We adapt. You succeed.

SUSE CaaS Platform

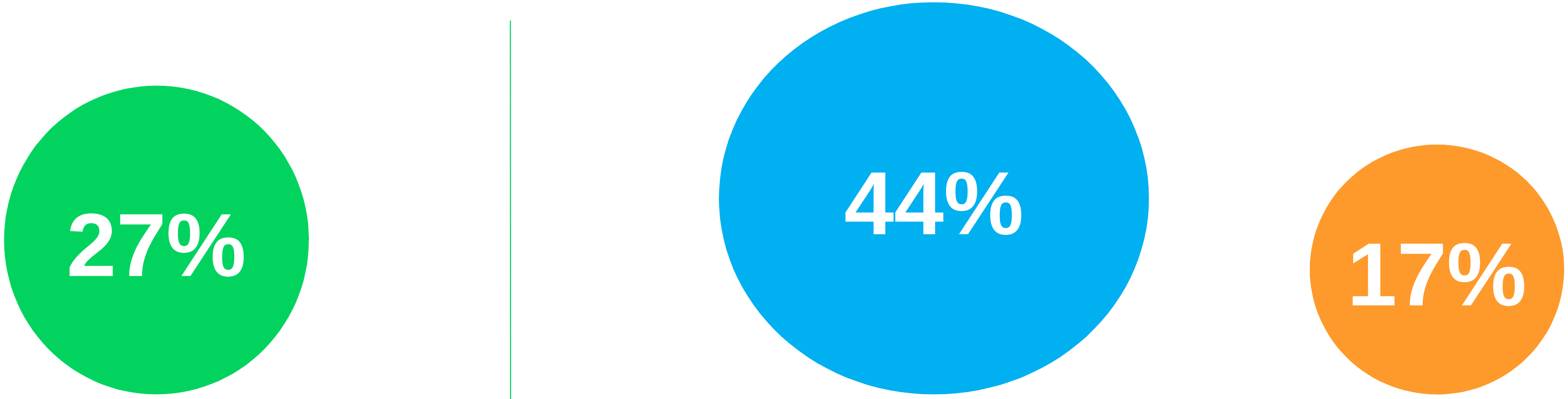
OSAD Atix München, 16.10.2018

Ralf Dannert

Systems Engineer

ralf.dannert@suse.com

Enterprises Want Container Workloads in Production



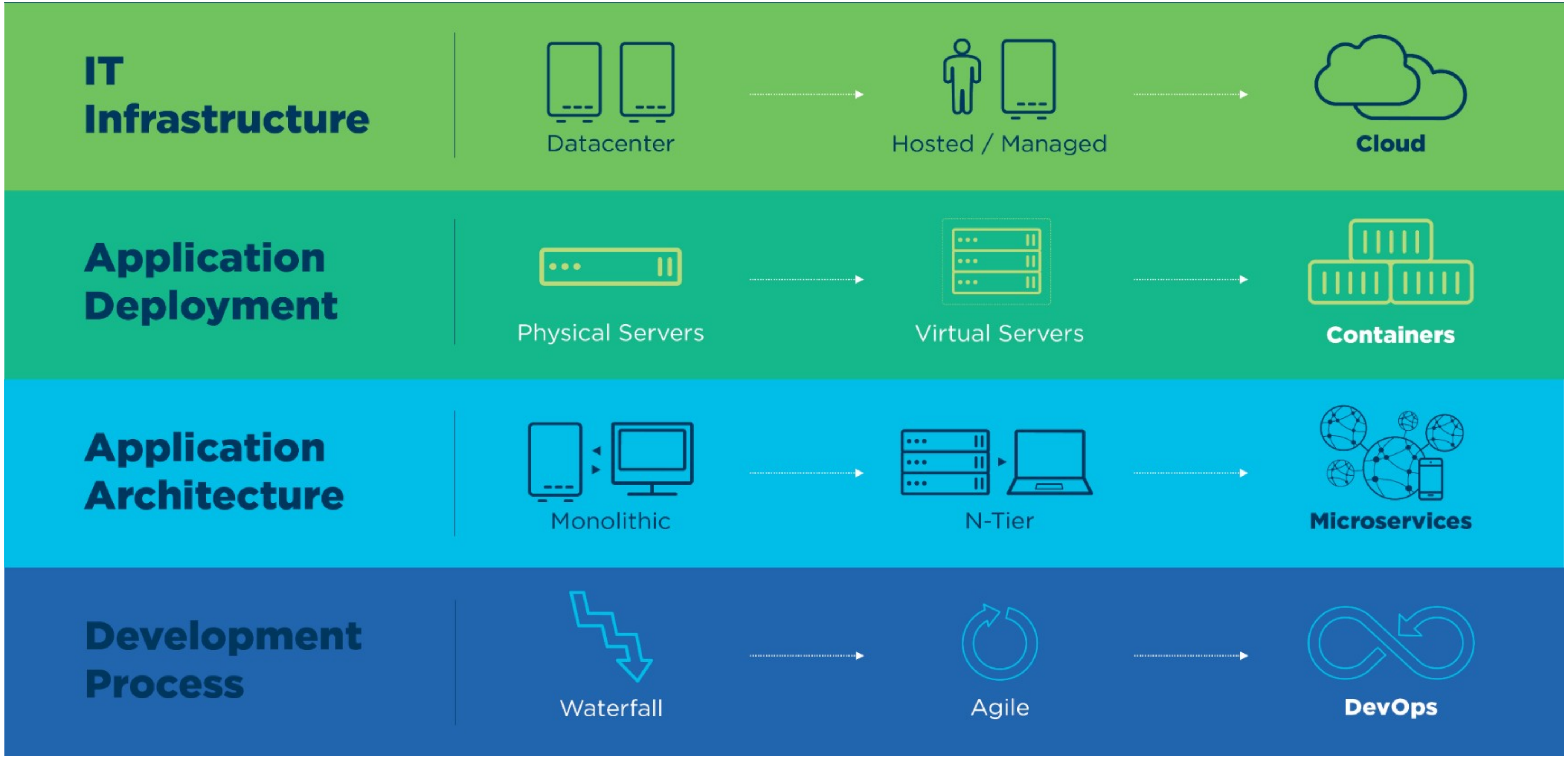
Running
Today

Planning to Run
Within 1 year

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

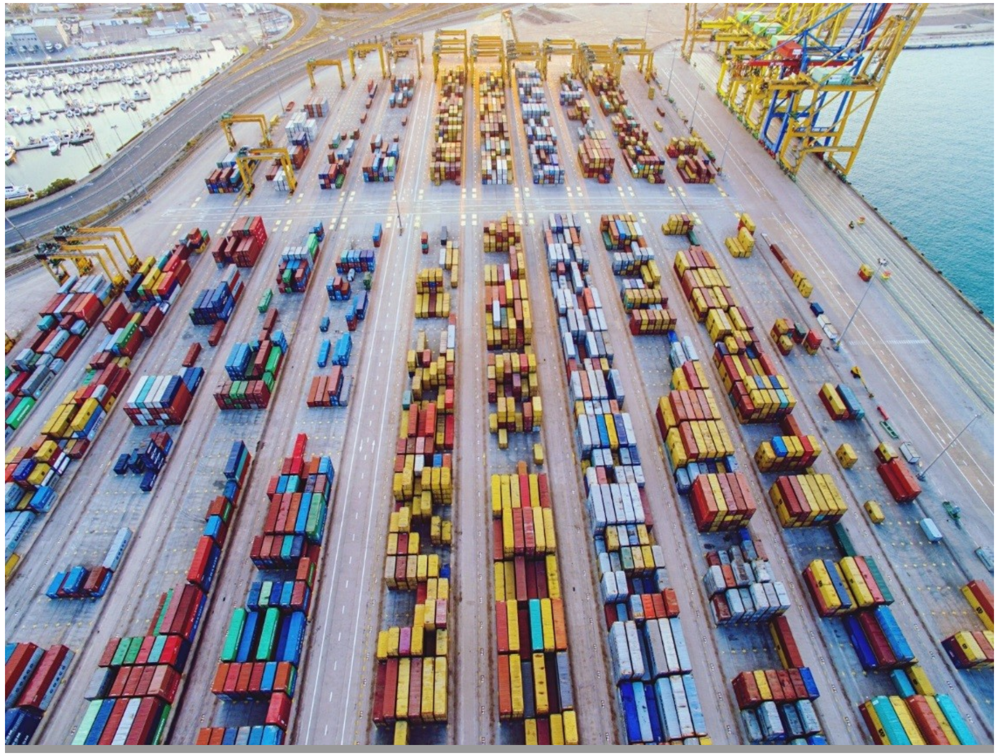
- Scheduling
- Service discovery

Performance and availability

- Scaling
- Load balancing
- Self-healing
- Monitoring

Maintenance

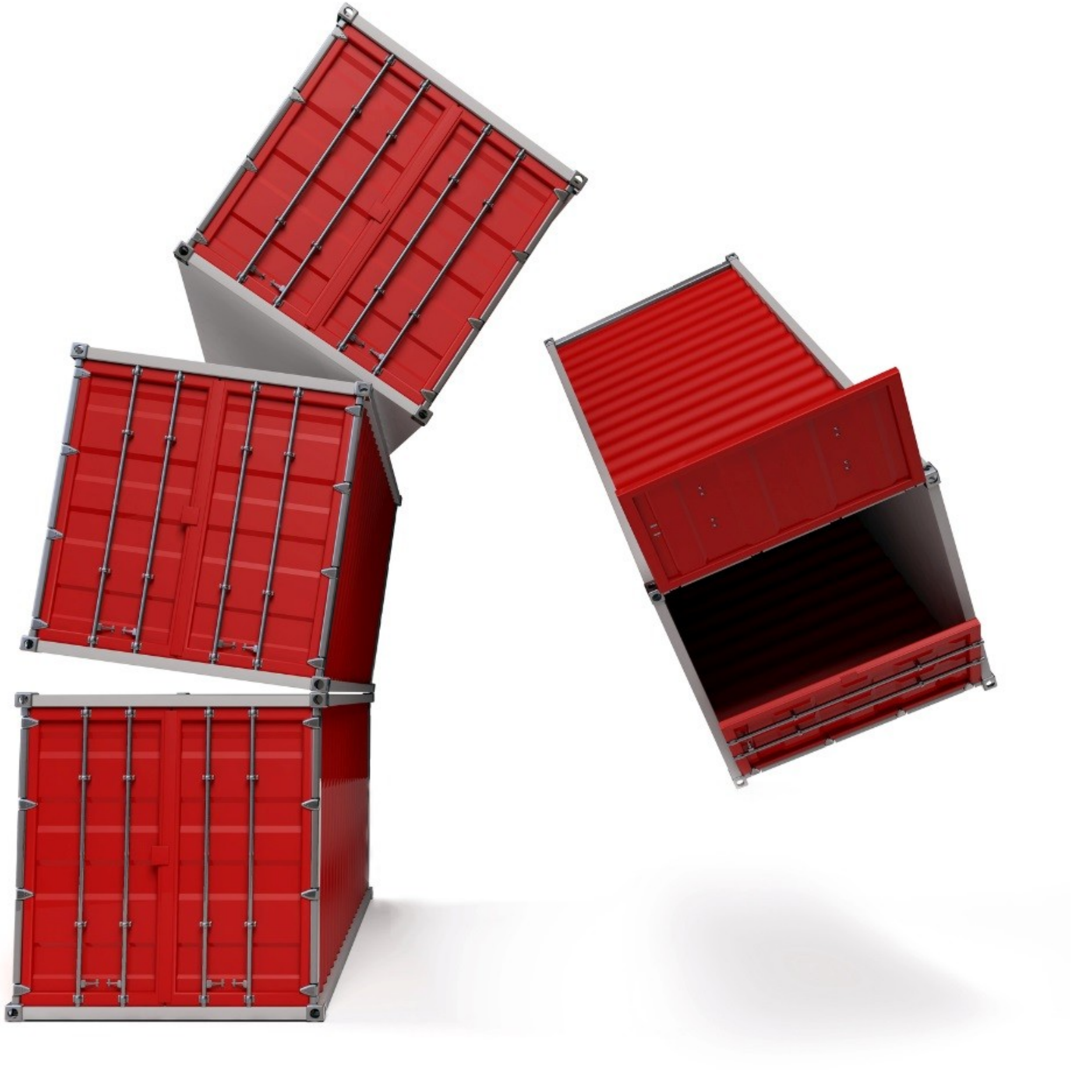
- Rollout
- Rollback



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.”

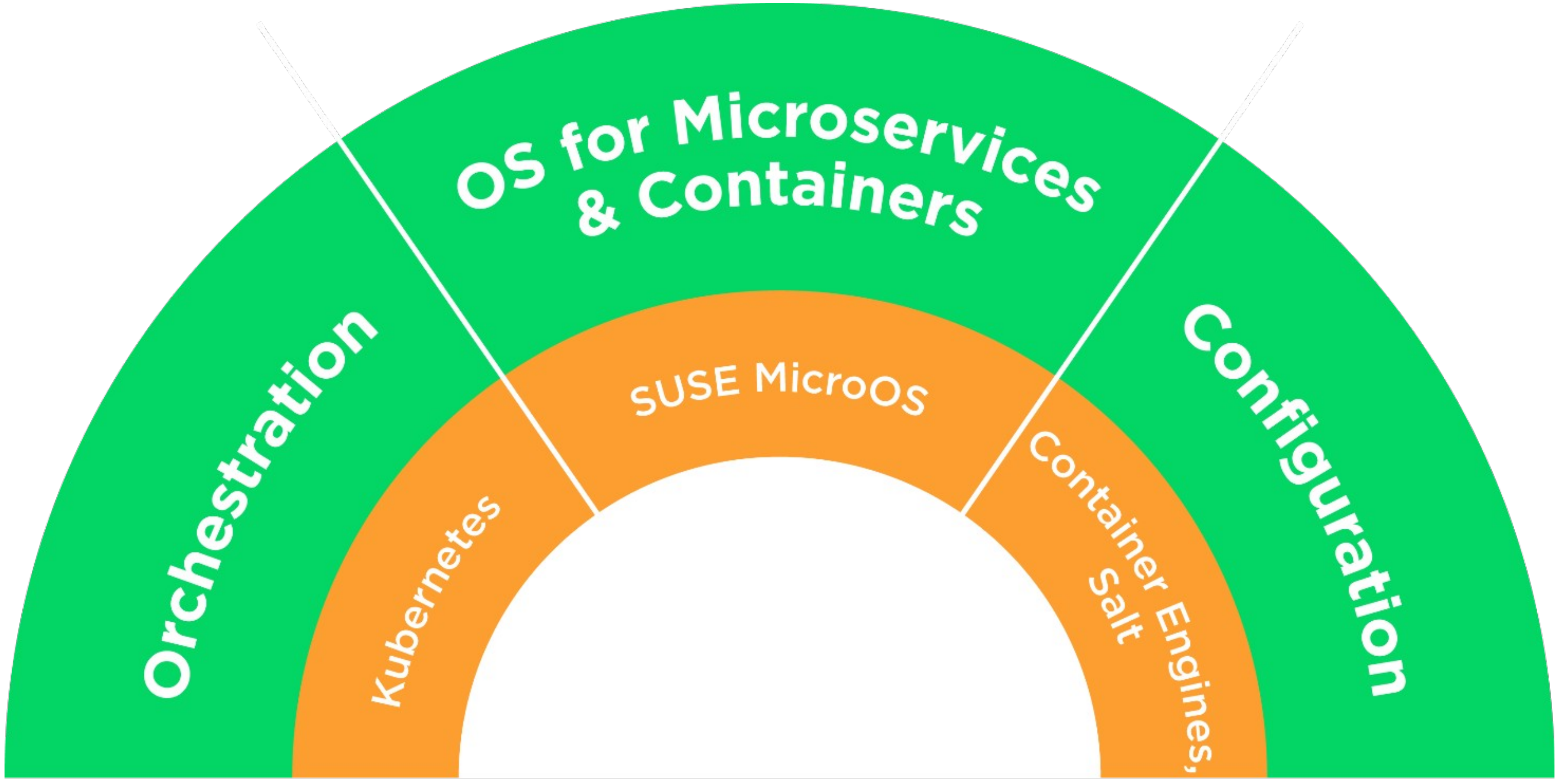
Container Services in the Public Cloud. Gartner 2017

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.

