabled by default

Use case 1:
• Vendor provides kernel driver only via DKMS
• Toolchain module can be used to build the driver on CaaS Platform
• Example: build NVIDIA GPU drivers

Use case 2:
• Troubleshooting system
Cluster Reconfiguration

• Prior to v3:
  – Cluster topology fixed in time
  – Customer could add only worker nodes
  – No node could be removed from the cluster

• Starting with v3:
  – Master nodes can be added
  – Node removal is supported
Cluster Reconfiguration: Growing a cluster

- Both master and worker nodes can now be added to a running cluster
- The topology of the cluster can change: from single master to multi master
Cluster Reconfiguration: Node removal

- It’s possible to remove both master and worker nodes
- Velum prevents the cluster from being unusable:
  - It must have at least one master
  - It must have at least one worker
- Velum warns the user about having unsupported cluster topologies, eg: when going from 3 master down to 1 master
- The cluster can be brought back to a supported topology by adding new master or worker nodes
Container Image Management
Groups of registries

- public registry.suse.com
- public ISV registry @suse.com
- ISV registry for running at ISV side
- public registry for openSUSE and PackageHub
- internal registry for maintenance and development
- public cloud proxy registries
- customer local registries for own purposes
- customer local proxy registries
Public registry.suse.com

• **Purpose:**
  - Distribute container images:
    - as needed for SUSE products like SUSE CaaS Platform, SUSE Cloud Application Platform etc
    - for ISVs and developer to get base images to use for building their own container images
  - Container images will be pushed at product release time and as maintenance updates.
  - Updating container images:
    - Images will be updated by development team when fixing bugs or adding features
    - Images will be updated by maintenance team if one of the used packages gets an update
    - Image updates will be done from build service
Handling insecure registries

• Prior to v2:
  – Only registries using a trusted CA could be used
  – Handling other registries required manual work and hacks

• Starting from v3:
  – Easy management of registries not using a certificate at all
  – Easy management of registries using self-signed certificates
Velum: handling registry mirrors
System wide certificates

• Propagate customer certificates to all the nodes of the cluster
• Allows access to internal resources like:
  – Ceph storage
  – OpenStack endpoints
  – ...
• UI available both at deployment time and afterwards
System wide certificates

When you require a self-signed certificate, you can add it here, so it will be distributed to your cluster.

**Name**

**Certificate**
Paste the self-signed certificate to be added to the system certificate store here.
New Mirror

Mirror of
Select a registry
Create new registry

Name
URL

Save Cancel
New LDAP Connector

Name
Name shown to user when selecting a connector

Server
Host
Host name of LDAP server reachable from the cluster

Port
The port on which to connect to the host (e.g. StartTLS On: 389, StartTLS Off: 636)

StartTLS
On
When enabled use StartTLS otherwise TLS will be used

Certificate
Upload the certificate of the root CA that issued the LDAP server certificate

Choose File
No file chosen

Authentication
Anonymous
Use anonymous authentication to do initial user search

DN
**User Search**

**Identifying User Attribute**

Label of LDAP attribute users will enter to identify themselves (e.g. `username`)

**Base DN**

Base DN where users are located (e.g. `cn=users,dc=example,dc=com`)

**Filter**

Filter to specify type of user objects (e.g. `"(objectclass=person)"`)
Compute resources reservations

Every node of the Kubernetes cluster has a kubelet instance running. By default, the kubelet process will try to use all available resources on each node. This behaviour can lead to resource starvation for critical system services as well as for Kubernetes' own components.

To prevent this, it is possible to instruct kubelet to reserve a certain amount of resources for the host system and for Kubernetes core services on each node. The Kubernetes scheduler takes these limits into account by when deciding on which node to schedule a certain pod.

Kubernetes core services

This category includes processes such as:
- kubernetes API server
- kubernetes controller manager
- kubernetes scheduler
- kubelet
- kube-proxy
- Container runtime: Docker daemon, contained, CSR-O or runc

CPU

The amount of CPU units to reserve, as a decimal number or in “millicores” (e.g. 100m, 0.1). Leave empty for no CPU reservation.

Memory

The amount of memory to reserve, measured in bytes (e.g. 1024, 19, 1G). Leave empty for no memory reservation.

Ephemeral storage

The amount of ephemeral storage to reserve, measured in bytes (e.g. 1024, 19, 1G). Leave empty for no ephemeral storage reservation.
Auditing

Enable Kubernetes auditing feature to keep track of the activities that affected the cluster.

Log backend

All the audit events will be logged to the `/var/log/kube-apiserver/audit.log` file on the Kubernetes master node(s).

From there the user can use a data collector software like fluentd or logstash to collect and distribute the events.

Enable auditing

- Enabled
- Disabled

Max size

- 10

Maximum size in megabytes of the audit log file before it gets rotated

Max age

- 15

Maximum number of days to retain old audit log files

Max backup

- 20

Maximum number of audit log files to retain

Policy
Don’t panic!

- New container engine available: CRI-O
- Docker open-source engine is still part of v3. It’s the default choice
- It’s tech preview, but we want to make it fully supported with v4
- We don’t know yet when we will replace docker with CRI-O
Why CRI-O?

