

# SUSE

We Adapt. You Succeed.



# SUSE Software-Defined Infrastructure and Application Delivery Approach



## 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



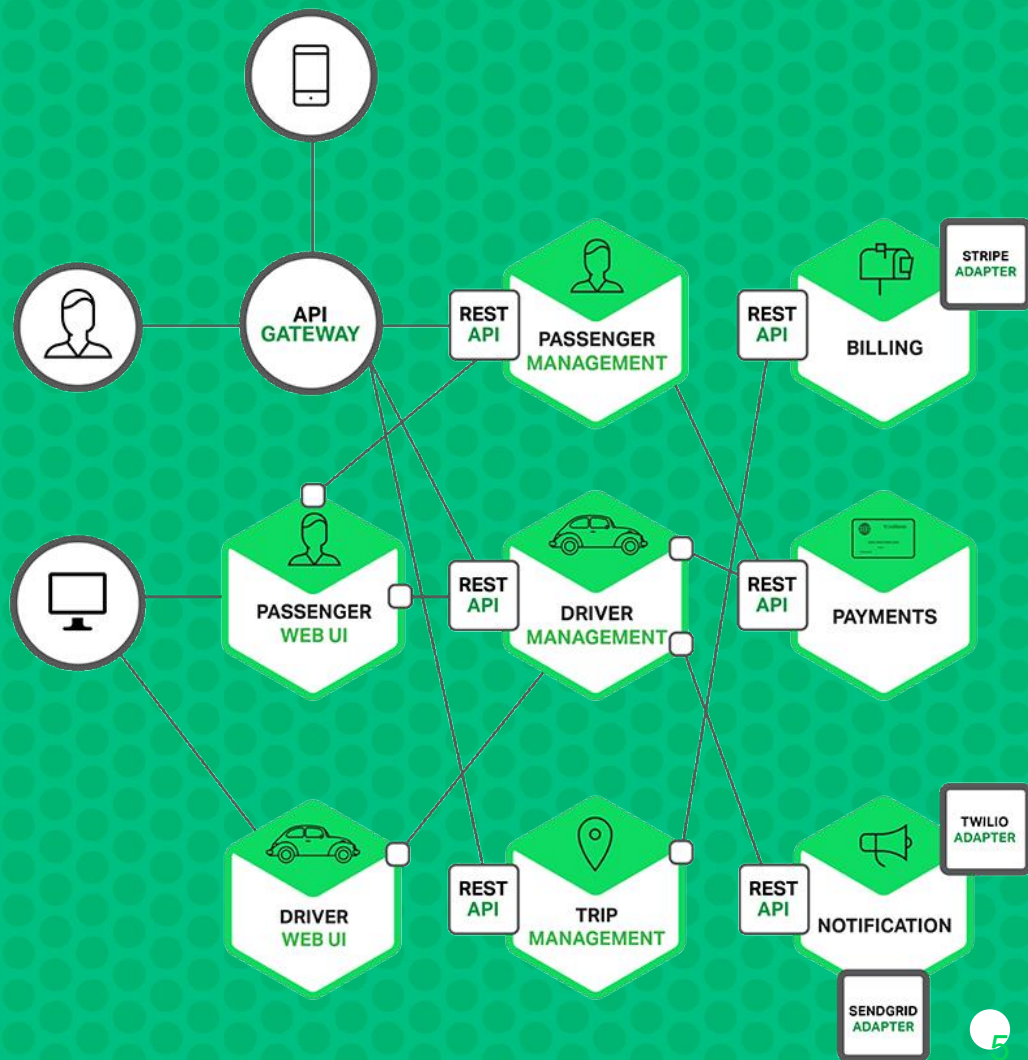
**Public Cloud**  
SUSE Cloud Service Provider Program