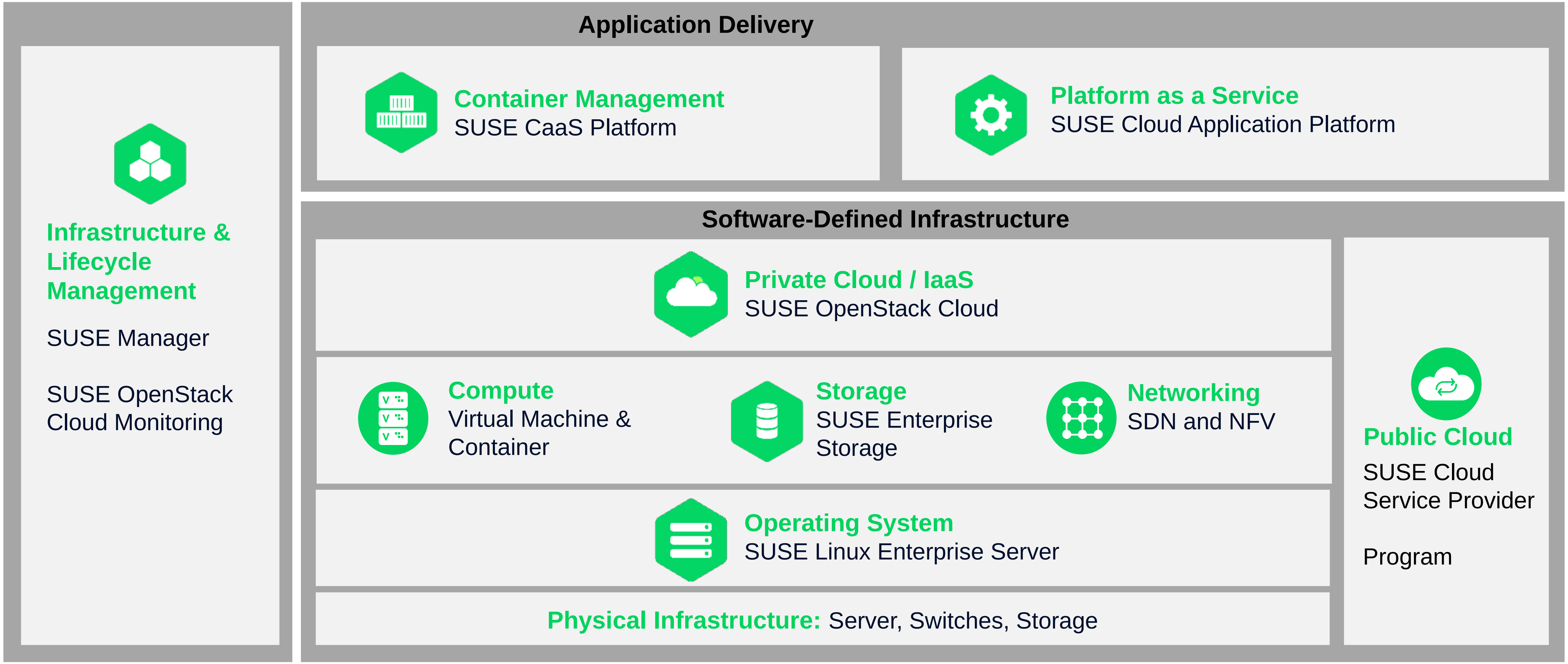
3 Key Technology Components



SUSE CaaS Platform

SUSE CaaS Platform

A Key Component of the SUSE Approach to IT Transformation



Infrastructure & Lifecycle Management

SUSE Manager

SUSE OpenStack Cloud Monitoring

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

Operating System
SUSE Linux Enterprise Server

Physical Infrastructure: Server, Switches, Storage

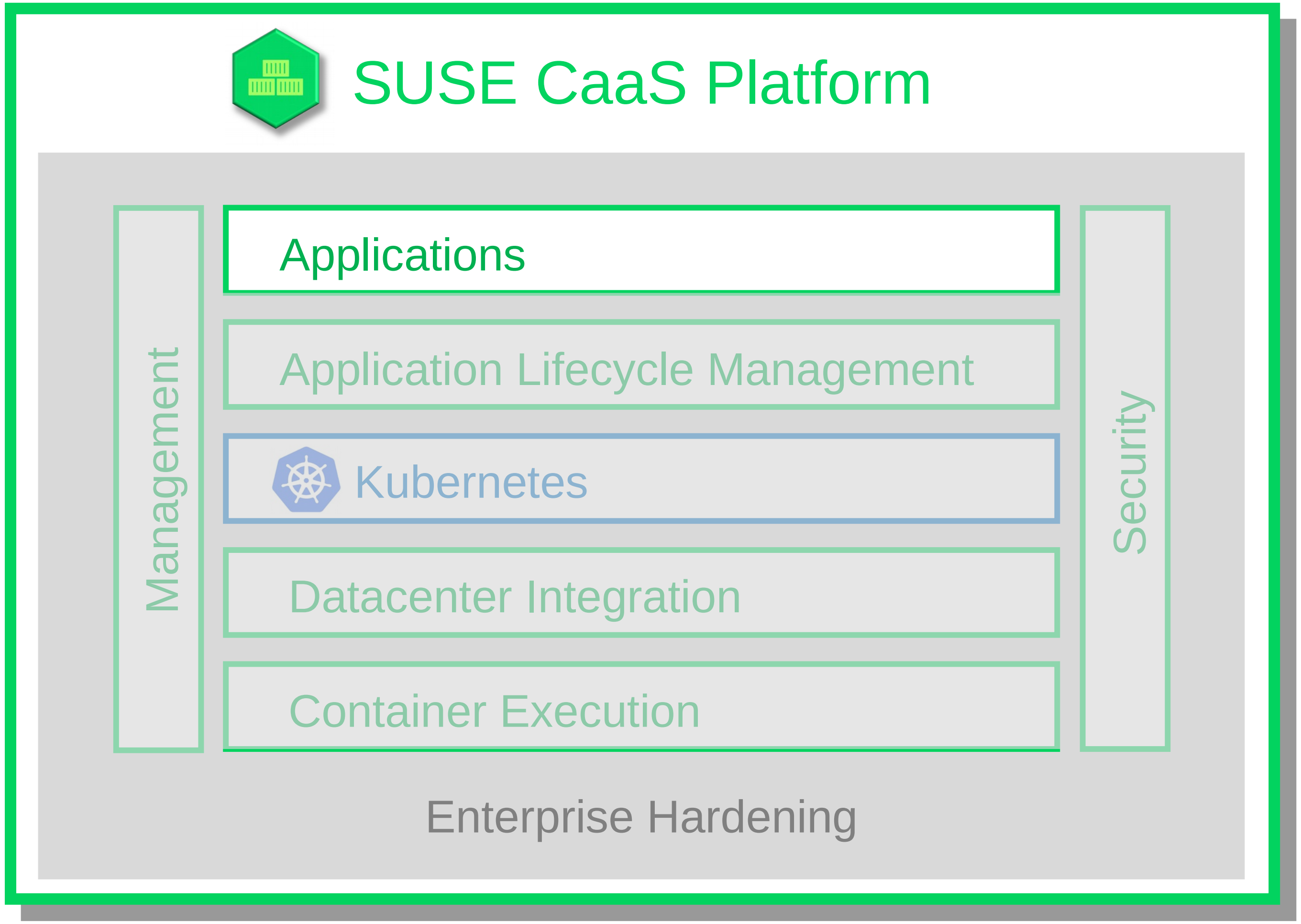
Public Cloud
SUSE Cloud Service Provider

