Agenda

- Welcome
- What are we talking about today?
- SUSE Containers as a Service Platform (CaaS)
  - Why do you want to invest in Containers?
- SUSE Cloud Application Platform (CAP)
  - Cloud Foundry Introduction
- Wrap-Up
- Demo
Intro – What are we talking about today?
SUSE Delivers from Edge to Core to Cloud

Application Delivery

- Container Management: SUSE CaaS Platform
- Platform as a Service: SUSE Cloud Application Platform

Software-defined Infrastructure

- Private Cloud / IaaS: SUSE OpenStack Cloud
- Compute Virtual Machine & Container
- Storage: SUSE Enterprise Storage
- Networking: SDN and NFV
- Multimodal Operating System: SUSE Linux Enterprise Server

Physical Infrastructure: Multi-platform Servers, Switches, Storage

Services

- SUSE Global Services
- Consulting Services
- Select Services
- Premium Support Services
SUSE Delivers from Edge to Core to Cloud
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

**Best Kubernetes Operator Experience**

**Best Kubernetes Developer Experience**
So let’s look about the why
Build and Deliver Cloud Native Applications

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

Increase agility with Microservices

Smaller Codebase + Specialized Teams + Deconstructed Services

= Agile IT

= Opportunity!
Build and Deliver Cloud Native Applications

Containers enable Microservices model

Fast start/stop

Densification

In-place upgrades (we destroy and re-deploy)
SUSE Containers as a Service Platform
What are Containers?

A package/image that can be deployed anywhere (that’s running a Linux Kernel)

Developers create a layered image of their application that contains everything their application requires to run on top of a base OS layer

A container image will be able to be built once and run everywhere

Developers can focus on what they do best – development - opposed to dealing with packaging or library dependency management

Operations can deploy an entire app stack in seconds as they don’t need to worry about prerequisites
Why Containers?

• **To Fix Virtual Machines!**
  • Benefits:
    - Abstraction – Modularity of Infrastructure, Portable, Flexible Deployment
    - Isolation/Containment of Applications
    - Familiar concepts (O/S, applications)
    - Strong security
  • Drawbacks
    - Overhead
      • Consumes a lot of resources: memory, cpu penalties
      • Very slow to start (booting)
Goal: A Fenced Sandbox

• Build it using no more than the standard features of the Linux kernel
• Isolated group of processes on a Linux host
  - Manageable as a single entity

• Tools
  - Isolation – chroot, unshare, nsenter (partial sharing)
  - Filespace – mount, mount –bind
  - Control – cgroups

• Issues
  - Format - CRI-O (Kubernetes Container Runtime Interface, Open Container) vs Docker
  - Security
  - Monitoring inside
  - Troubleshooting inside
To Manage the Sandbox

• **Basic Services**
  - Host OS (SUSE MicroOS)

• **Orchestration**
  - Kubernetes

• **User Interface**
  - Horizon, Helm/Tiller

• **Building Sandboxes, Made Easy(er)**
  - Cloud Application Platform/Cloud Foundry
How Containers Ease App Development

Deploying a POC app using modern Ruby on Rails technology

Without Containers

*Isolated* development environment
Small team builds a POC
POC hand over to Dev/Ops teams
One of the following issues is likely:
  • POC doesn’t run in production/staging
  • Development environment not easy to setup (different OS, different tools, missing packages)

POC can’t be safely used in production or easily migrated to production

With Containers

Team creating POC can also deploy
POC deployed with containers
POC runs in a staging/production environment
New dev environments are easy to setup
Developers can continue using the preferred tools and OS

Happy Developers! Productive teams!
Old (monolithic) Way
A Better Way…
What is SUSE CaaS Platform?
What is SUSE CaaS Platform?

Speed application delivery to improve business agility

SUSE CaaS Platform is an enterprise class container management solution that enables IT and DevOps professionals to more easily deploy, manage, and scale container-based applications and services.
SUSE CaaS Platform

Orchestration (Kubernetes)  Services (e.g. Deployment Dashboard)

Persistent Storage (local disk, NFS, SES)  Networking (Flannel)  Registry  Security  Logging

Automation (Salt + cloud-init)
Configuration & Management of each node

Container Runtime & Packaging

SUSE Linux Enterprise MicroOS (Container Host OS)

(Physical) Infrastructure
Three Key Technology Components

SUSE CaaS Platform
SUSE MicroOS for Hosting Containers

A purpose built operating system designed for running containers and optimized for large deployments.