**Physical Infrastructure:** Server, Switches, Storage

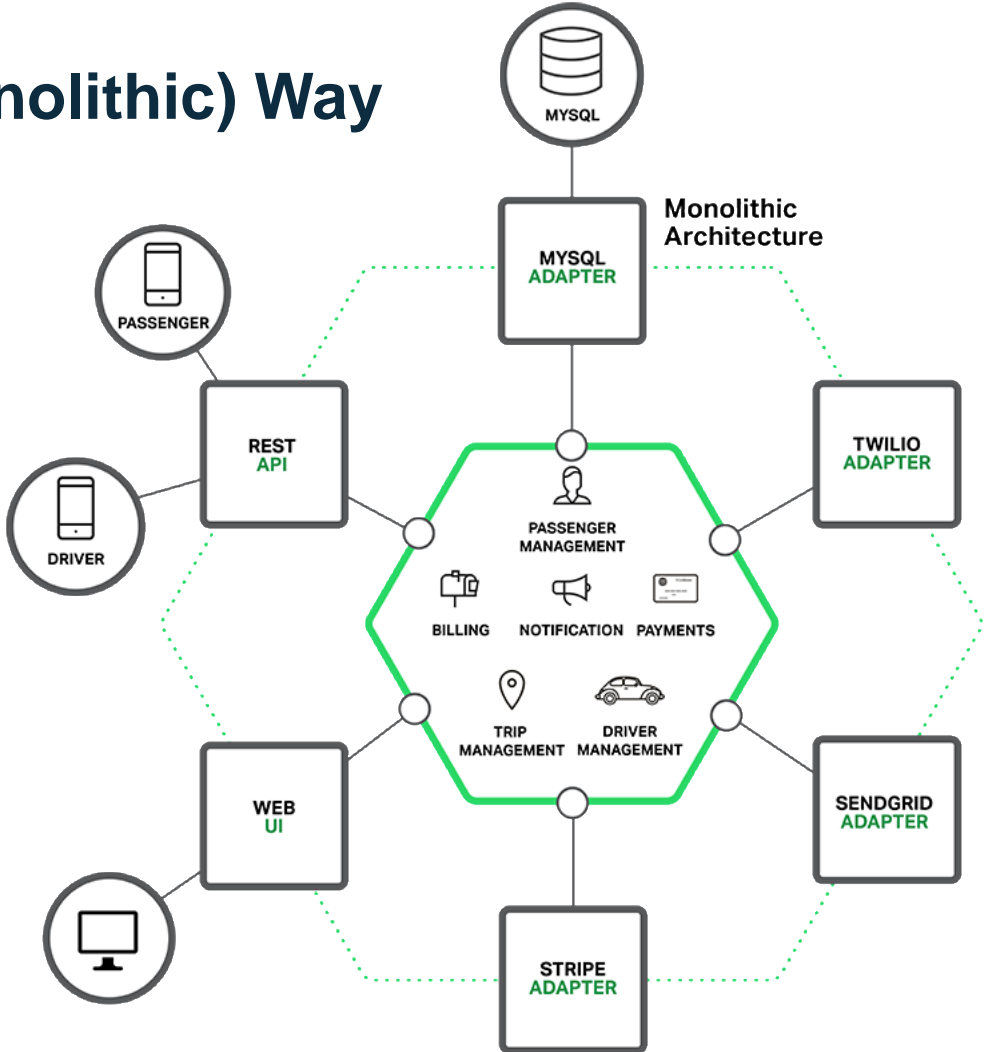
# Collaboration and Contribution

 OpenPOWER™	 CLOUD NATIVE COMPUTING FOUNDATION	 QEMU	 spec	 GNOME™	 openHPC	 OPEN CONTAINER INITIATIVE	 mozilla FOUNDATION
 openstack™	 KVM		 SPACEWALK	 iVISOR PROJECT	 YaST	 openATTIC	 openSUSE.
 HA HighAvailability	 open build service	 OPEN MAINFRAME PROJECT	 X.Org	 Xen™	 THE LINUX FOUNDATION	 ceph	 SAMBA
	 MariaDB	 OPNFV	 openinventionnetwork	 CLOUDFOUNDRY		 ESDA Electronic System Design Alliance	