Program

SUSE CaaS Platform
Key Features

SUSE CaaS Platform simplifies and extends Kubernetes

Container management for the enterprise

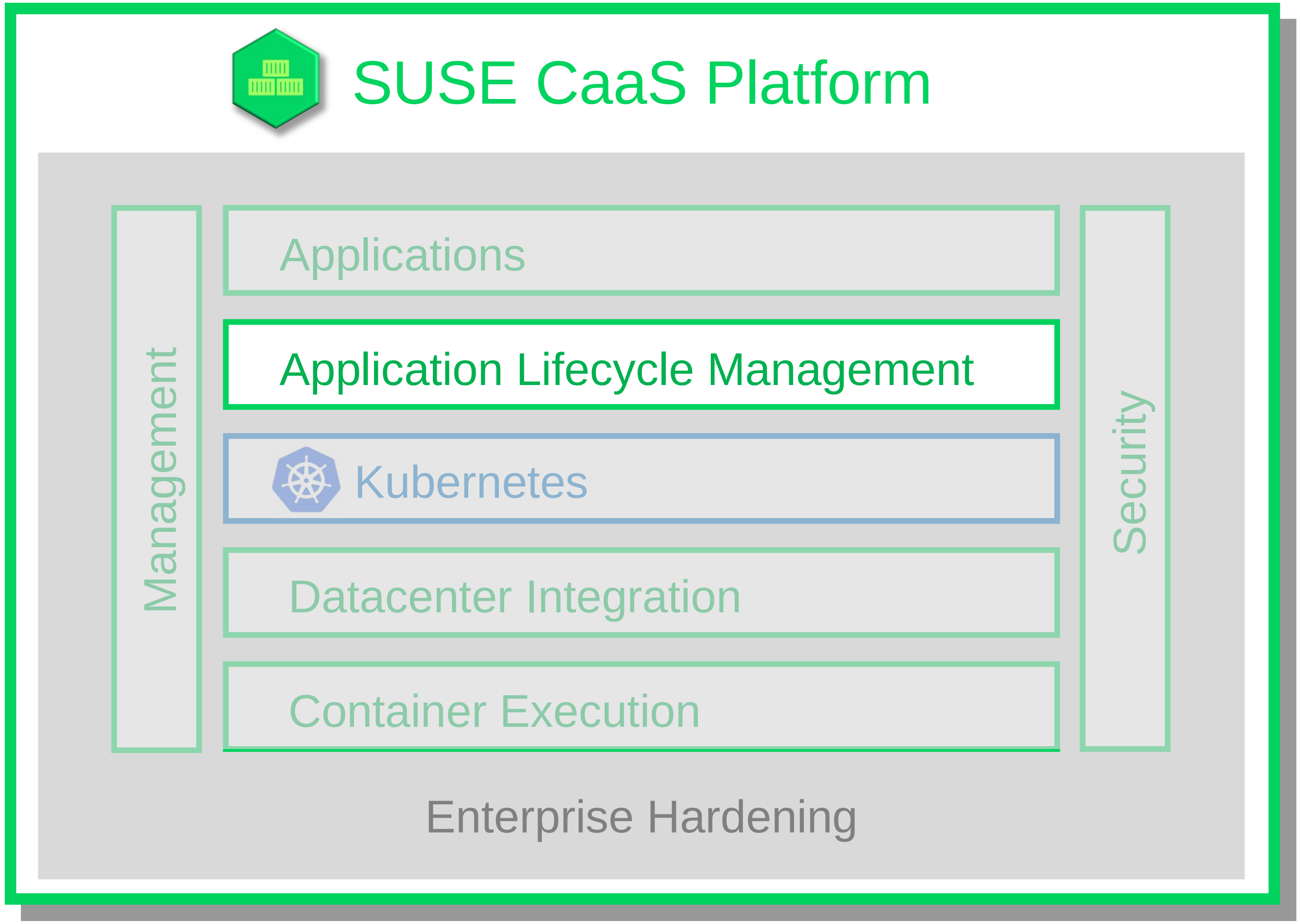


Applications

- SUSE Linux container base images

SUSE CaaS Platform simplifies and extends Kubernetes

Container management for the enterprise

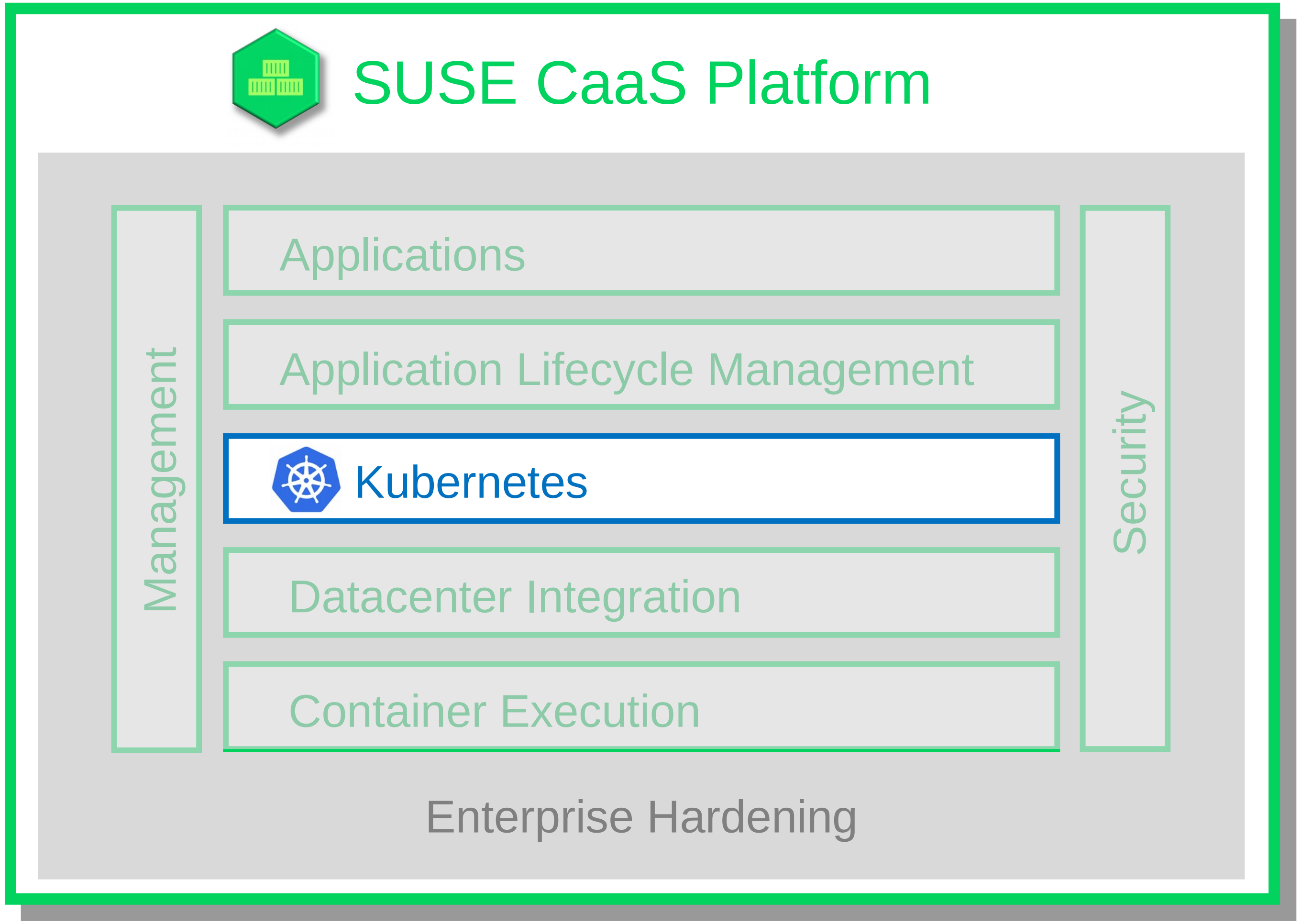


Application Lifecycle Management

- 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

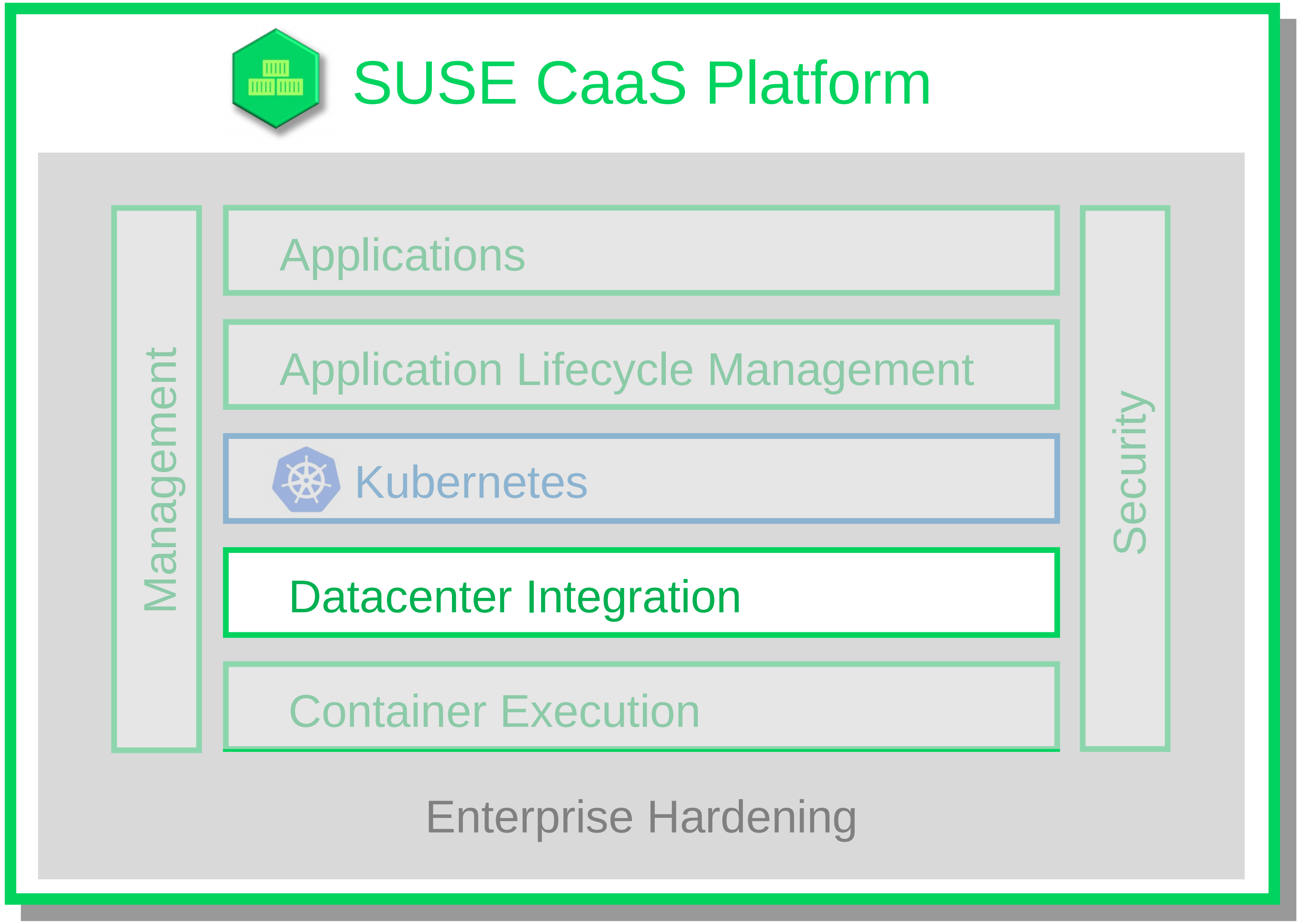


Kubernetes

- CNCF certified distribution

SUSE CaaS Platform simplifies and extends Kubernetes

Container management for the enterprise

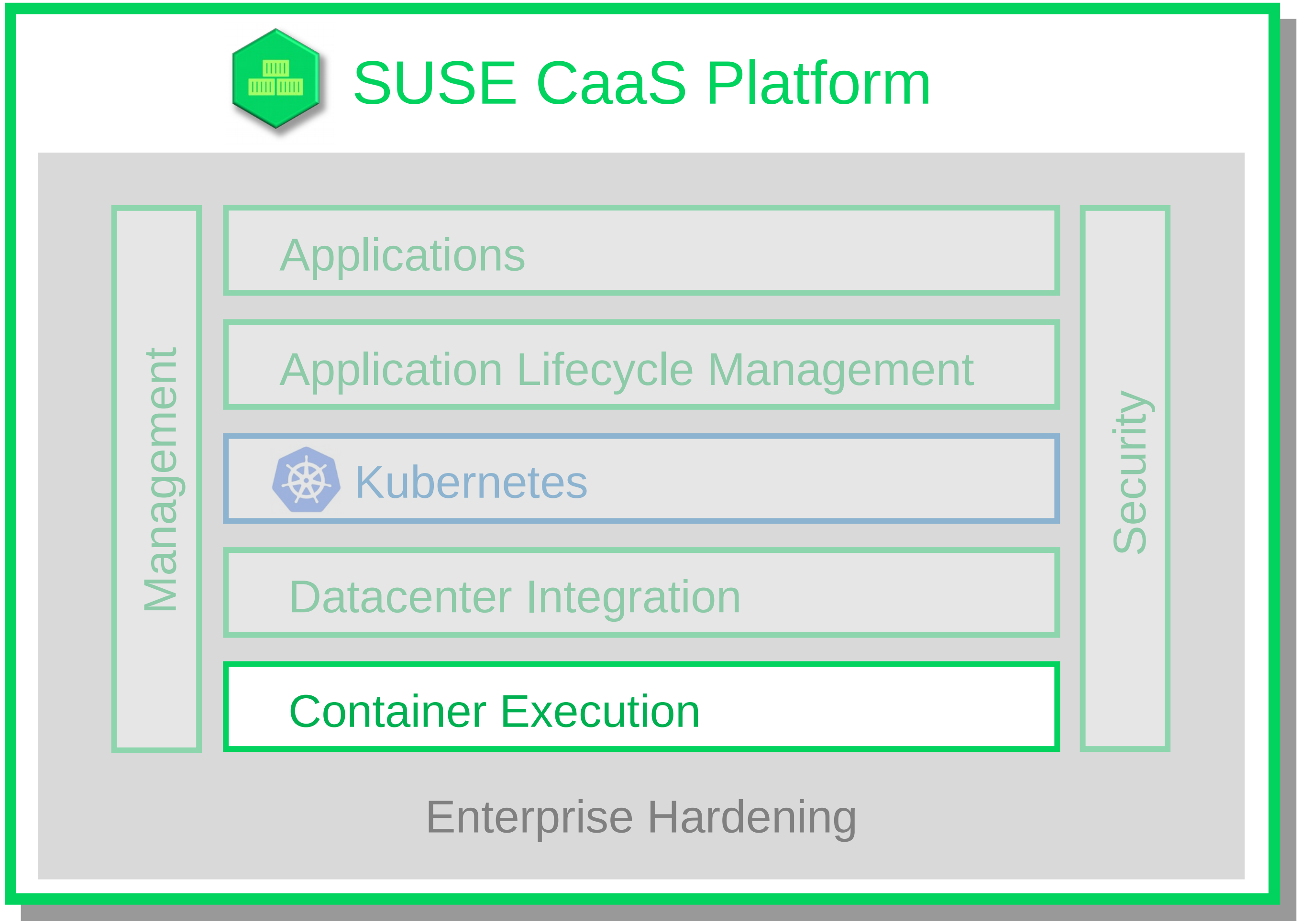


Datacenter Integration

- 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

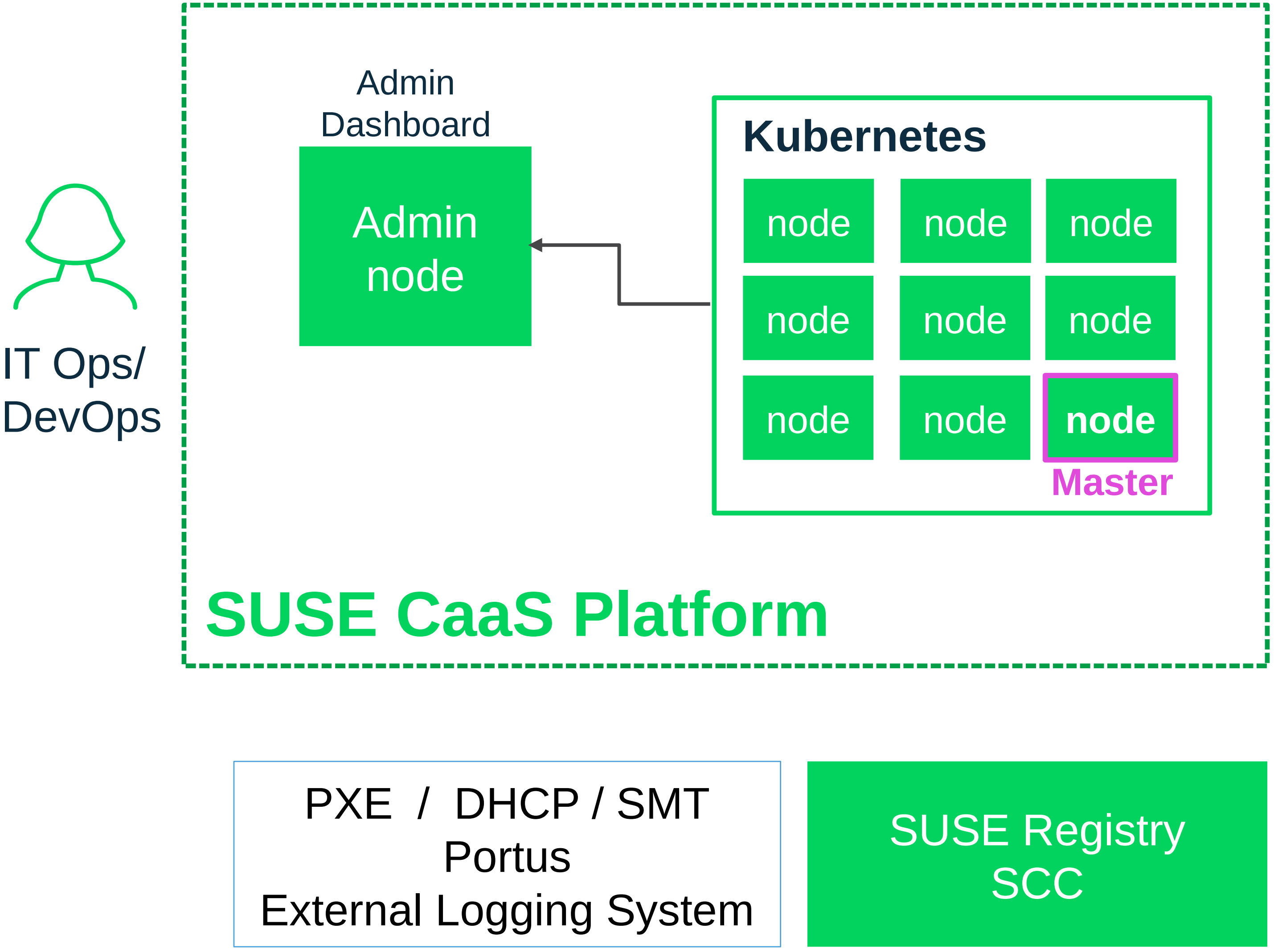


Container Execution

- Container runtime
- Image registries
- Container OS

SUSE CaaS Platform Technology Overview

SUSE CaaS Platform Deployment - Setup Infrastructure

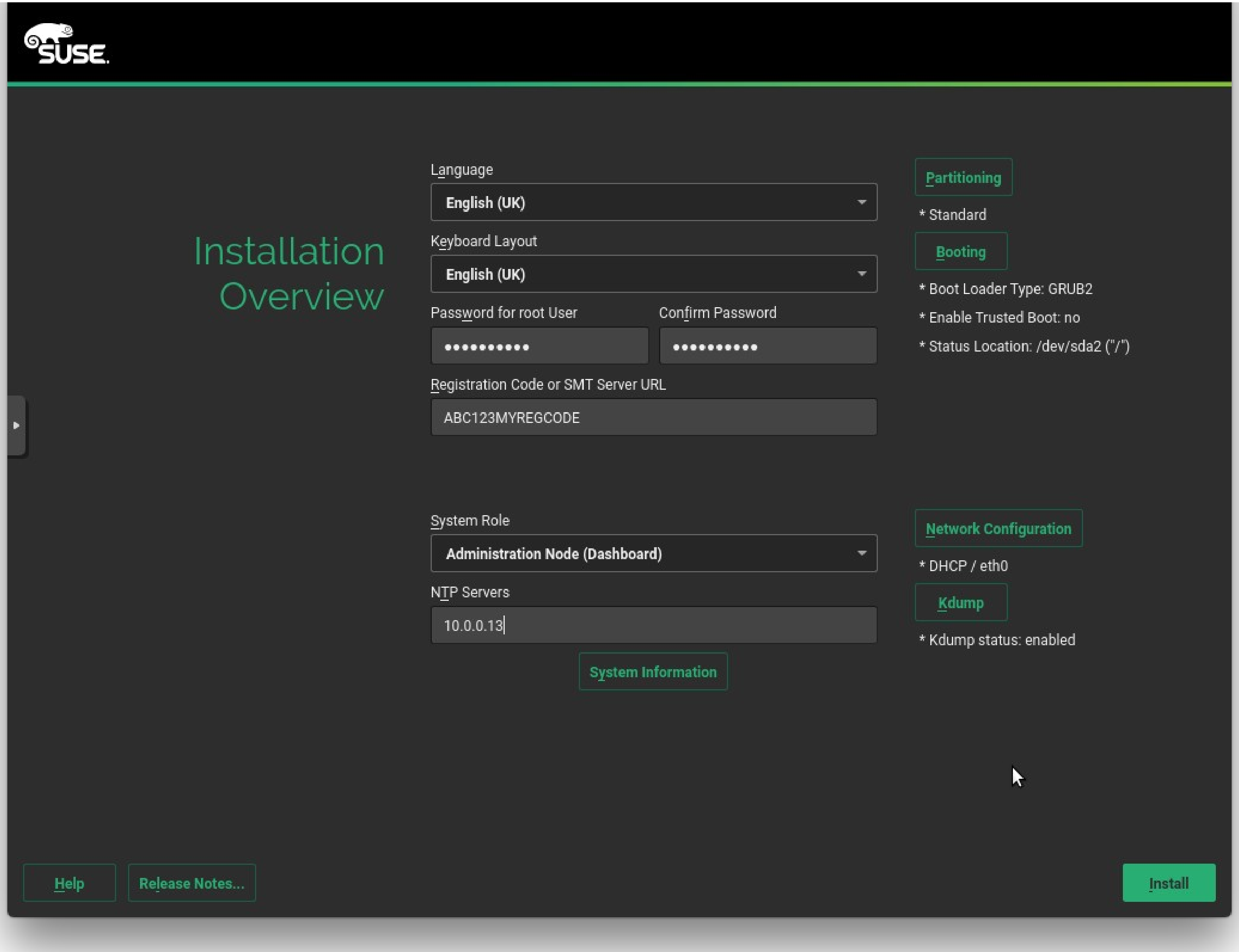


1 **Install Admin node**
MicroOS one step installation
Create AutoYaST profile
Set up Admin Dashboard

2 **Connect to Admin Dashboard**

3 **Deploy Nodes**
Uses AutoYaST profile

4 **Configure Cluster**
Set up kubernetes, etcd, flannel,...



Installation Overview

Language: English (UK)

Keyboard Layout: English (UK)

Password for root User: ••••••••••

Confirm Password: ••••••••••

Registration Code or SMT Server URL: ABC123MYREGCODE

System Role: Administration Node (Dashboard)

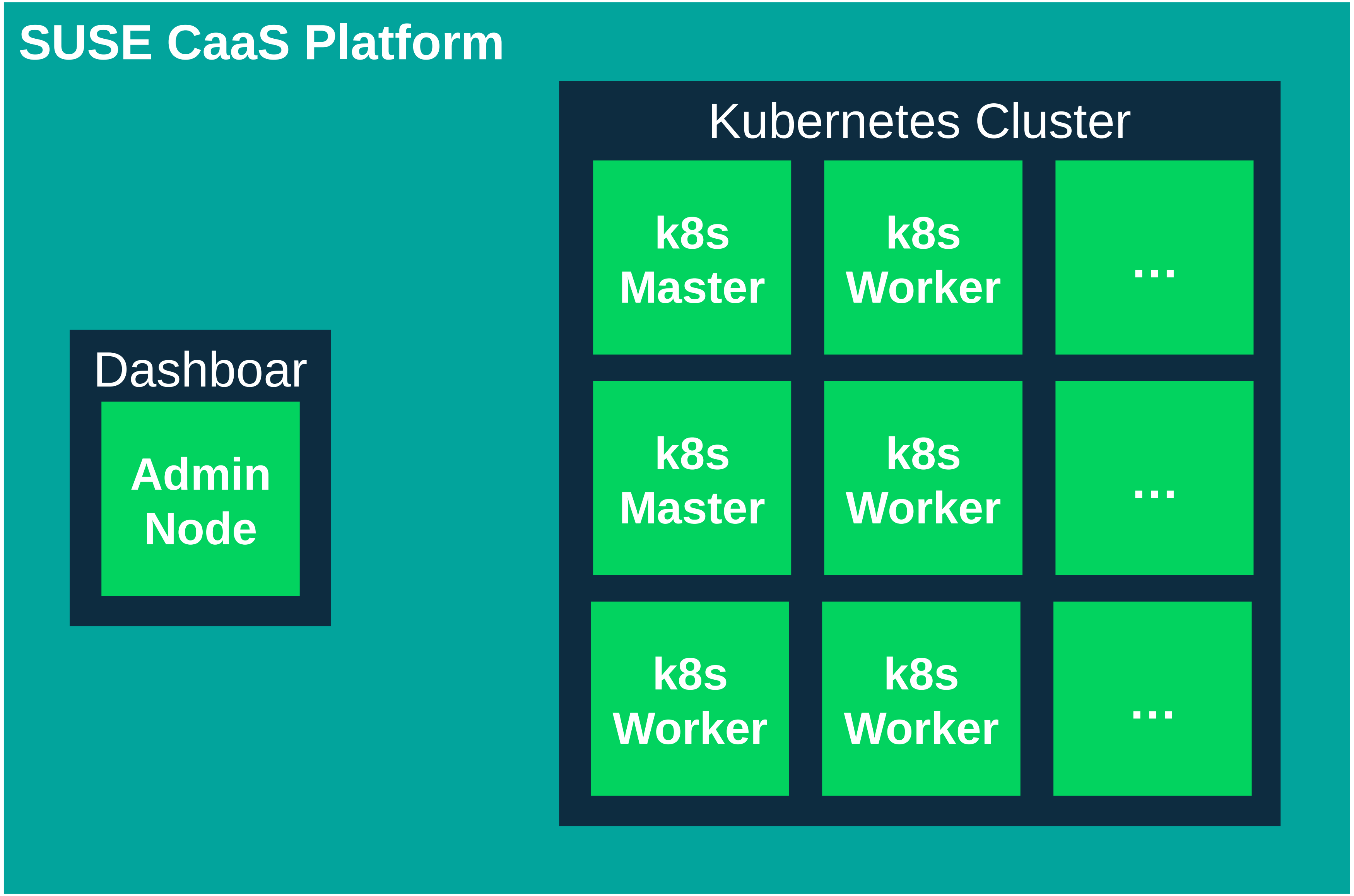
NTP Servers: 10.0.0.13

Partitioning
- * Standard

Booting
- * Boot Loader Type: GRUB2
- * Enable Trusted Boot: no
- * Status Location: /dev/sda2 ("/")