Contains everything you need for running containers in production

Key features include:
- An easy to manage/upgrade OS
- Easily setup/manage a cluster of nodes
- Use core SUSE Linux technologies, such as btrfs, RPM, autoyast
- Scalable - up to thousands of nodes
- Transactional/Atomic updates
Kubernetes for Container Orchestration

Helps you deploy and run large numbers of containers at scale in a production ready environment

Deploy
Manage
Scale
Clustered

Allows users to manage applications, not machines

- Complete container orchestration solution
- Open source
- Production grade
- Industry-leading container technology
- Designed by Google, now part of Cloud Native Computing Foundation
Kubernetes Deployment is Not Easy

Kubernetes is great for container orchestration but is notoriously hard to

• Set up/install
• Configure
• Update
• Manage
• Secure

SUSE CaaS Platform takes this pain away

What used to take hours for skilled engineers can be done in minutes by junior administrators
SUSE CaaS Platform

Dashboard
Admin Node

Kubernetes Cluster

- k8s Master - k8s Worker - ...
- k8s Master - k8s Worker - ...
- k8s Worker - k8s Worker - ...
- ... - ... - ...

...
** SUSE CaaS Platform Roadmap **

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

** Container Host OS **
- Codebase: SUSE Linux Enterprise 12 SP3
- Tested with 150 nodes
- Enhance administration dashboard
- Container engine alternative (cri-o) (tech preview)
- Toolchain module

** Orchestration **
- Kubernetes 1.9; Docker 17.09
- Ingress Controller integration (software)
- Network options (CNI using flannel)
- Cloud integration for Storage and network (CPI)

** Ecosystem **
- Proxy registry
- Trusted container images
- Doc rework, best practices
- SUSE Container Certification Program

** Orchestration **
- Kubernetes version update
- Network options (Cilium as first plugin)

** Ecosystem **
- VMware storage
- Overlap support
- Hosting SUSE OpenStack Cloud and SUSE Enterprise Storage

** Container Host OS **
- Codebase: SUSE Linux Enterprise 12 SP3
- Supported Container engine alternative (cri-o)
- Support customer certificate authority

** Orchestration **
- Kubernetes 1.10
- Authentication integration: LDAP, AD

** Ecosystem **
- Proxy registry
- Trusted container images
- Doc rework, best practices
- SUSE Container Certification Program

** Orchestration **
- Kubernetes version update
- Network options (Kuryr plugin)
- GPU Integration
- Unattended installation

** Container Host OS **
- Codebase: SUSE Linux Enterprise 15
- Container isolated via virtualization (Kata containers)
- Monitoring (Prometheus)

** Orchestration **
- Kubernetes version updates
- Kubeadm based administration

** Container Host OS **
- Codebase: SUSE Linux Enterprise 15 SP1+

** Orchestration **
- Kubernetes version updates

** Ecosystem **
- VMware storage
- Overlap support
- Hosting SUSE OpenStack Cloud and SUSE Enterprise Storage

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

** Information is forward looking and subject to change at any time. **
SUSE Cloud Application Platform
SUSE Cloud Application Platform

Bringing Advanced Cloud Foundry Productivity to Modern Kubernetes Infrastructure
SUSE Cloud Application Platform
Cloud Native Developer Experience in Kubernetes

• Containerized
• Easy for Kubernetes users
• Small memory footprint
• 100% open source
• Enterprise grade Linux

An ideal solution for organizations pursuing an engineering-led approach to application delivery transformation
Kubernetes is Aimed at Operators, not Developers

**SUSE Cloud Application Platform** uses the Cloud Foundry Application Runtime to add functionality for developers:

- One step command to containerize, deploy, and manage an application
- Automatically identifies and pulls in language libraries, frameworks, and other dependencies via buildpacks
- Open source service brokers automatically create and bind services to applications
- Automates application lifecycle management by assigning appropriate resources, managing routing, load balancing, scaling, and more
What is Cloud Foundry?

- Platform as a Service
- Cloud native application platform with enhanced features to support enriched development and deployment
  - Integrated routing with advanced route handling
  - Consolidating logging for both system and application levels
  - Enhanced RBAC with organization and space management model
  - Service integration for applications
- Originally open sourced by VMWare in 2011
- https://cloudfoundry.org
Pizza as a Service 2.0

http://www.paulkerrison.co.uk

 Tradition On-Premises (legacy)
 - Conversation
 - Friends
 - Beer
 - Pizza
 - Fire
 - Oven
 - Electric / Gas

 Infrastructure as a Service (IaaS)
 - Conversation
 - Friends
 - Beer
 - Pizza
 - Fire
 - Oven
 - Electric / Gas

 Containers as a Service (CaaS)
 - Conversation
 - Friends
 - Beer
 - Pizza
 - Fire
 - Oven
 - Electric / Gas

 Platform as a Service (PaaS)
 - Conversation
 - Friends
 - Beer
 - Pizza
 - Fire
 - Oven
 - Electric / Gas

 Function as a Service (FaaS)
 - Conversation
 - Friends
 - Beer
 - Pizza
 - Fire
 - Oven
 - Electric / Gas