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

- Ships with Kubernetes 1.9.8
- Work already in the progress to update to latest 1.10 release
- One of the significant changes: DaemonSet, Deployment, ReplicaSet and StatefulSet have been promoted to the apps group, they are considered stable.
Notary

• Signing and verifying content on the SUSE Docker Registry
• By default, your docker client is very trusting
• Content Trust - check whether or not an image has been signed by a trusted authority each time you run `docker pull`
• `DOCKER_CONTENT_TRUST` : hardened mode
• `DOCKER_CONTENT_TRUST_SERVER` source of truth

• `DOCKER_CONTENT_TRUST_SERVER="https://notary.docker.io"` docker pull nginx:latest
Each ECS Host gets assigned to run the Docker Distribution service.
Access registry.suse.com

zypper ar https://download.opensuse.org/repositories/systemsmanagement:/SCC/openSUSE_Leap_42.2/systemsmanagement:SCC.repo
zypper in docker-ls
docker-ls repositories --registry https://registry.suse.com
requesting list . done
repositories:
- cap-beta/scf-acceptance-tests
- cap-beta/scf-acceptance-tests-brain
- cap-beta/scf-api
[…]
- pause
- pv-recycler-node
- scc/busybox
- scc/redis
- sles12-mariadb
- sles12-salt-api
- sles12-salt-master
- sles12-salt-minion
- sles12-velum
- sles12sp2
- sles12sp3
# SUSE CaaS Platform 2

**Public Cloud**

<table>
<thead>
<tr>
<th>Images available in Market Place</th>
<th>Ready to run images for Amazon AWS, Azure and Google Cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Bring your own subscription</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>On-demand</th>
<th>Discussion started with Google Cloud and Azure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Not available yet. Timeline: 3-6 months</td>
</tr>
</tbody>
</table>

| Federation | In discussion for upcoming releases |
Role Base Access Control (RBAC)

- In enterprise settings, access might be based on job function or role of the user
- Users authenticate themselves to the system
- (Some) Users can activate one or more roles for themselves
## SUSE CaaS Platform 2

### RBAC Examples

<table>
<thead>
<tr>
<th>Role</th>
<th>Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sys Admin</td>
<td>Operate the infrastructure&lt;br&gt;Block access to the infrastructure level&lt;br&gt;Allow developers to interact with Kubernetes</td>
</tr>
<tr>
<td>Developer</td>
<td>Full access for my team to manage the application&lt;br&gt;No access to other teams work&lt;br&gt;No access from other team to our work</td>
</tr>
<tr>
<td>Manager</td>
<td>Check the usage&lt;br&gt;Have an overview of resources</td>
</tr>
</tbody>
</table>
What is SUSE MicroOS?

- OS focused only on containers
  - Minimal image designed for one special Use Case
- Focused on large deployments
  - Reduced end-user interactions
- An always up-to-date Operating System
  - Safe way to update the system
  - Read-only root filesystem
    - Btrfs with snapshots and rollback for transactional updates
- https://en.opensuse.org/Kubic:MicroOS
Consolidated Benefits – One Slide
SUSE CaaS Platform simplifies and extends Kubernetes Container management for the enterprise

<table>
<thead>
<tr>
<th>Platform management</th>
<th>Application Services</th>
<th>APIs and Integrations</th>
<th>SUSE Micro OS</th>
<th>Enterprise Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Install</td>
<td>• SUSE Registry</td>
<td>• LDAP integration</td>
<td>• OCI-compliant container runtime</td>
<td>- integrated</td>
</tr>
<tr>
<td></td>
<td>• Secure SUSE base Images</td>
<td></td>
<td>• Transactional update/rollback</td>
<td>- tested</td>
</tr>
<tr>
<td>Configure</td>
<td>• Third party images</td>
<td>• Monitoring</td>
<td>• HA/DR</td>
<td>- maintained</td>
</tr>
<tr>
<td>Update</td>
<td>• Third party image scanning</td>
<td>• Package Mgt (Helm)</td>
<td>• SELinux</td>
<td></td>
</tr>
<tr>
<td>Scale</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Future plans

- Monitoring
- Package Mgt (Helm)
- Networking (Flannel/Cilium)
- Storage (SUSE Enterprise Storage, NFS)
- Dashboard (Velum)

Kubernetes
- Certified conformant

Enterprise Grade - integrated - tested - maintained
Container Management for the Enterprise
Choose SUSE CaaS Platform to...

**Achieve faster time to value**
- Enterprise-ready
- Industry leading technologies
- Complete package

**Simplify management and control**
- Efficient installation
- Easy scaling
- Update automation

**Maximize return on your investment**
- Flexible solution
- Designed for today and tomorrow
- Cloud service economics
How does it “Reduce Time to Market”

- **Complete Container Solution**
  - Deploy, Manage, Scale

- **Trusted images from SUSE Registry**
  - Security certifications, Up-to-date security patches

- **Self-healing**
  - Auto-start, Re-schedule, Replicate containers

- **High Availability**

- **Run Apps Everywhere**
  - On-premise, On-bare-metal, Private or Public cloud

- **Improve Security**

- **Faster Time to Value**
  - Simple to deploy, Open source, Production grade container orchestration

- **Improve Service Delivery**
  - Auto-start, Re-schedule, Replicate containers
How does it “Increase Operational Efficiency”

- **Reduce Manual Steps**
  - Automate – Scale services up/down, Select hosts for high availability, Connect storage/networking components

- **Reduce Development Costs**
  - Leverage tools for full container application lifecycle support

- **Maximize Service Availability**
  - Transactional updates
  - No disruption during upgrade

- **Avoid Security Issues**
  - Signed container images from SUSE Registry - Secure, Trusted source

- **Improve CI/CD**
  - Efficient resource utilization – Both people & tools.
Partners and Ecosystem

SUSE CaaS Platform
Ready TODAY!
SUSE CaaS Platform Partners
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.