Network Configuration
- * DHCP / eth0

Kdump
- * Kdump status: enabled



SUSE CaaS Platform

Dashboar

Admin Node

Kubernetes Cluster

k8s Master

k8s Worker

…

k8s Master

k8s Worker

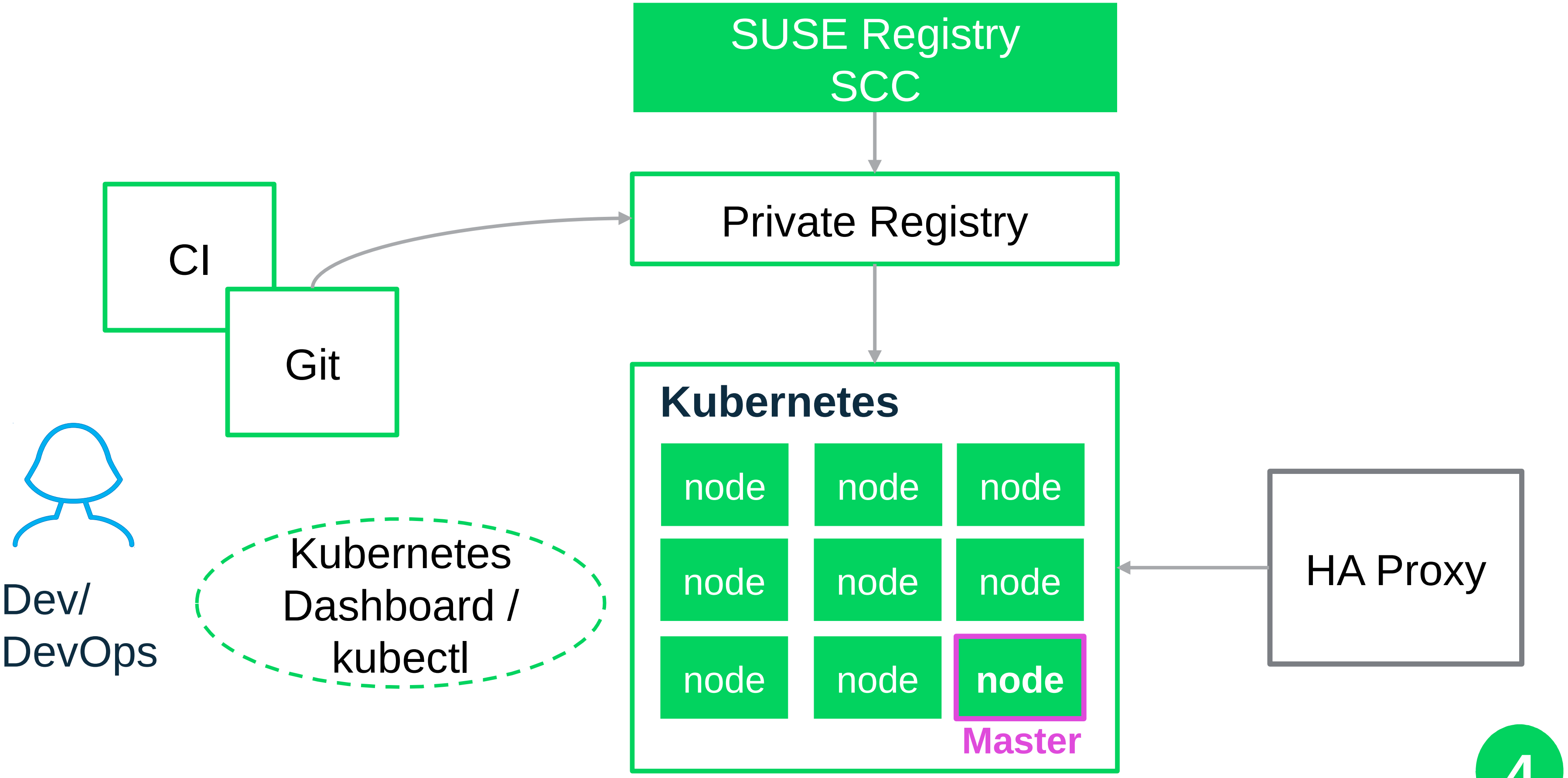
…

k8s Worker

k8s Worker

…

SUSE CaaS Platform Deployment – Run Containers



4 Ready for production

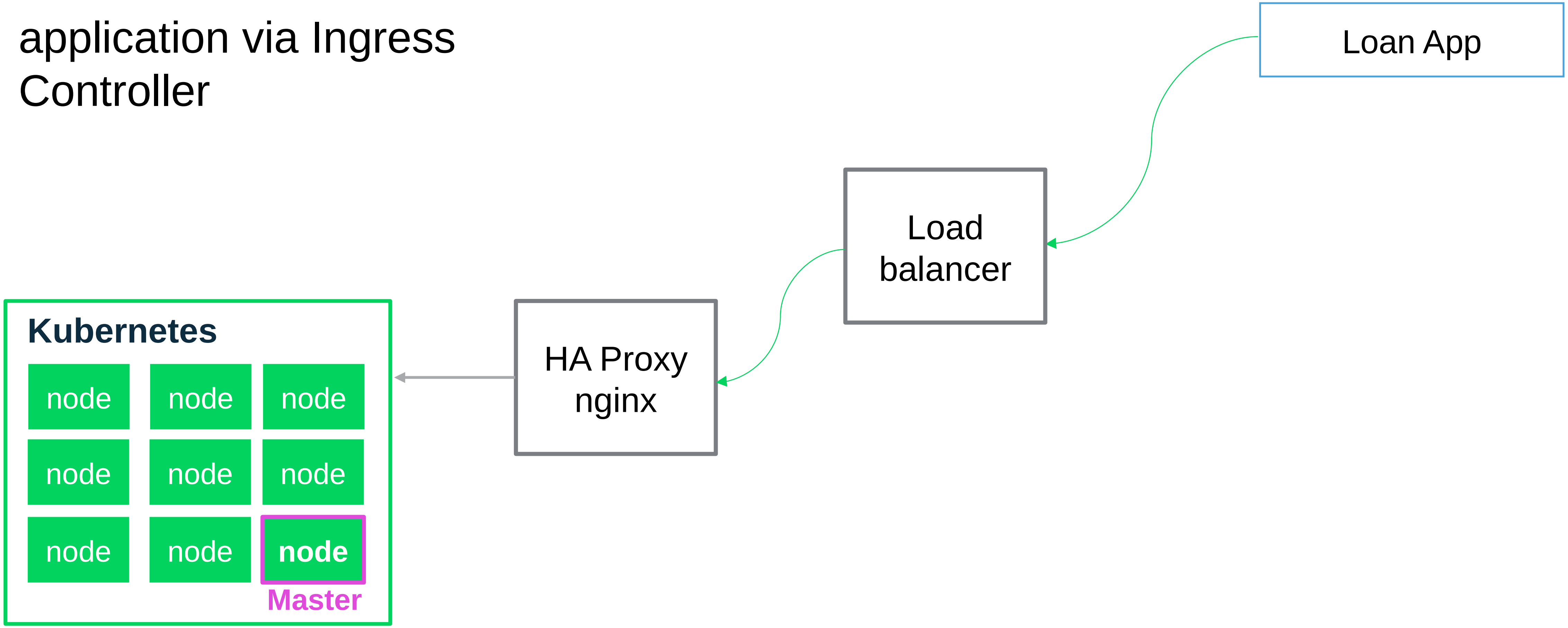
1 Push code to git

2 Build container image(s)

3 Run app on kubernetes cluster

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

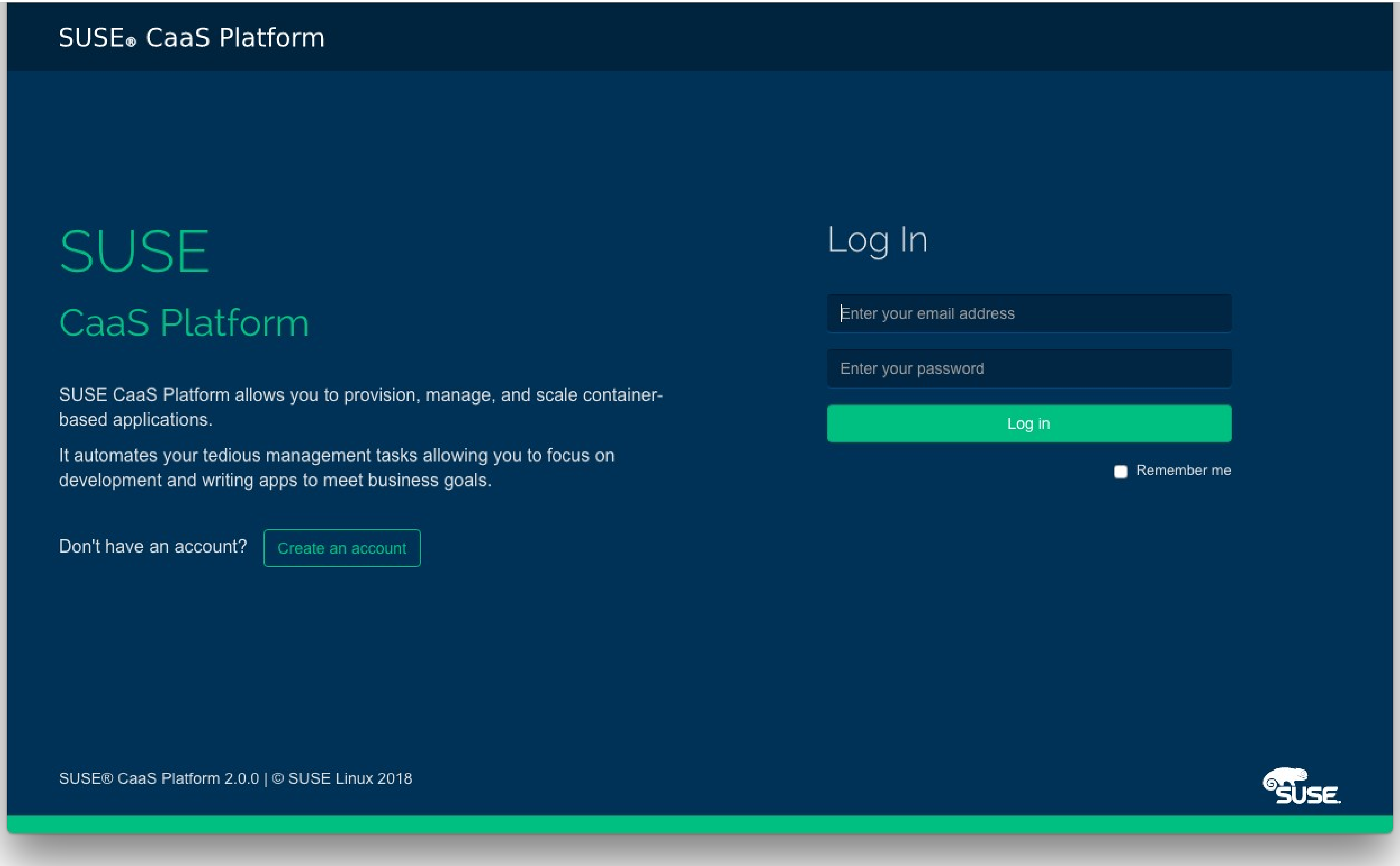
Atomic

Eliminate failures due to partial updates

Recoverable

Roll back easily to any previous release

SUSE CaaS Platform Setup



SUSE® CaaS Platform

SUSE

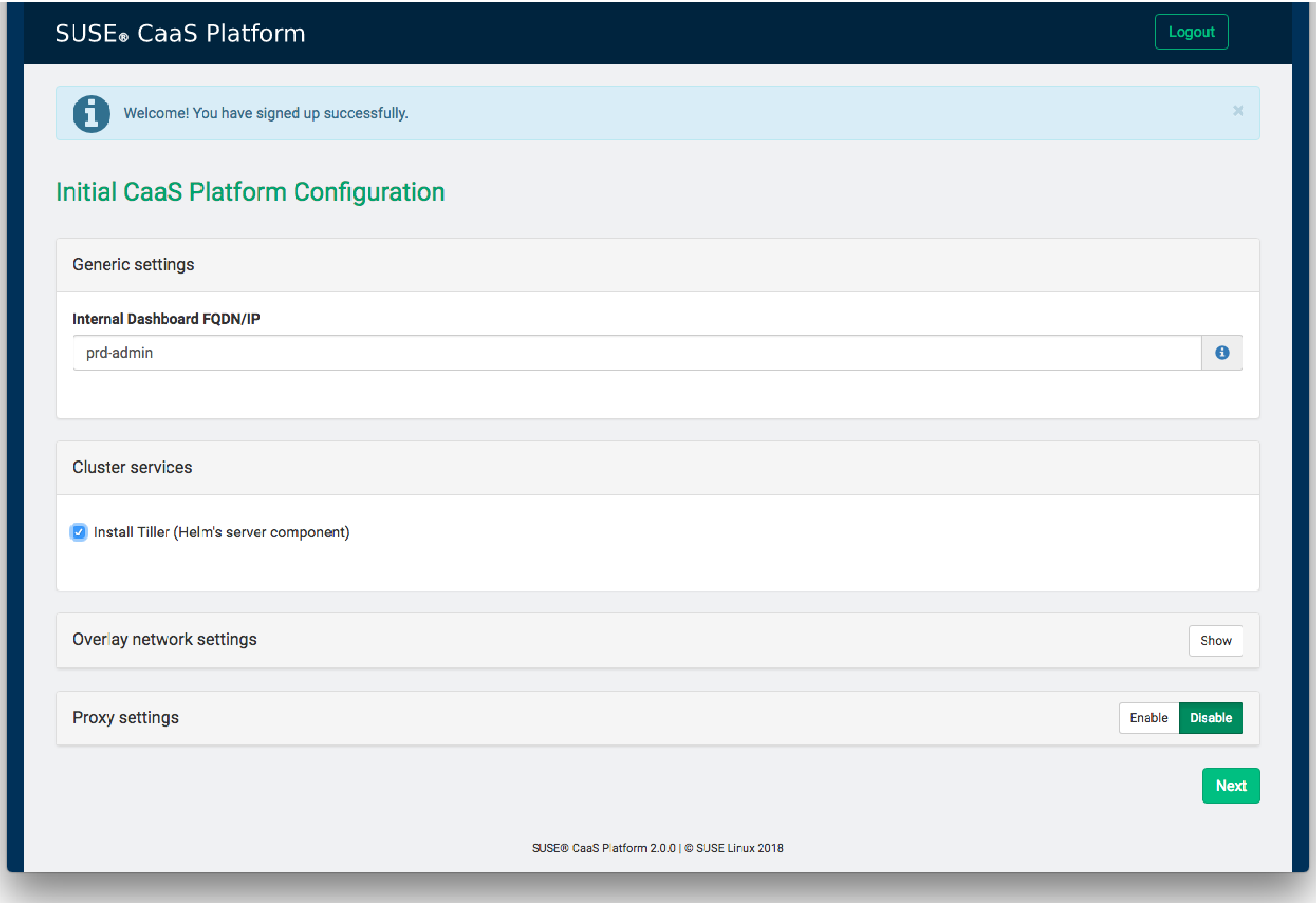
CaaS Platform

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?

SUSE® CaaS Platform 2.0.0 | © SUSE Linux 2018



SUSE® CaaS Platform

Welcome! You have signed up successfully.

Initial CaaS Platform Configuration

Generic settings

Internal Dashboard FQDN/IP

prd-admin

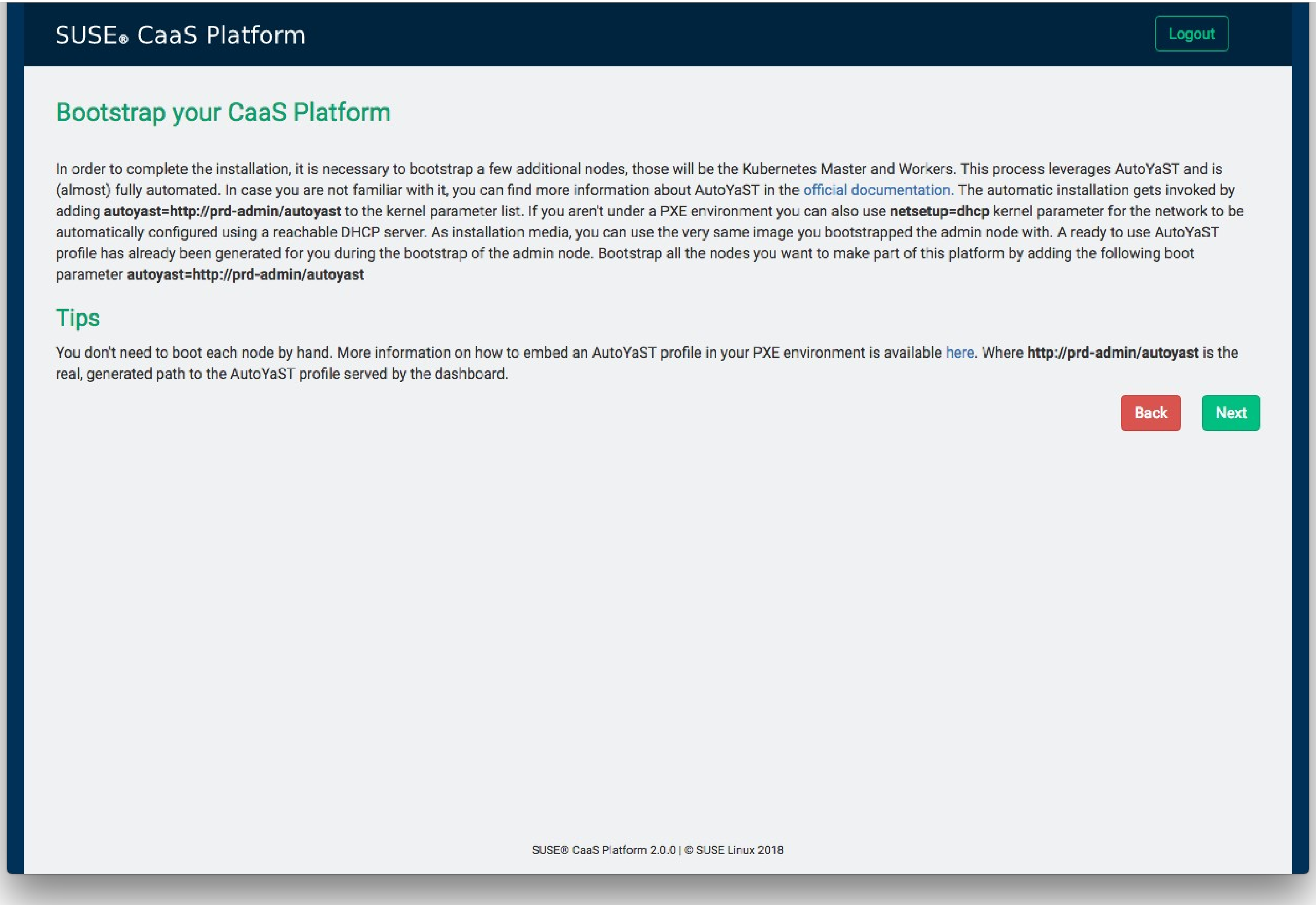
Cluster services

☑ Install Tiller (Helm's server component)

Overlay network settings

Proxy settings

SUSE® CaaS Platform 2.0.0 | © SUSE Linux 2018



SUSE® CaaS Platform

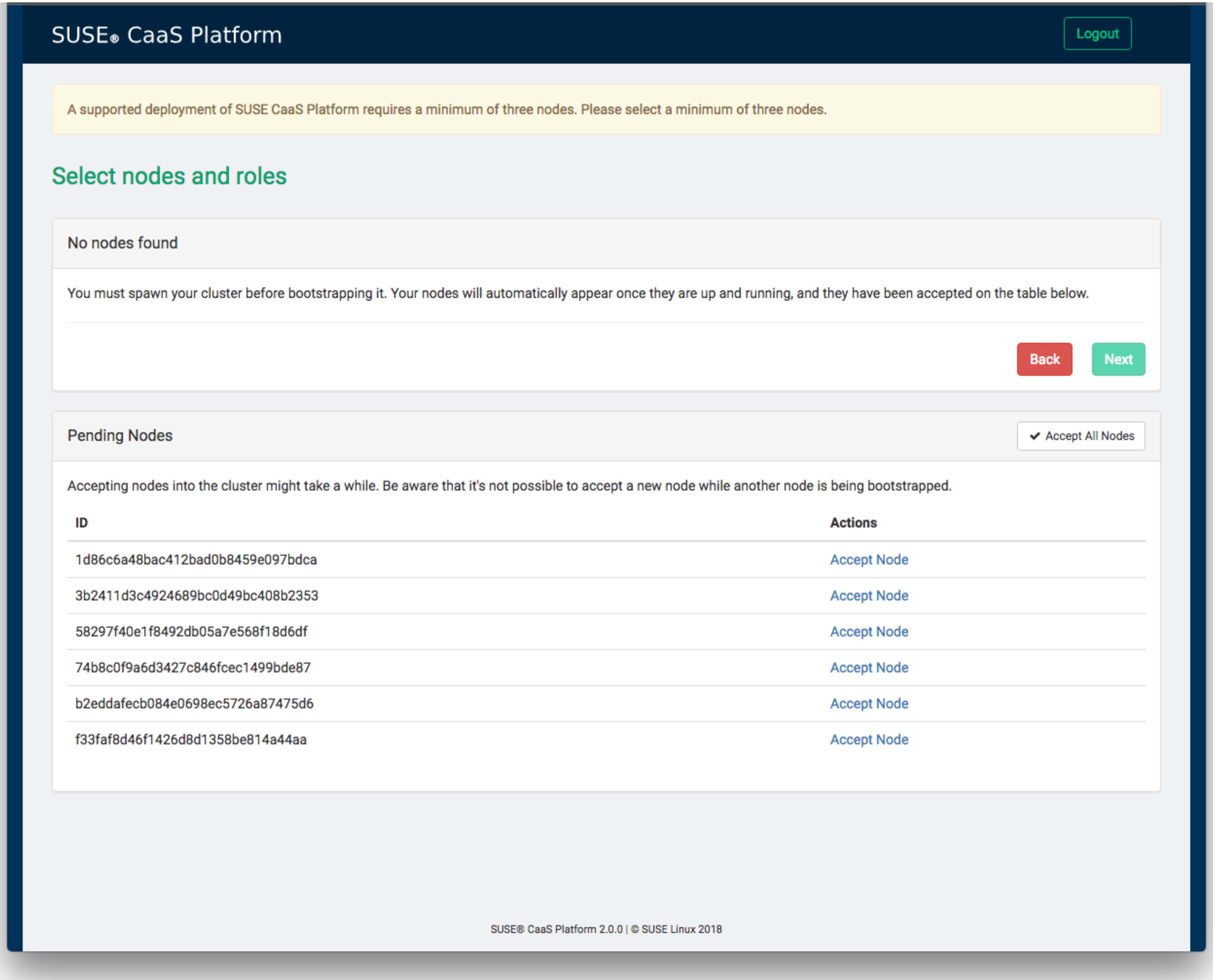
Bootstrap your CaaS Platform

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.

SUSE® CaaS Platform 2.0.0 | © SUSE Linux 2018



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

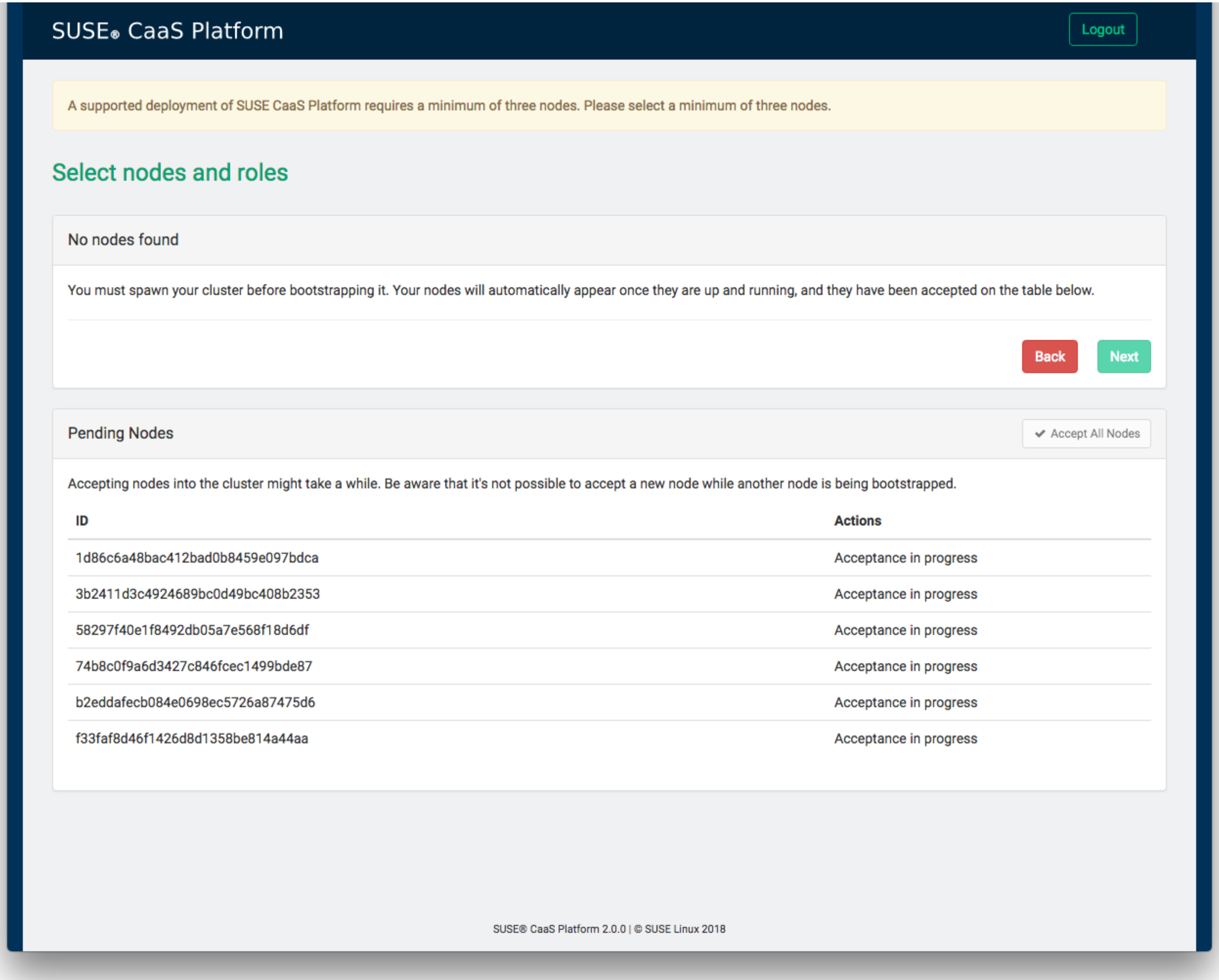
You must spawn your cluster before bootstrapping it. Your nodes will automatically appear once they are up and running, and they have been accepted on the table below.

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.

ID	Actions
1d86c6a48bac412bad0b8459e097bdca	Accept Node
3b2411d3c4924689bc0d49bc408b2353	Accept Node
58297f40e1f8492db05a7e568f18d6df	Accept Node
74b8c0f9a6d3427c846fcec1499bde87	Accept Node
b2eddafecb084e0698ec5726a87475d6	Accept Node
f33faf8d46f1426d8d1358be814a44aa	Accept Node

SUSE® CaaS Platform 2.0.0 | © SUSE Linux 2018



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

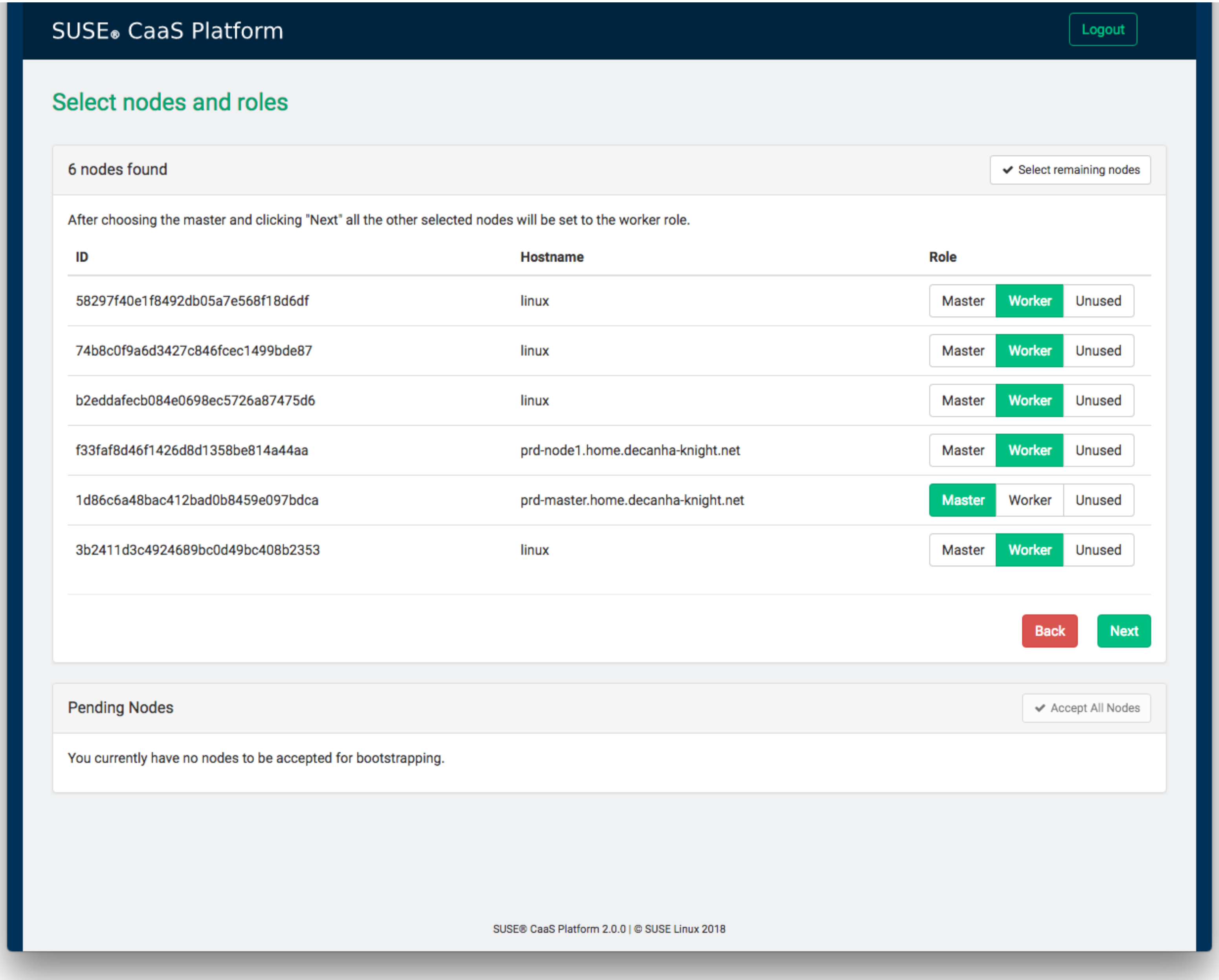
You must spawn your cluster before bootstrapping it. Your nodes will automatically appear once they are up and running, and they have been accepted on the table below.