 Software as a Service (SaaS)
 - Conversation
 - Friends
 - Beer
 - Pizza
 - Fire
 - Oven
 - Electric / Gas

 Homemade
 Communal Kitchen
 Bring Your Own
 Takeaway
 Restaurant
 Party

 Blue: You Manage
 Green: Vendor Manages

 Configuration
 Functions
 Scaling...
 Runtime
 OS
 Virtualisation
 Hardware
Code + Buildpack + Stack = App
Cloud Foundry on Kubernetes Architecture
Code + Buildpack + Stack = App
Applications are not Containers

- How do I …
  - build it (once or 1000 times)
  - connect it to services
  - scale it
  - route to it
  - manage its lifecycle

- Cloud Foundry understands the world from the application point of view
SUSE Cloud Application Platform

Containerized Cloud Foundry, Built on SUSE Linux Enterprise

BYO Containers

SUSE Cloud Foundry (containerized)

Control plane: SUSE Linux Enterprise containers

Application execution: SUSE Linux Enterprise containers

Broker
SLE

UAA
SLE

Volume
SLE

Router
SLE

CC API
SLE

Logging
SLE

Workloads
SLE

Workloads
SLE

Workloads
SLE

Workloads
SLE

Workloads
SLE

Workloads
SLE

Public Cloud K8s-as-a-Service
EKS, AKS (GKE, AliCloud)

SUSE CaaS Platform
Kubernetes + MicroOS

Public Cloud
Amazon, Google, Microsoft

Private Cloud
SUSE OpenStack Cloud

Virtual Machines
KVM, Xen, VMware, Microsoft

Bare Metal
SLES supported hardware
SUSE Cloud Application Platform

1.0
- Cloud Foundry: built from upstream CF Bosh release (cf-release v280)
- SUSE Linux Enterprise stemcells
- SUSE stacks compatible with CF buildpacks
- Kubernetes 1.7 (SUSE CaaS Platform 2)
- Stratos-UI 1.0 web console
- HA pod configuration
- Volume Service (NFS)
- MySQL service broker

1.1
- Cloud Foundry 2018 certification
- Supported on Azure Kubernetes Service (AKS)
- Backup/restore cf plugin
- PostgreSQL service broker
- CF updates - every point release
  - component releases
  - buildpacks

1.2
- Supported on Amazon EKS
- App-AutoScaler: Dynamic scaling based on performance metrics, time/date (with IBM*).
- OSBAPI Broker for Helm: MySQL, PostgreSQL, Redis, or MongoDB instances on Kubernetes (experimental).
- Stratos UI v2: service management, Angular updates
- Support for Azure Load Balancer.

1.3
- CredHub: credential management
- Brokers: Minibroker (Helm), AWS, Azure OSBA
- Stratos UI: Kubernetes endpoint plugin and metrics reporting.

2.0
- Eirini: Kubernetes-native app scheduling.
- CF Containerization: integration of upstream work on CF Operator
- Supported on Google Kubernetes Engine (GKE)
- CF version updates and 2019 recertification
- CF-Abacus: Metering and aggregation
- Istio & Envoy in CF (pending upstream readiness)
- Minibroker improvements
- App Versioning (depending on v3 API readiness)
- DRDB persistent store support

2.x
- Serverless: Function-as-a-Service in Cloud Foundry and Kubernetes (possibly via Knative)
- Istio & Envoy in CF
- Integrated online IDE (e.g. Eclipse Che)
- Windows-native .NET
- Integrated CI/CD (Concourse, Jenkins, other)
- Git hosting: Github Enterprise, Gitlab, or Gitea
- CF App SSO via Oauth2/UA: Route service for limiting access to CF-hosted applications.
- Better SNI/certificate management
- IBM zSystem support

* Information is forward looking and subject to change at any time.
Wrap-up
SUSE Application Delivery Platforms
Support multiple approaches to speed application delivery

Cloud Foundry
Productivity

Kubernetes
Flexibility

SUSE Cloud Application Platform
• Accelerate end-to-end application development and delivery at scale
• Maximize productivity with abstractions, patterns, and full lifecycle automation

SUSE CaaS Platform
• Simplify deployment and management of containers and containerized applications
Demo!
Thank you for joining us today!
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.
Extra Slides
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
Welcome! You have signed up successfully.