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.

ID	Actions
1d86c6a48bac412bad0b8459e097bdca	Acceptance in progress
3b2411d3c4924689bc0d49bc408b2353	Acceptance in progress
58297f40e1f8492db05a7e568f18d6df	Acceptance in progress
74b8c0f9a6d3427c846fcec1499bde87	Acceptance in progress
b2eddafecb084e0698ec5726a87475d6	Acceptance in progress
f33faf8d46f1426d8d1358be814a44aa	Acceptance in progress

SUSE® CaaS Platform 2.0.0 | © SUSE Linux 2018



Select nodes and roles

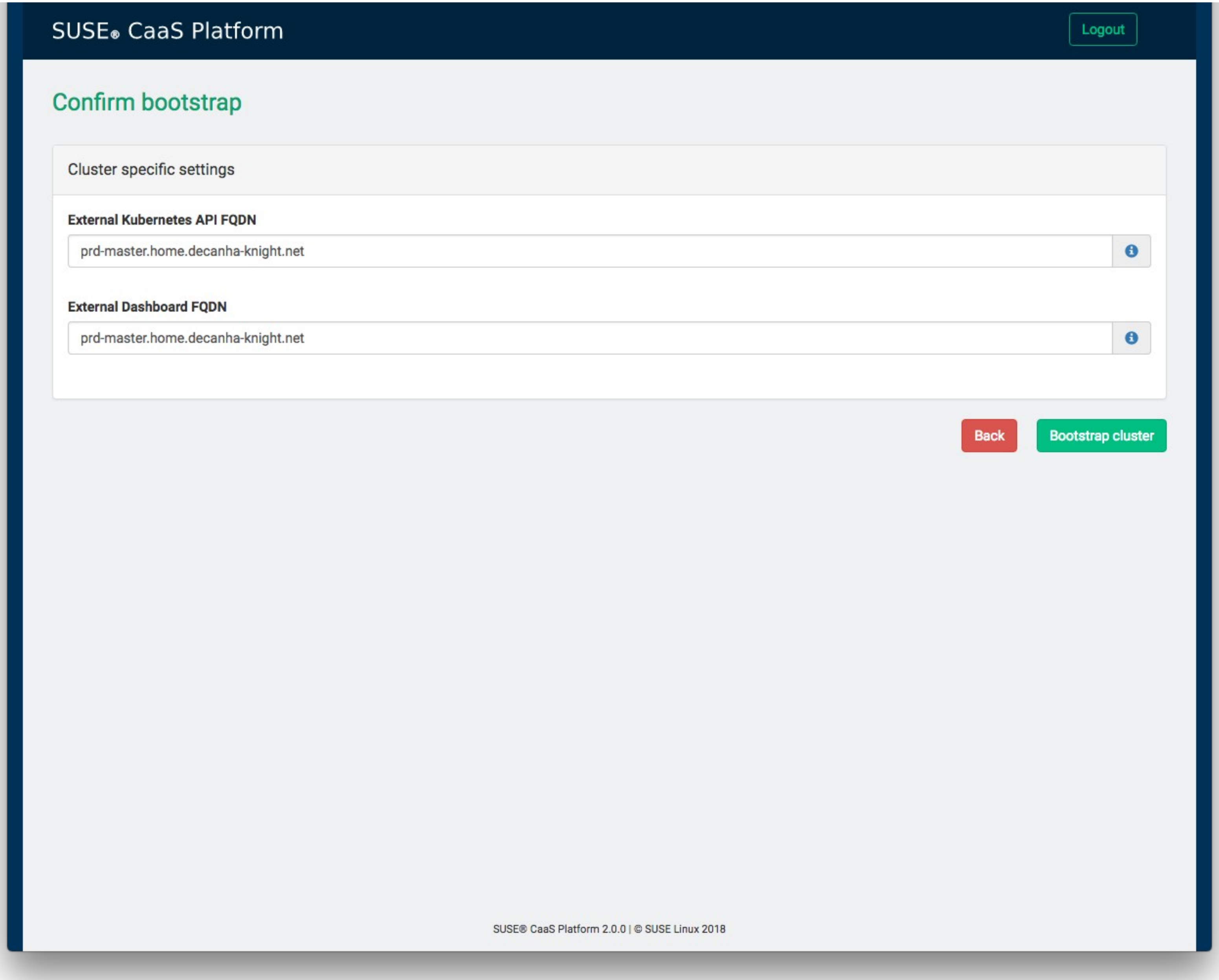
6 nodes found

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

ID	Hostname	Role
58297f40e1f8492db05a7e568f18d6df	linux	Master [✓ Worker] Unused
74b8c0f9a6d3427c846fcec1499bde87	linux	Master [✓ Worker] Unused
b2eddafecb084e0698ec5726a87475d6	linux	Master [✓ Worker] Unused
f33faf8d46f1426d8d1358be814a44aa	prd-node1.home.decanha-knight.net	Master [✓ Worker] Unused
1d86c6a48bac412bad0b8459e097bdca	prd-master.home.decanha-knight.net	[✓ Master] Worker Unused
3b2411d3c4924689bc0d49bc408b2353	linux	Master [✓ Worker] Unused

Pending Nodes

You currently have no nodes to be accepted for bootstrapping.



Confirm bootstrap

Cluster specific settings

External Kubernetes API FQDN

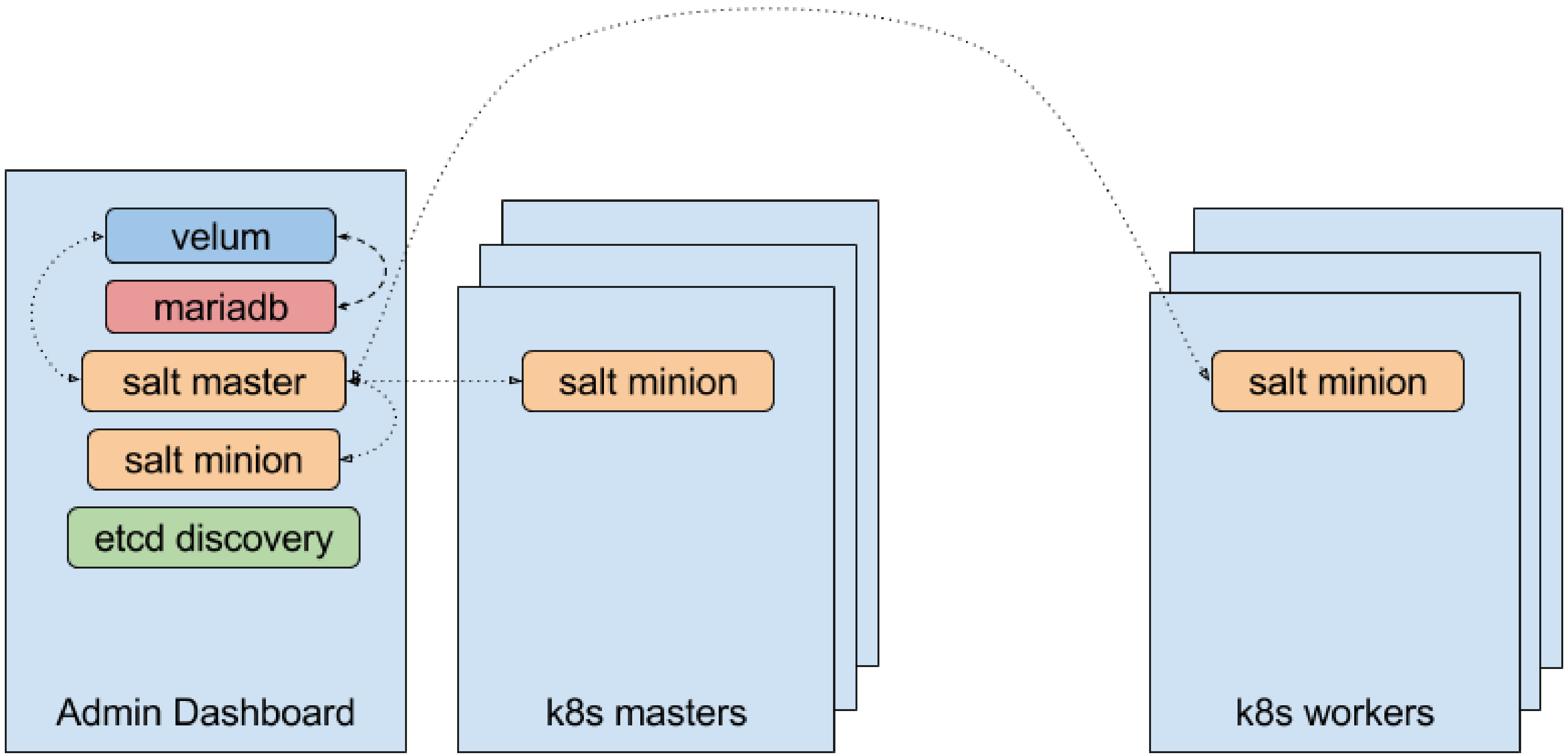
prd-master.home.decanha-knight.net

External Dashboard FQDN

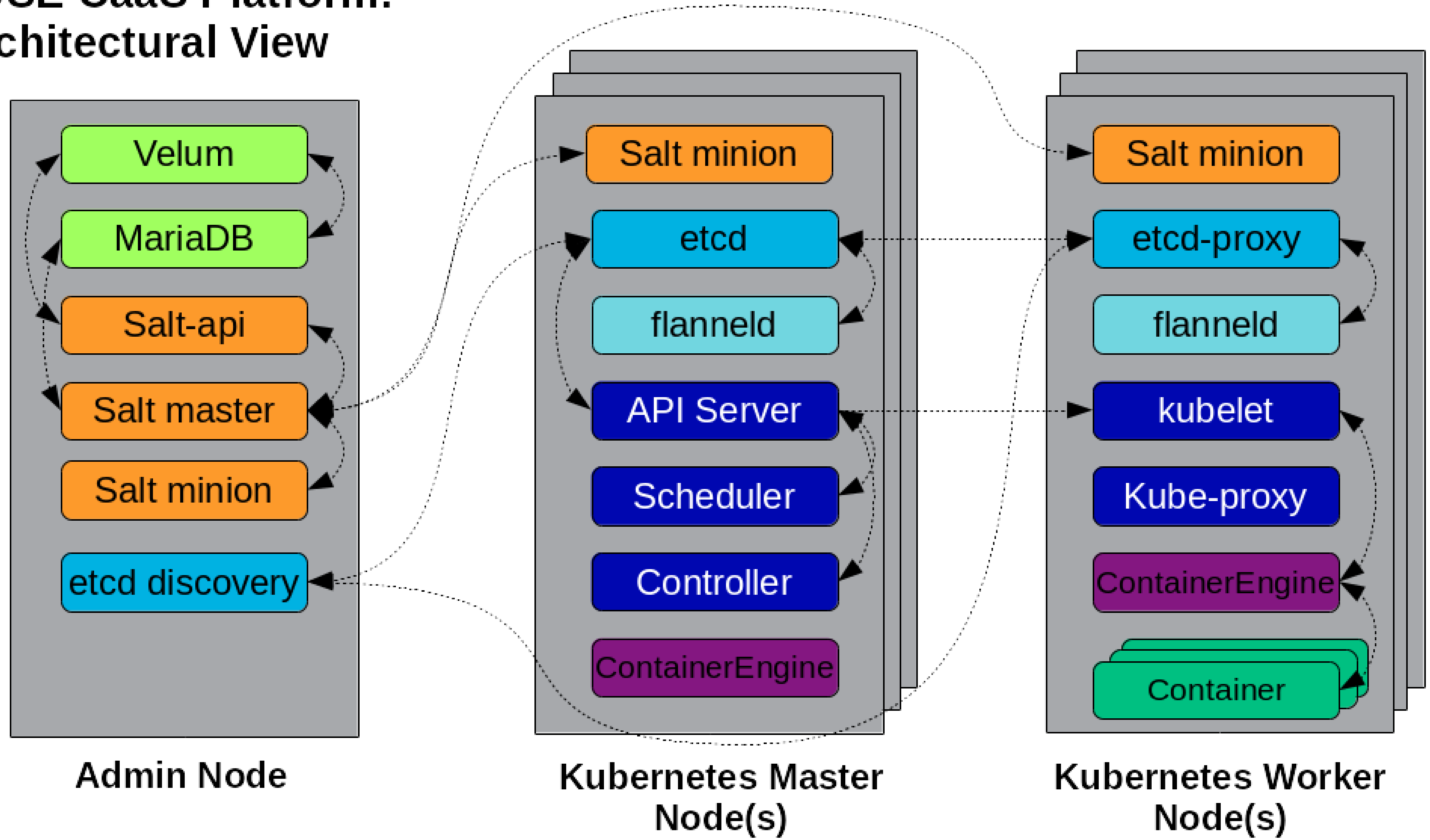
prd-master.home.decanha-knight.net

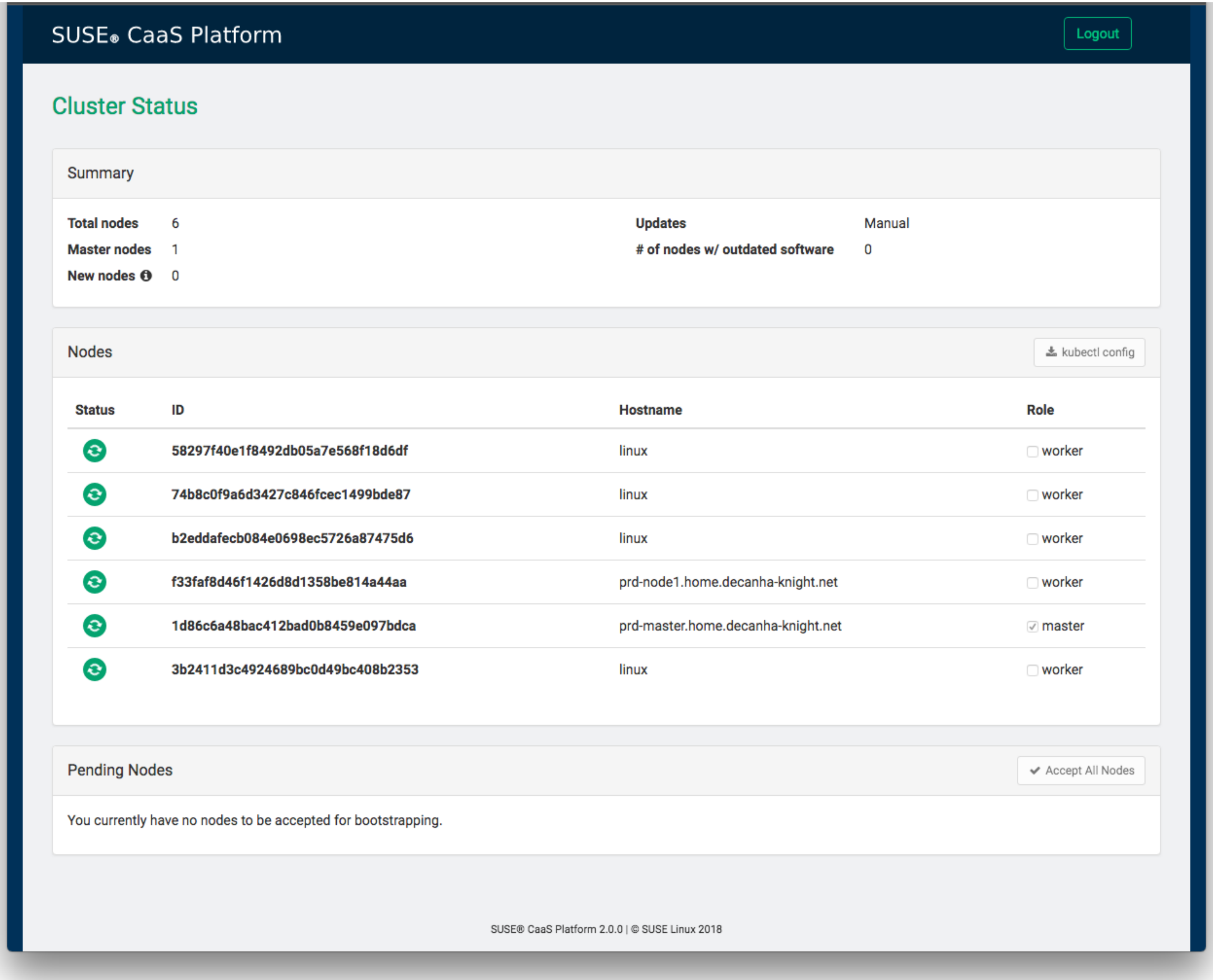
SUSE® CaaS Platform 2.0.0 | © SUSE Linux 2018

Before Salt Orchestration



SUSE CaaS Platform: Architectural View





Cluster Status

Summary

Total nodes	6	**Updates**	Manual
Master nodes	1	**# of nodes w/ outdated software**	0
New nodes ℹ	0		

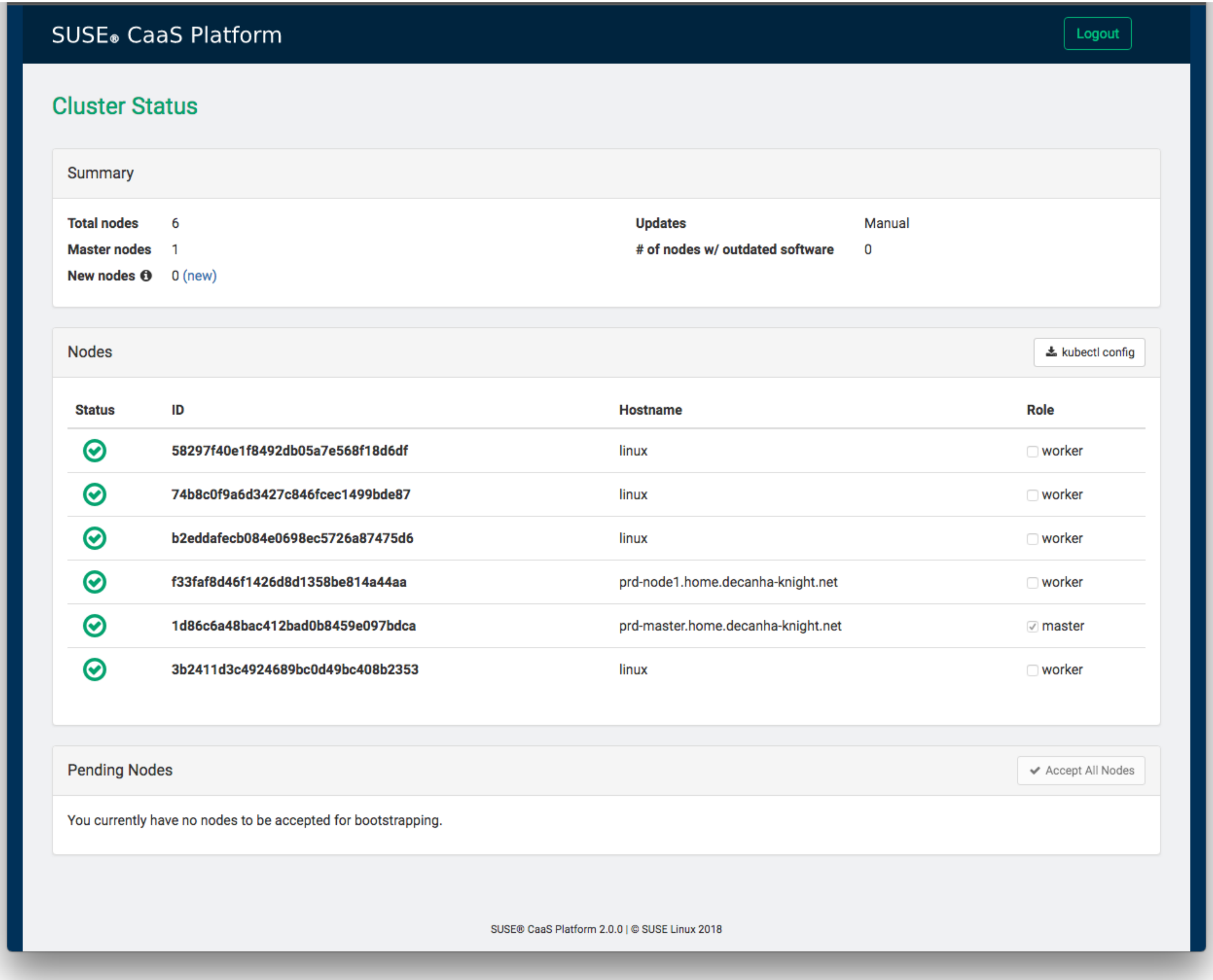
Nodes

Status	ID	Hostname	Role
⟳	58297f40e1f8492db05a7e568f18d6df	linux	□ worker
⟳	74b8c0f9a6d3427c846fcec1499bde87	linux	□ worker
⟳	b2eddafecb084e0698ec5726a87475d6	linux	□ worker
⟳	f33faf8d46f1426d8d1358be814a44aa	prd-node1.home.decanha-knight.net	□ worker
⟳	1d86c6a48bac412bad0b8459e097bdca	prd-master.home.decanha-knight.net	☑ master
⟳	3b2411d3c4924689bc0d49bc408b2353	linux	□ worker

Pending Nodes

You currently have no nodes to be accepted for bootstrapping.

SUSE® CaaS Platform 2.0.0 | © SUSE Linux 2018



Cluster Status

Summary

Total nodes	6	**Updates**	Manual
Master nodes	1	**# of nodes w/ outdated software**	0
New nodes	0 (new)		

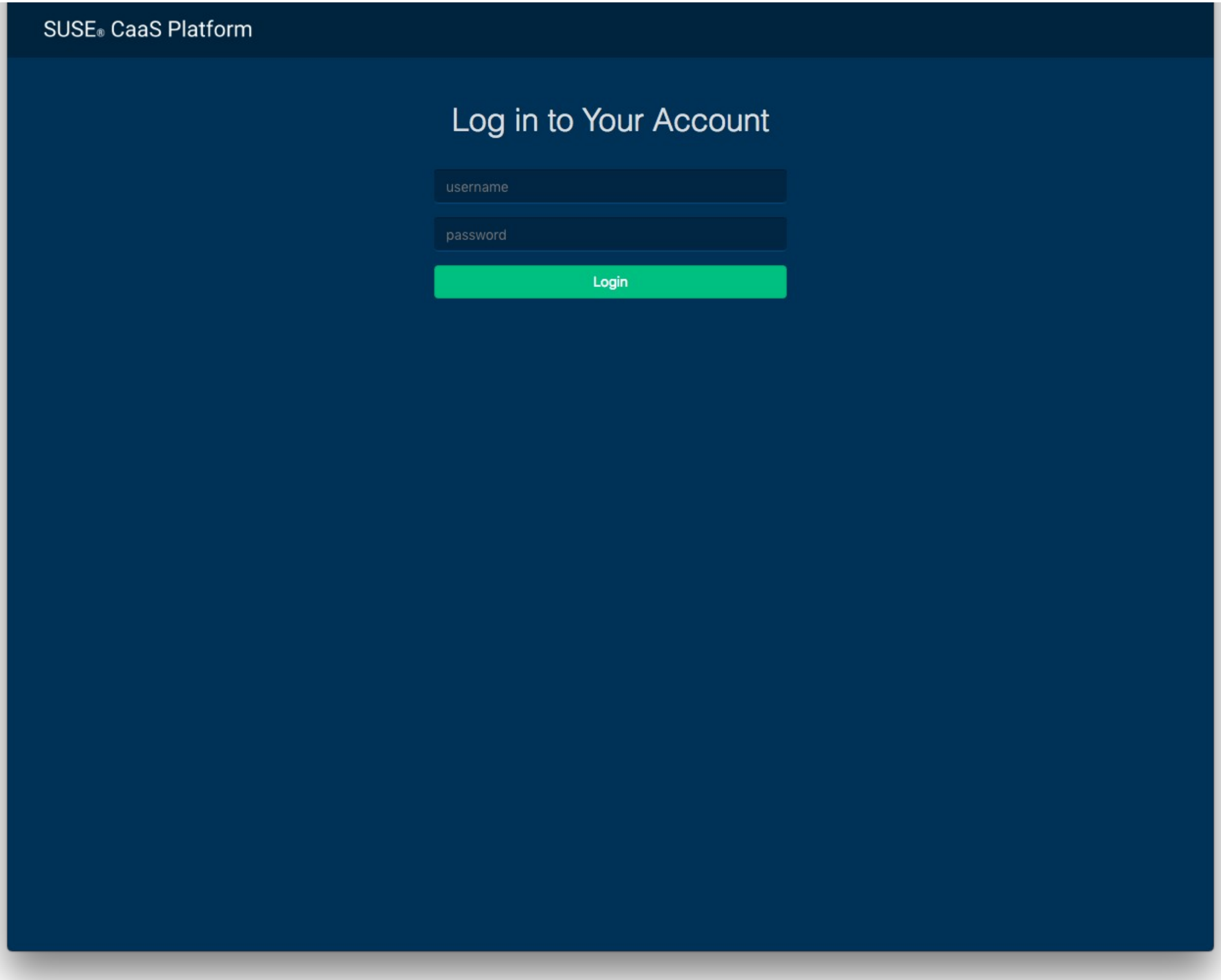
Nodes

Status	ID	Hostname	Role
✓	58297f40e1f8492db05a7e568f18d6df	linux	□ worker
✓	74b8c0f9a6d3427c846fcec1499bde87	linux	□ worker
✓	b2eddafecb084e0698ec5726a87475d6	linux	□ worker
✓	f33faf8d46f1426d8d1358be814a44aa	prd-node1.home.decanha-knight.net	□ worker
✓	1d86c6a48bac412bad0b8459e097bdca	prd-master.home.decanha-knight.net	☑ master
✓	3b2411d3c4924689bc0d49bc408b2353	linux	□ worker

Pending Nodes

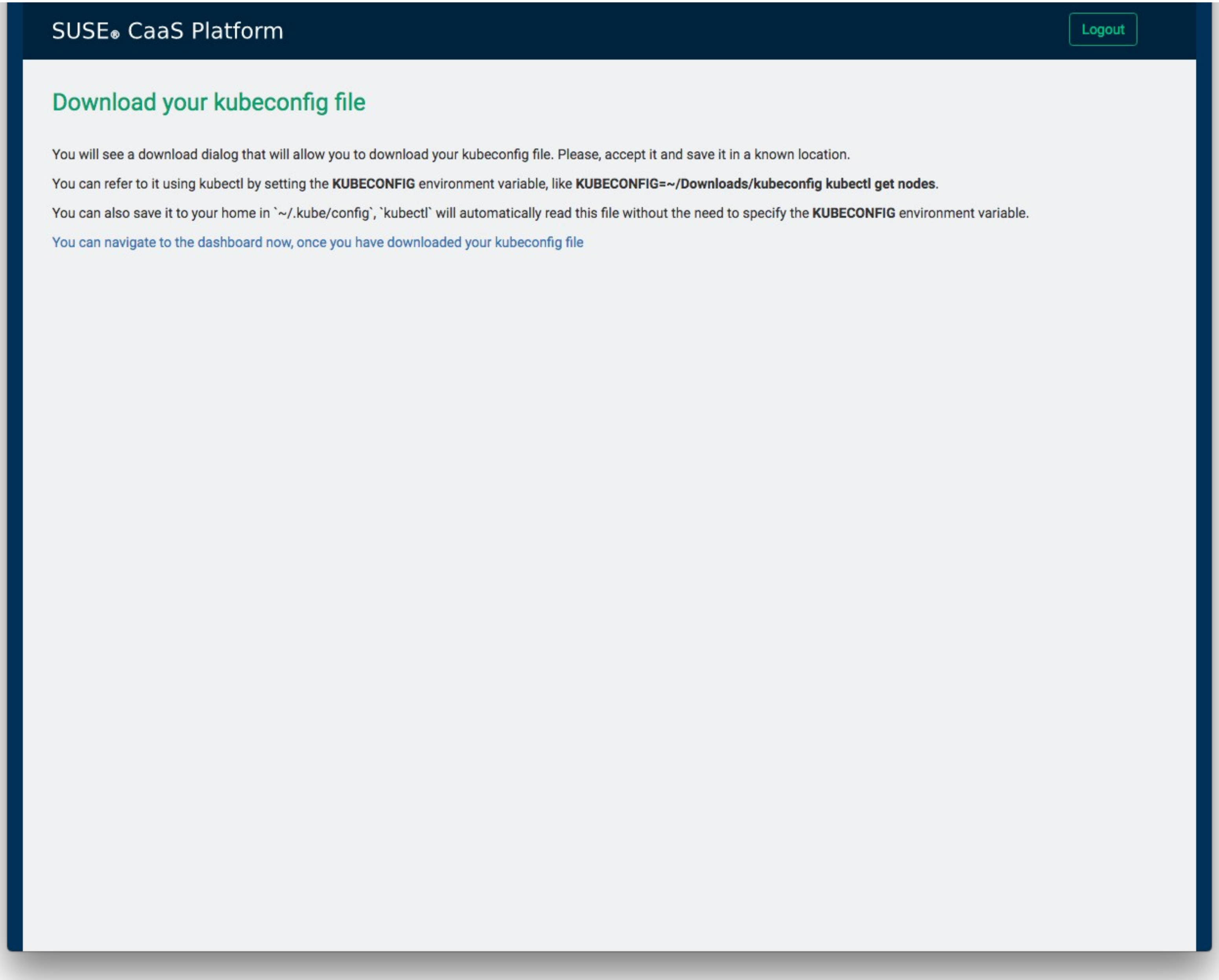
You currently have no nodes to be accepted for bootstrapping.

SUSE® CaaS Platform 2.0.0 | © SUSE Linux 2018



SUSE® CaaS Platform

Log in to Your Account



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.

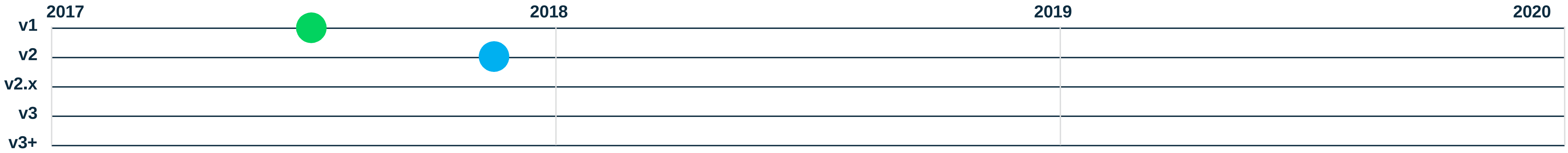
You can refer to it using kubectl by setting the **KUBECONFIG** environment variable, like **KUBECONFIG=~/Downloads/kubeconfig kubectl get nodes**.

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



v1

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

v2

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

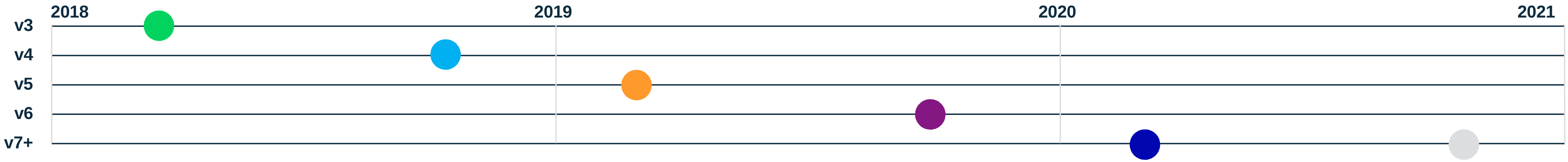
Ecosystem

- 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

v4

Container Host OS

- Codebase: SUSE Linux Enterprise 15
- Codebase and packaging further optimized as container host OS
- Improve isolation options
- Container engine alternative (cri-o)
- Disaster recovery
- Containerized control pane
- Support customer certificate authority

Orchestration

- Kubernetes version update
- Network options (CNI and Cilium as first plugin)
- Kubernetes dashboard (kube-dash)
- Federation of Kubernetes cluster

Ecosystem

- Enable SUSE Enterprise Storage on top of SUSE CaaS Platform
- Smaller base container images
- Additional container images for workloads
- Overlap support

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)
- Monitoring (Prometheus)

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
- IPv6

Ecosystem

- Full Management of CaaS Platform within SUSE Manager

Themes

- Continue to make K8s easy to install, update, and operate
- Multi-cloud, multi-cluster
- Integration into customer environments (storage, networking)

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:

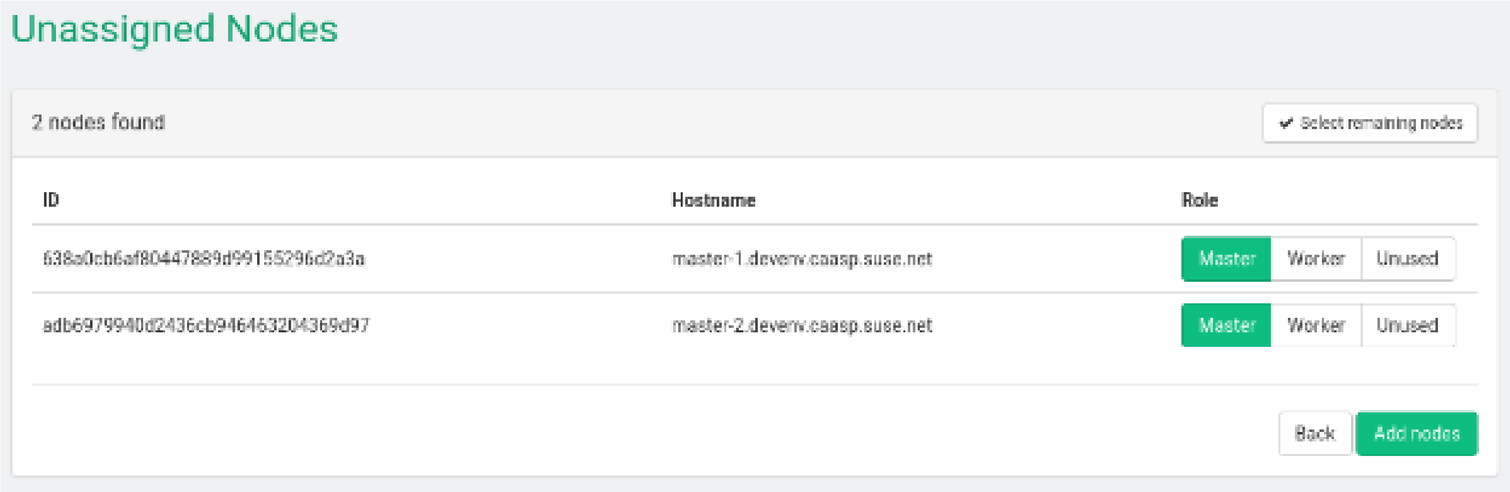
- Troubleshooting system

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



Unassigned Nodes

2 nodes found

ID	Hostname	Role
638a0cb6af80447889d99155296d2a3a	master-1.devenv.caasp.suse.net	[✓ Master] Worker Unused
adb6979940d2436cb946463204369d97	master-2.devenv.caasp.suse.net	[✓ Master] Worker Unused

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**

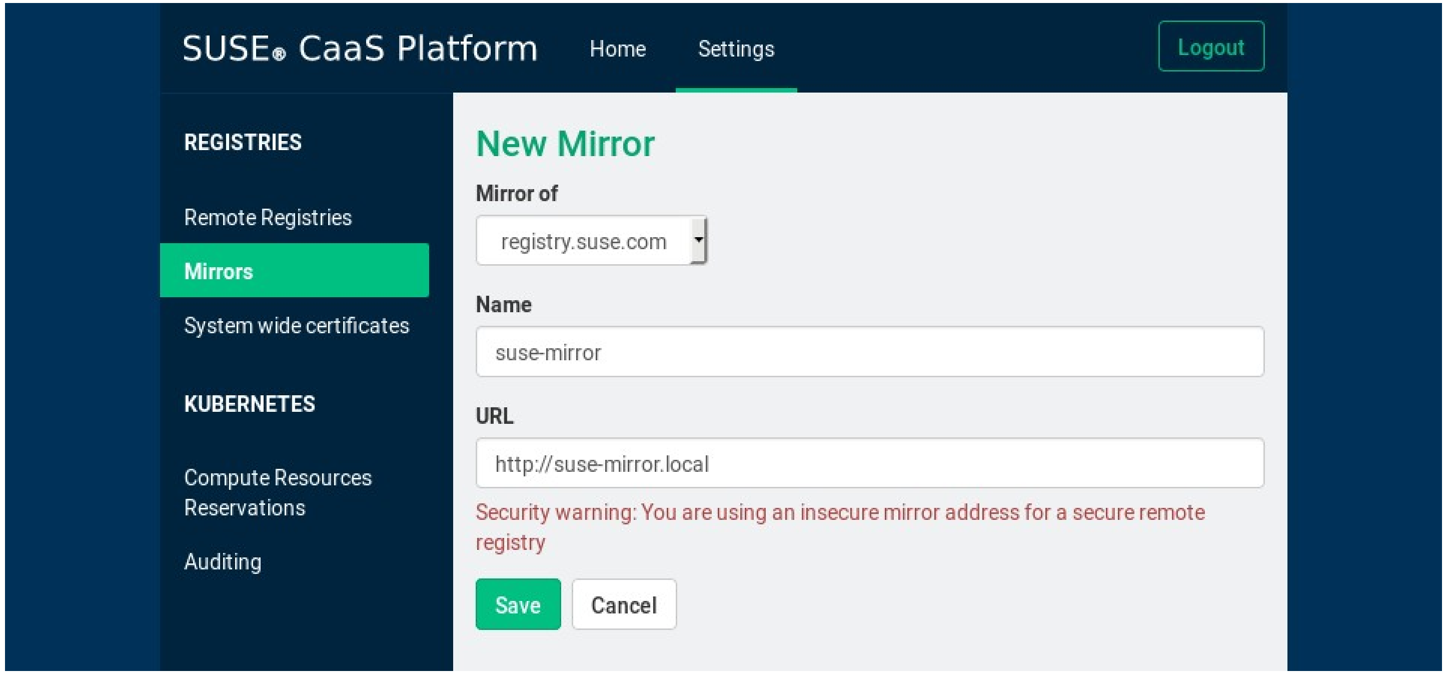
Public registry.suse.com

- **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 develoment 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



New Mirror

Mirror of: registry.suse.com

Name: suse-mirror

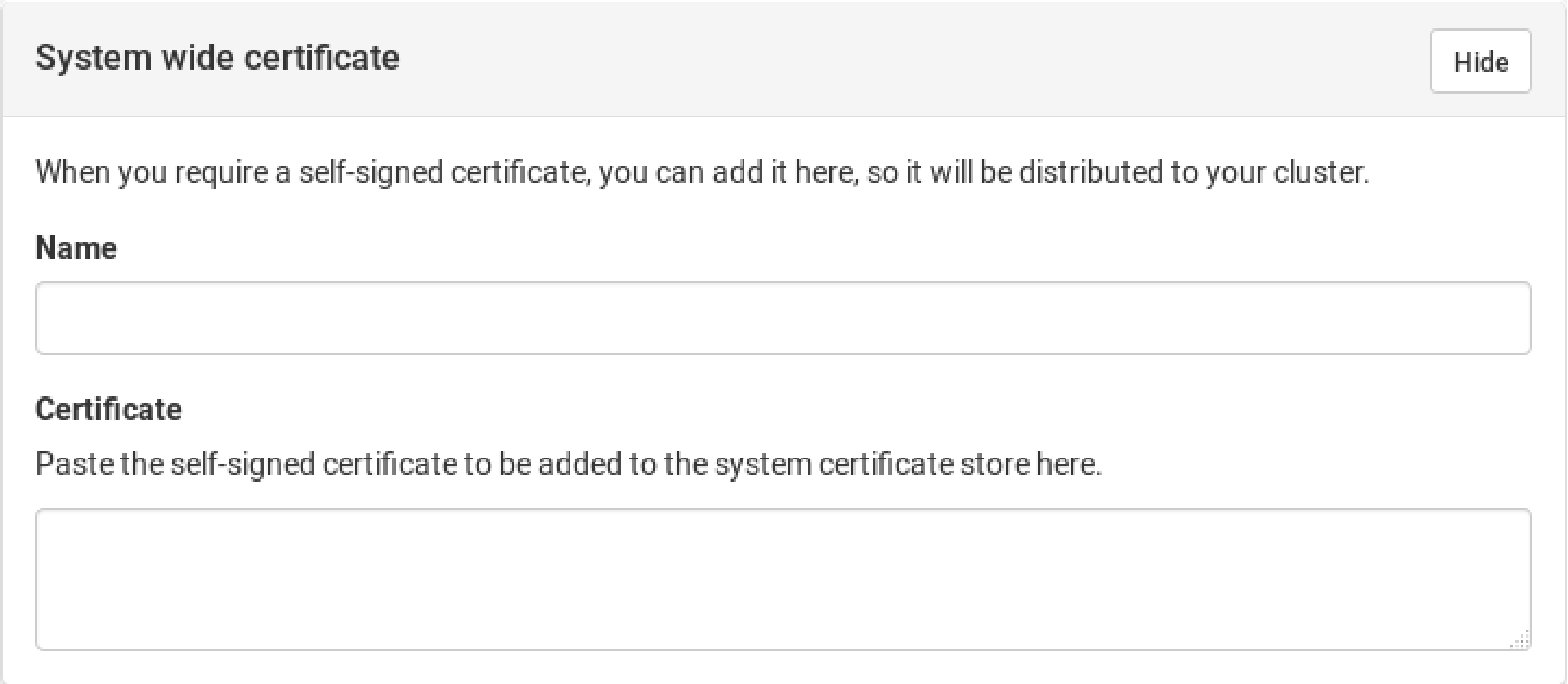
URL: http://suse-mirror.local

Security warning: You are using an insecure mirror address for a secure remote registry