Initial CaaS Platform Configuration

Generic settings

Internal Dashboard FQDN/IP

Cluster services

- Install Tiller (Helm's server component)

Overlay network settings

Proxy settings
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>1d86c6a48bec412dab08459e97bda0</td>
<td></td>
</tr>
<tr>
<td>3c2a11d3c4924689c3d49bc408b2353</td>
<td>Accept Node</td>
</tr>
<tr>
<td>58297f4e1f8492db55a7e568f18df6f</td>
<td>Accept Node</td>
</tr>
<tr>
<td>74b80f9a63d4327c846f8ec1499bde87</td>
<td>Accept Node</td>
</tr>
<tr>
<td>b2ed30c0b04e8f98ec5726a87475d6</td>
<td>Accept Node</td>
</tr>
<tr>
<td>f39a8d46f1426d8d135b9e814a44a</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>1d86c6a48bac412bad0b8459e0977bc9a</td>
<td>Acceptance in progress</td>
</tr>
<tr>
<td>3c24113d3c4924689b9c3d49bc08b2353</td>
<td>Acceptance in progress</td>
</tr>
<tr>
<td>56297f40ef8492db505a7e568f16d6ff</td>
<td>Acceptance in progress</td>
</tr>
<tr>
<td>74b6c09a5d3a427c8466fce14998d87</td>
<td>Acceptance in progress</td>
</tr>
<tr>
<td>b2eddafecb084e0698e5726a87475d6</td>
<td>Acceptance in progress</td>
</tr>
<tr>
<td>f39af8d46f7426d8d1358be1844aa</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>58297f40e1f8492db05a7e658f18d6df</td>
<td>linux</td>
<td>Master</td>
</tr>
<tr>
<td>74bbdefda63d427c846f660f199be87</td>
<td>linux</td>
<td>Master</td>
</tr>
<tr>
<td>b2eddefeb084e068bec572e87475d6</td>
<td>linux</td>
<td>Master</td>
</tr>
<tr>
<td>f3dbadb68138fae2d13588b81444aa</td>
<td>prd-node1.home.decanha-knight.net</td>
<td>Master</td>
</tr>
<tr>
<td>1d86ca48ac412bad0088459e097b7c4a</td>
<td>prd-master.home.decanha-knight.net</td>
<td>Master</td>
</tr>
<tr>
<td>3b2411d3dc924669b9c0d94bc04082353</td>
<td>linux</td>
<td>Master</td>
</tr>
</tbody>
</table>

#### Pending Nodes

You currently have no nodes to be accepted for bootstrapping.
**Cluster Status**

### Summary

<table>
<thead>
<tr>
<th>Status</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total nodes</td>
<td>6</td>
</tr>
<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>58297f40e1f8492d85a7e568f18d6df</td>
<td>linux</td>
<td></td>
</tr>
<tr>
<td>✔️</td>
<td>748b09f9ed3427c846f8c1499d6e87</td>
<td>linux</td>
<td></td>
</tr>
<tr>
<td>✔️</td>
<td>b2edaf6ecb084e6f8ac5726a87475d6</td>
<td>linux</td>
<td></td>
</tr>
<tr>
<td>✔️</td>
<td>f33f8b846f14268d135b8e14a44aa</td>
<td>prd-node1:home.decanha-knight.net</td>
<td></td>
</tr>
<tr>
<td>✔️</td>
<td>1d8ed648bb0412bad0b8459e097bca</td>
<td>prd-master:home.decanha-knight.net</td>
<td>master</td>
</tr>
<tr>
<td>✔️</td>
<td>3b2411d3e492e689bc0d49b408b2353</td>
<td>linux</td>
<td></td>
</tr>
</tbody>
</table>

### Pending Nodes

You currently have no nodes to be accepted for bootstrapping.
Cluster Status

Summary

<table>
<thead>
<tr>
<th>Node Type</th>
<th>Count</th>
<th>Updates</th>
<th>Manual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total nodes</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Master nodes</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>New nodes</td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td># of nodes w/ outdated software</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>58297f4b1f8492db05a7e568f18d6df</td>
<td>linux</td>
<td>worker</td>
</tr>
<tr>
<td>✔</td>
<td>74b8c0f94e3427c6461ece1499bde87</td>
<td>linux</td>
<td>worker</td>
</tr>
<tr>
<td>✔</td>
<td>b2ed8df5c084e698ac5726a784750d6</td>
<td>linux</td>
<td>worker</td>
</tr>
<tr>
<td>✔</td>
<td>f33affb046f1526d1358be871a4a4aa</td>
<td>prd-node1.home.decanha-knight.net</td>
<td>worker</td>
</tr>
<tr>
<td>✔</td>
<td>1d86c6a48bac4f25da0e0859e097badc</td>
<td>prd-master.home.decanha-knight.net</td>
<td>master</td>
</tr>
<tr>
<td>✔</td>
<td>3b2411d3a4924689bc0d49b408b2353</td>
<td>linux</td>
<td>worker</td>
</tr>
</tbody>
</table>

Pending Nodes

You currently have no nodes to be accepted for bootstrapping.
Download your kubectl file

You will see a download dialog that will allow you to download your kubectl 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/kubernetes 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 kubectl file