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



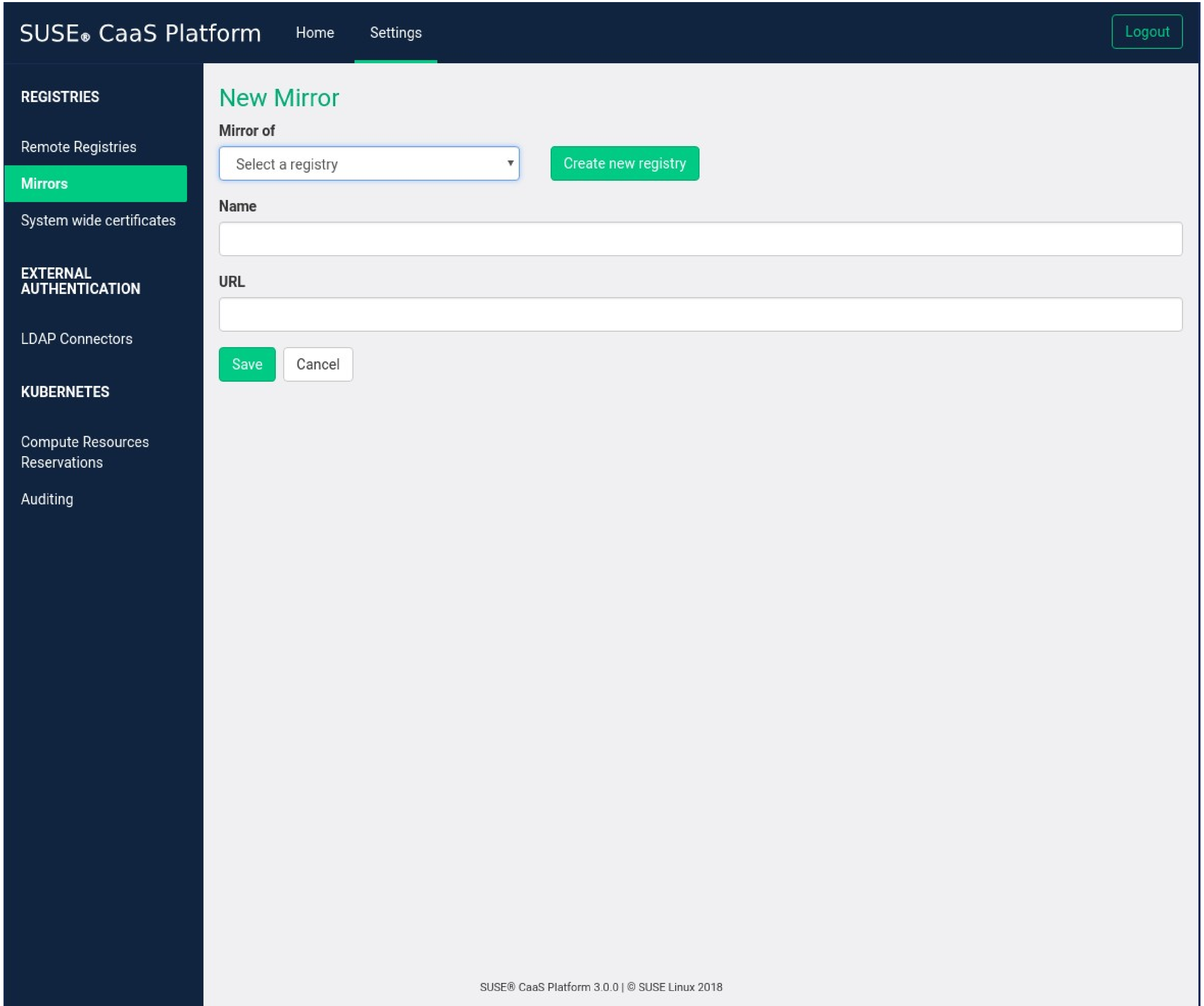
System wide certificate

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

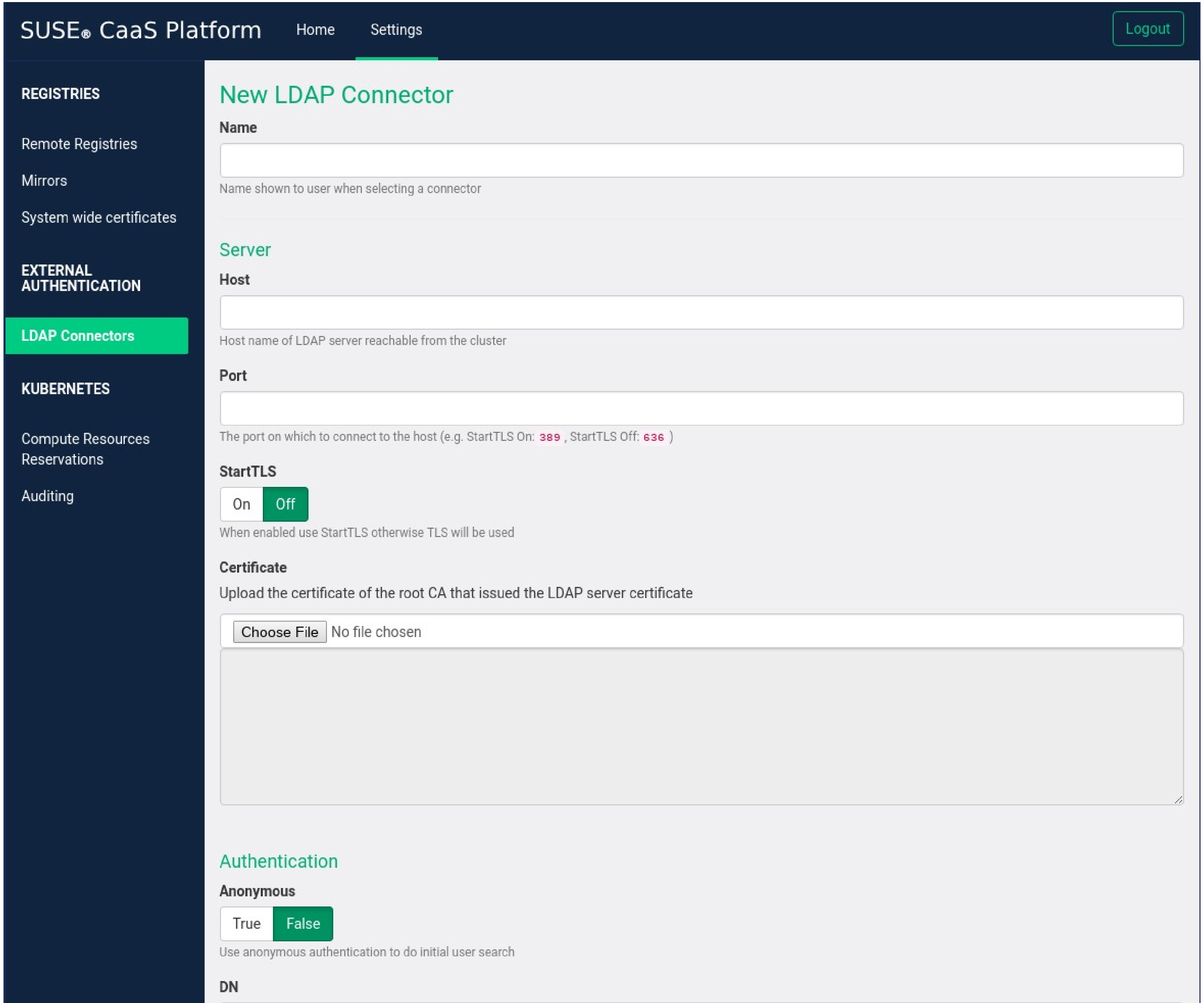
Mirror of

Select a registry ▾

Name

URL

SUSE® CaaS Platform 3.0.0 | © SUSE Linux 2018



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 [✓ Off]

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

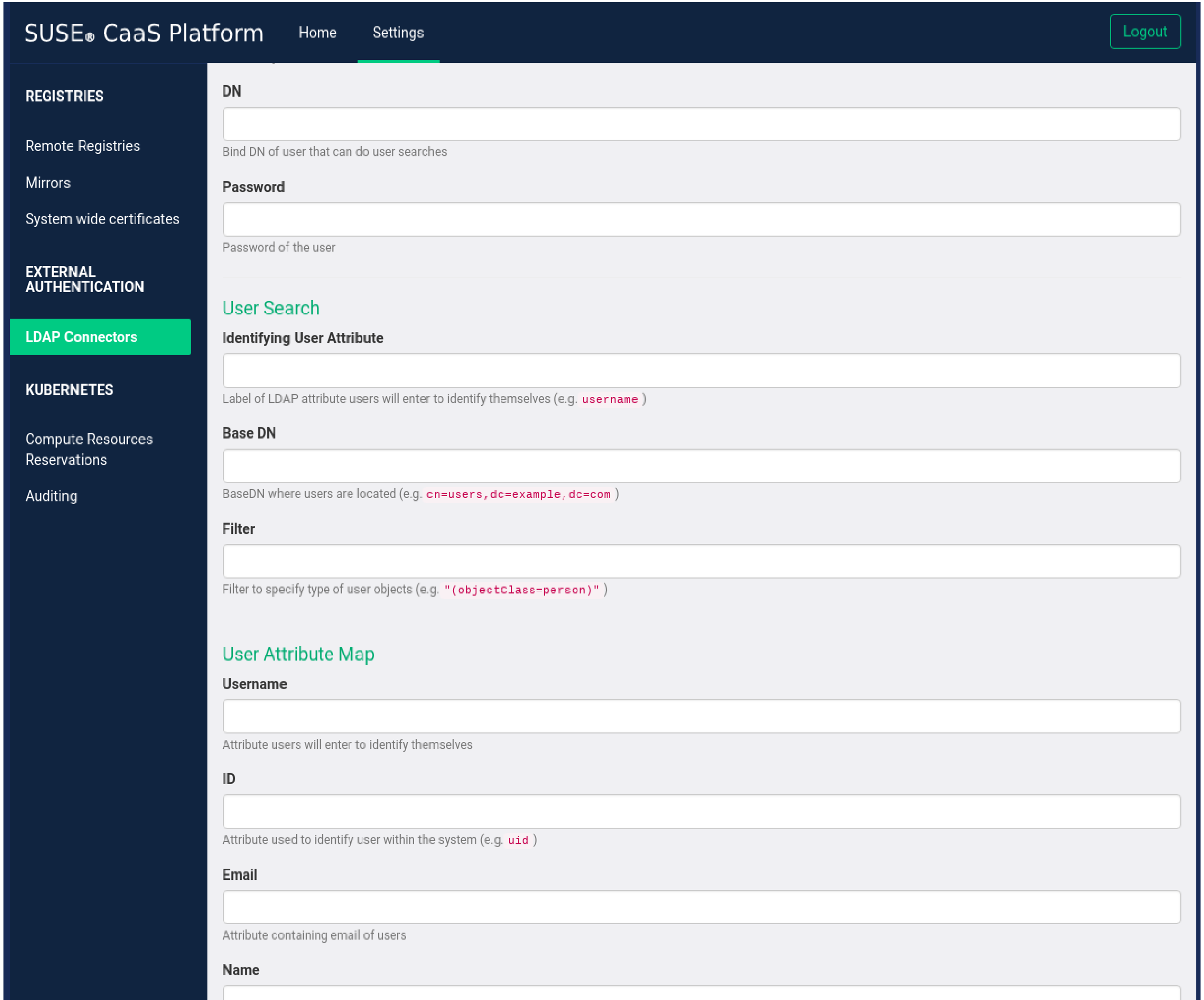
Authentication

Anonymous

True [✓ False]

Use anonymous authentication to do initial user search

DN



DN

Bind DN of user that can do user searches

Password

Password of the user

User Search

Identifying User Attribute

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

Base DN

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

Filter

Filter to specify type of user objects (e.g. `"(objectClass=person)"`)

User Attribute Map

Username

Attribute users will enter to identify themselves

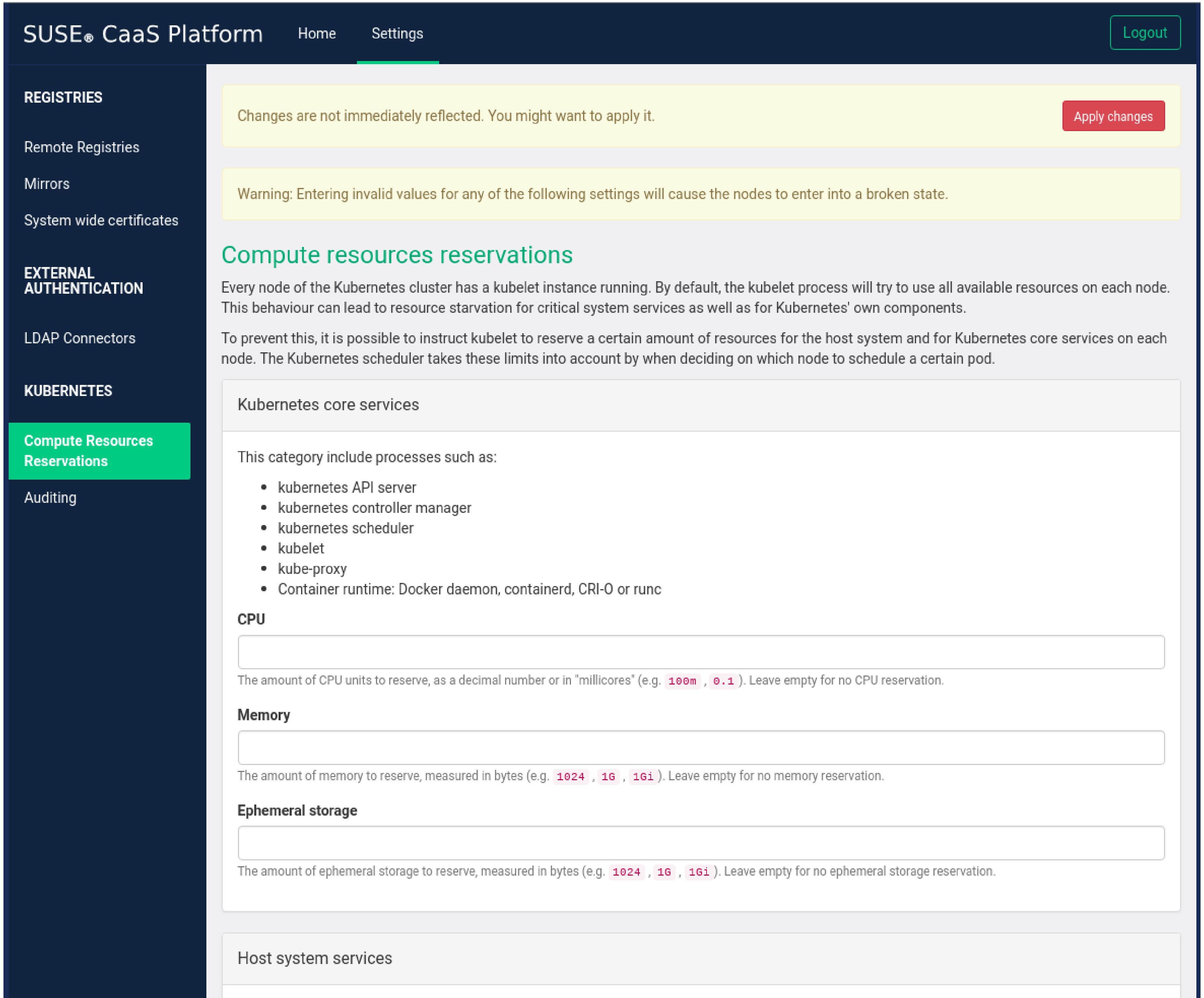
ID

Attribute used to identify user within the system (e.g. `uid`)

Email

Attribute containing email of users

Name



Changes are not immediately reflected. You might want to apply it.

Warning: Entering invalid values for any of the following settings will cause the nodes to enter into a broken state.

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 include processes such as:

- kubernetes API server
- kubernetes controller manager
- kubernetes scheduler
- kubelet
- kube-proxy
- Container runtime: Docker daemon, containerd, CRI-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.

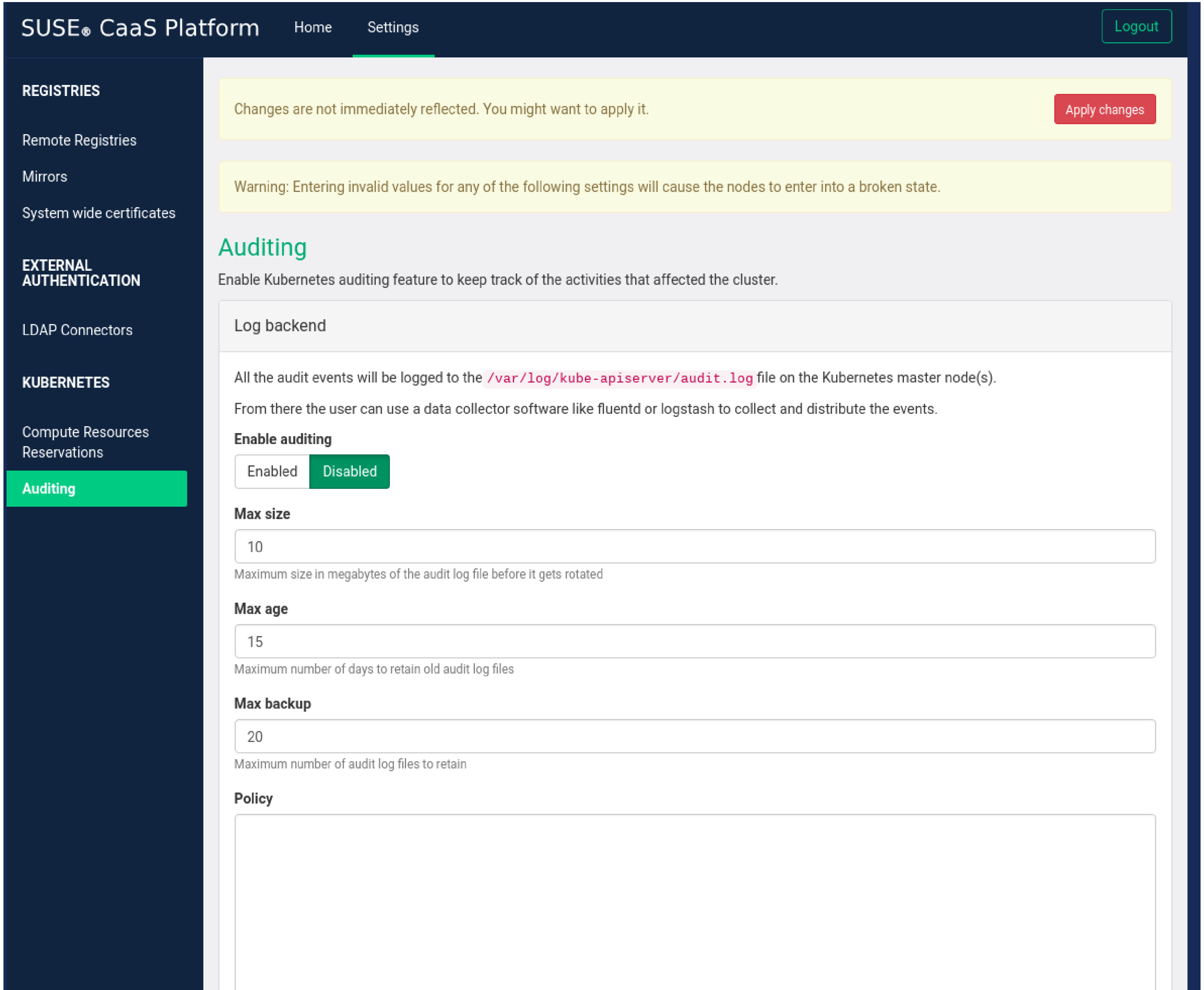
Memory

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

Ephemeral storage

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

Host system services



Changes are not immediately reflected. You might want to apply it.

Warning: Entering invalid values for any of the following settings will cause the nodes to enter into a broken state.

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:
  kubelet → docker-shim → dockerd → containerd → runC
- Designed with Kubernetes in mind:
  kubelet → crio → 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

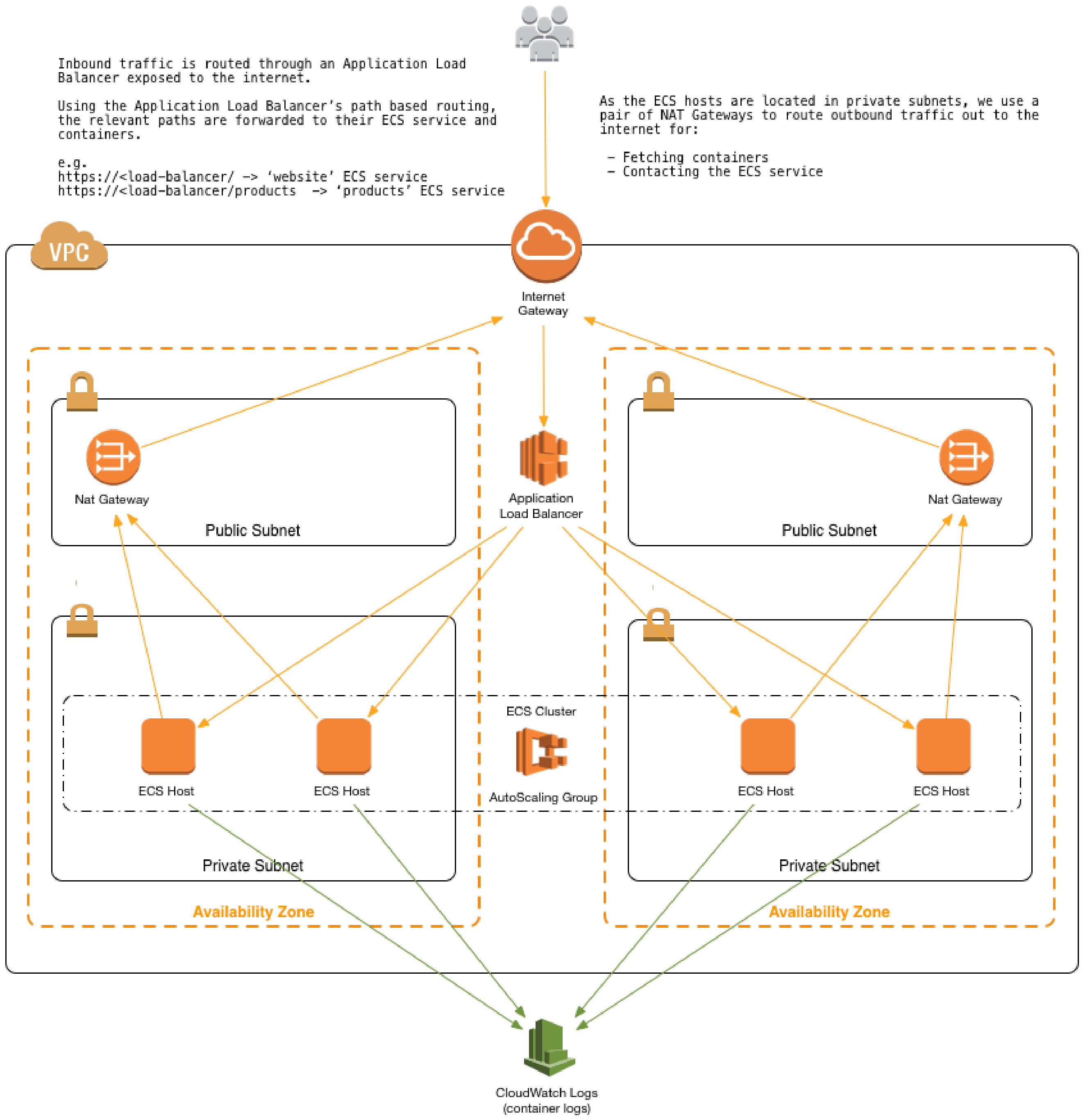
Kubernetes version upgrade

- 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



```
Inbound traffic is routed through an Application Load
Balancer exposed to the internet.

Using the Application Load Balancer's path based routing,
the relevant paths are forwarded to their ECS service and
containers.

e.g.
https://<load-balancer/ -> 'website' ECS service
https://<load-balancer/products  -> 'products' ECS service
```

```
As the ECS hosts are located in private subnets, we use a
pair of NAT Gateways to route outbound traffic out to the
internet for:

 - Fetching containers
 - Contacting the ECS service
```

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
```

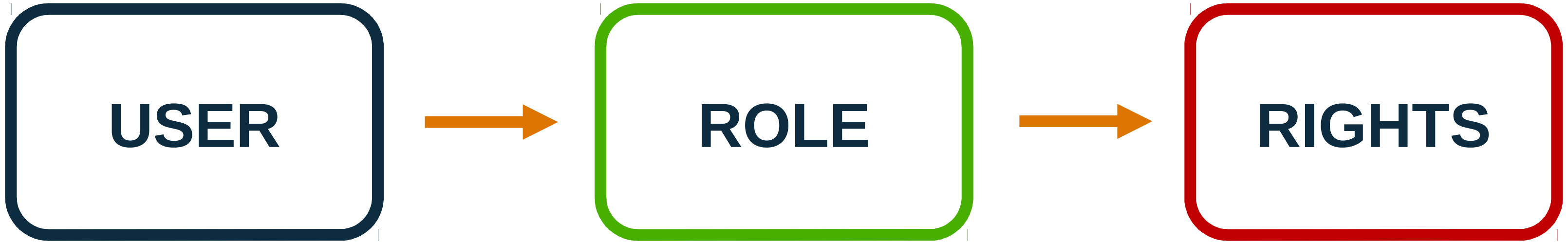
SUSE CaaS Platform 2

Public Cloud

Images available in Market Place	Ready to run images for Amazon AWS, Azure and Google Cloud Bring your own subscription
On-demand	Discussion started with Google Cloud and Azure Not available yet. Timeline: 3-6 months
Federation	In discussion for upcoming releases

SUSE CaaS Platform 2

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

Sys Admin	Operate the infrastructure Block access to the infrastructure level Allow developers to interact with Kubernetes
Developer	Full access for my team to manage the application No access to other teams work No access from other team to our work
Manager	Check the usage Have an overview of resources

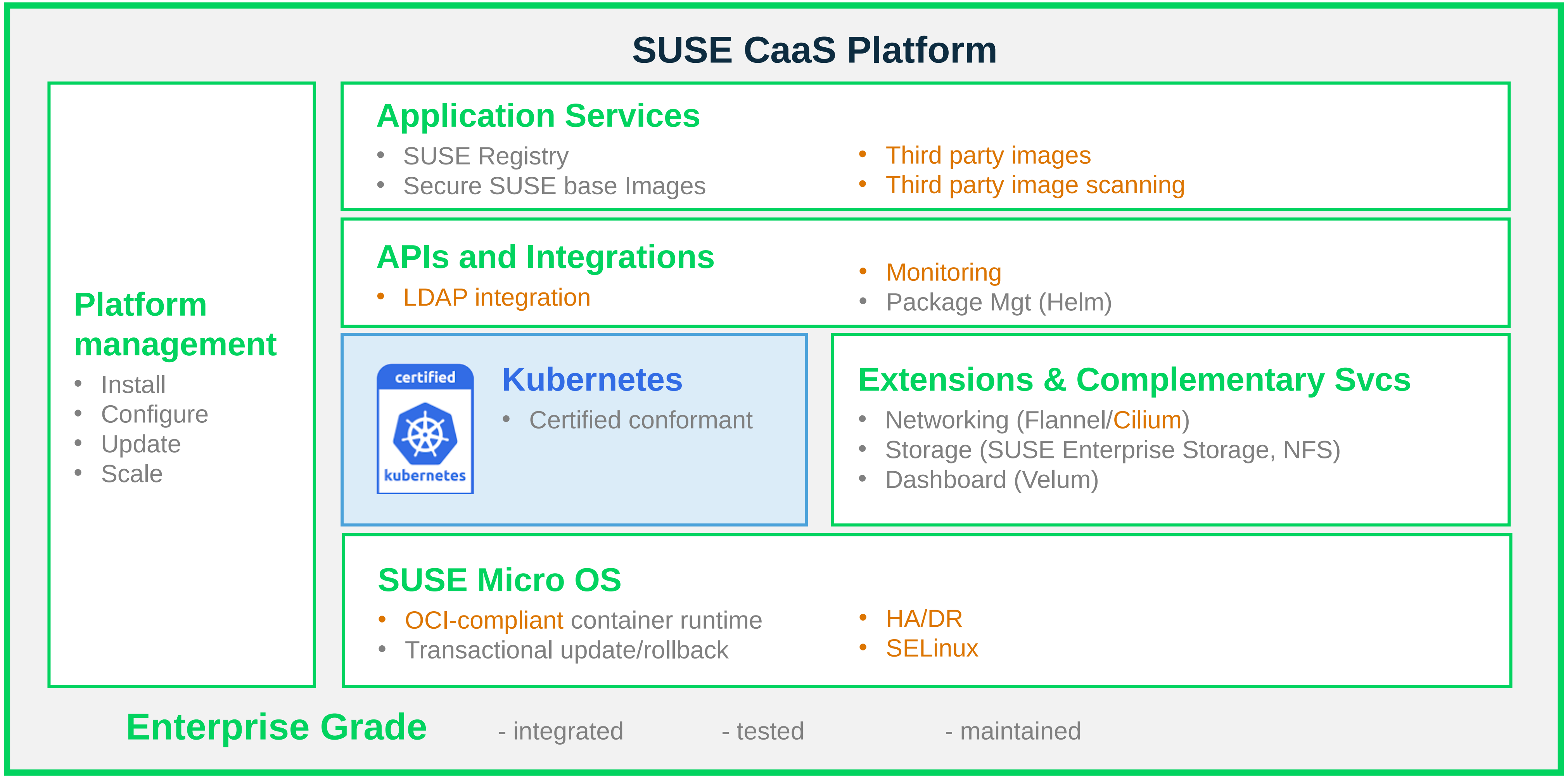
What is SUSE MicroOS?

- 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



SUSE CaaS Platform

Platform management

- Install
- Configure
- Update
- Scale

Application Services

- SUSE Registry
- Secure SUSE base Images
- Third party images
- Third party image scanning

APIs and Integrations

- LDAP integration
- Monitoring
- Package Mgt (Helm)

Kubernetes

- Certified conformant

Extensions & Complementary Svcs

- Networking (Flannel/Cilium)
- Storage (SUSE Enterprise Storage, NFS)
- Dashboard (Velum)

SUSE Micro OS

- OCI-compliant container runtime
- Transactional update/rollback
- HA/DR
- SELinux

Enterprise Grade - integrated - tested - maintained

Future plans

Container Management for the Enterprise

Choose SUSE CaaS Platform to…

Achieve faster time to value

- Enterprise-ready
- Industry leading technologies
- Complete package

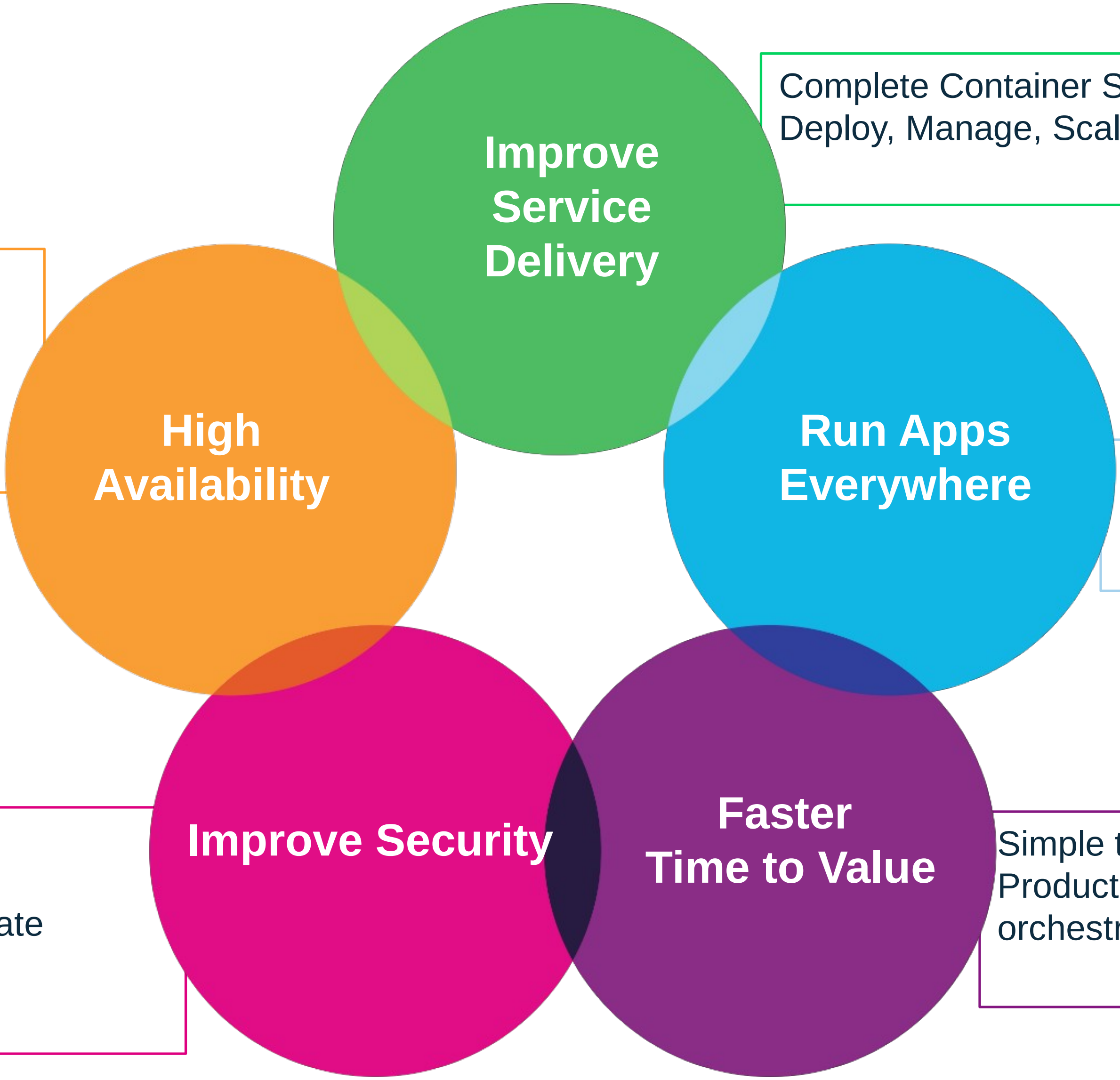
Simplify management and control

- 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”



Improve Service Delivery

Complete Container Solution
Deploy, Manage, Scale

High Availability

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

Run Apps Everywhere

On-premise, On-bare-metal, Private or Public cloud

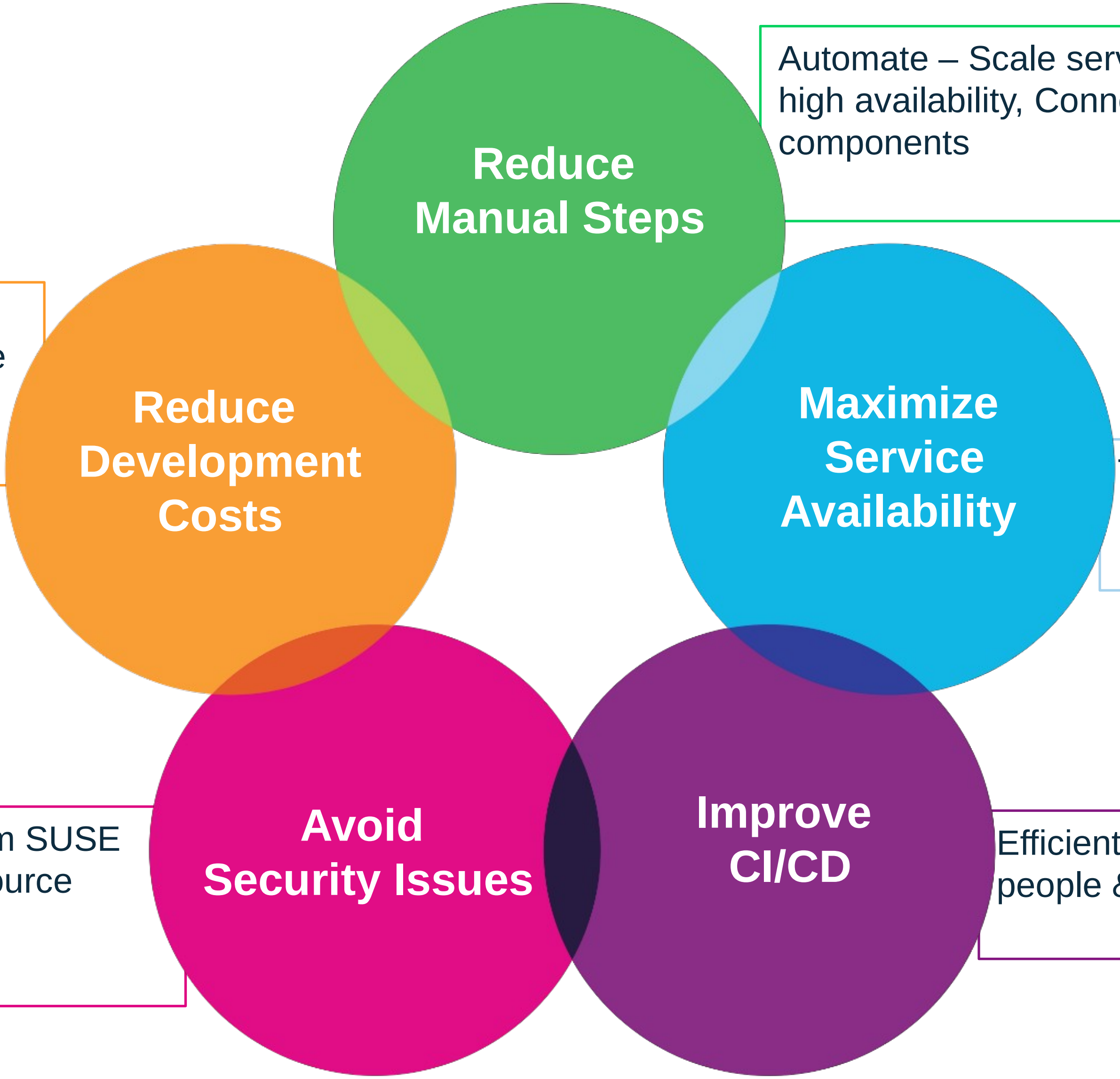
Improve Security

Trusted images from
SUSE Registry,
Security certifications, Uptodate
security patches

Faster Time to Value

Simple to deploy, Open source,
Production grade container
orchestration

How does it “Increase Operational Efficiency”



Automate – Scale services up/down, Select hosts for high availability, Connect storage/networking components

Leverage tools for full container application lifecycle support

Transactional updates
No disruption during upgrade

Signed container images from SUSE Registry - Secure, Trusted source

Efficient resource utilization – Both people & tools.

Partners and Ecosystem

SUSE CaaS Platform

Ready TODAY!

SUSE CaaS Platform Partners

dynatrace

XebiaLabs

SAP

aqua

DATADOG

TYMLEZ

JFrog

MICRO FOCUS

Pachyderm

NeuVector

MINIO

AVI Networks

UNIVA

We adapt. You succeed.

Unpublished Work of SUSE LLC. All Rights Reserved.

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.