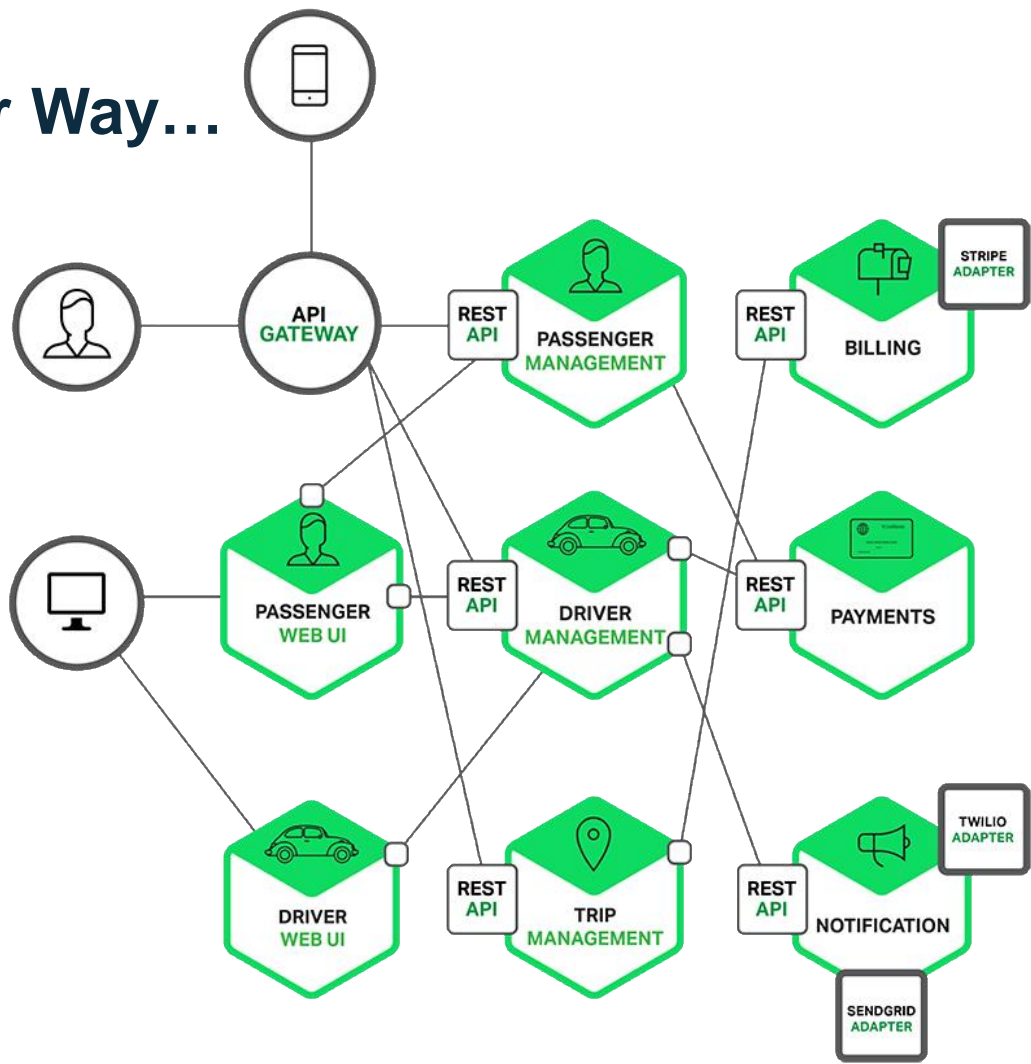
# Microservices



# Old (monolithic) Way

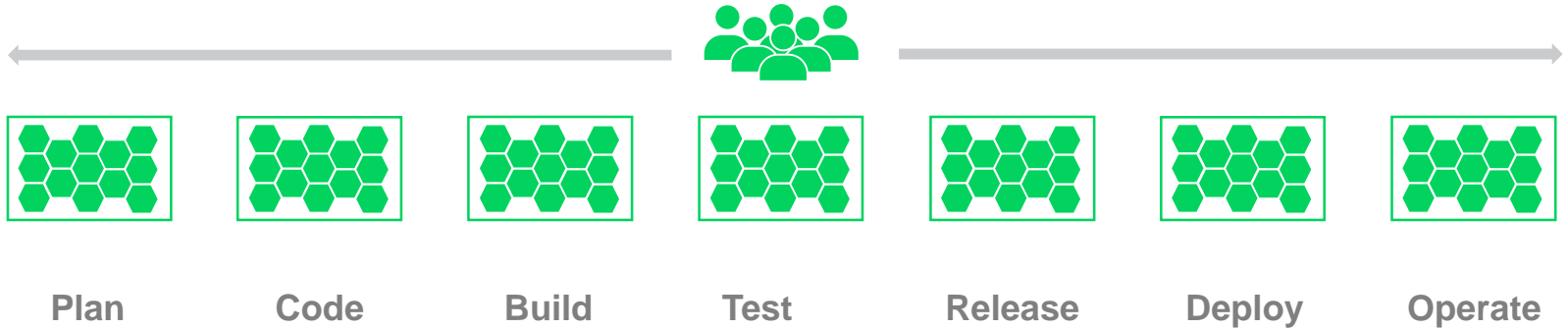


# A Better Way...



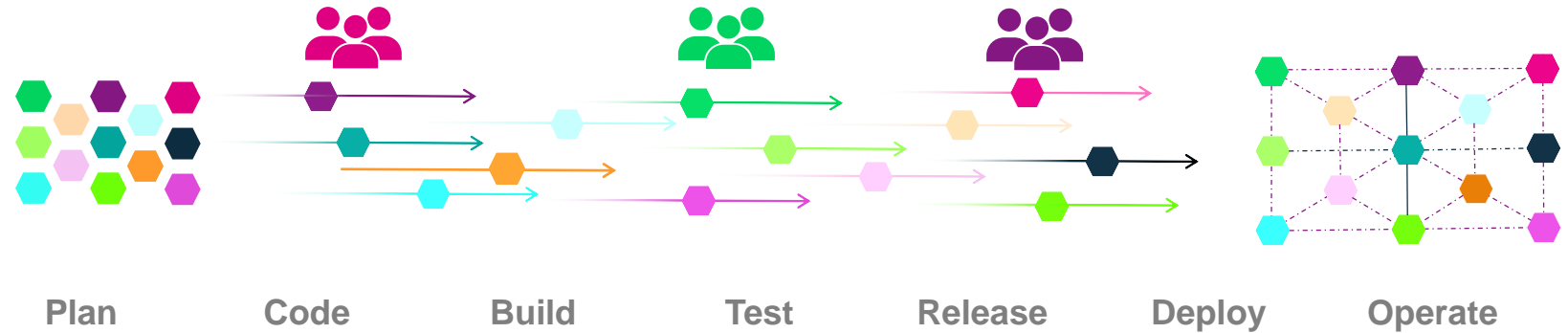
# Build and Deliver Cloud Native Applications

Instead of larger, monolithic services ...



# Build and Deliver Cloud Native Applications

Increase agility with Microservices



Smaller Codebase + Specialized Teams + Deconstructed Services  
= *Agile IT*

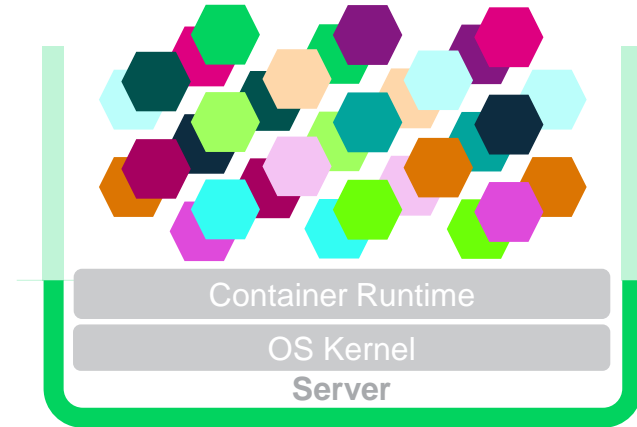
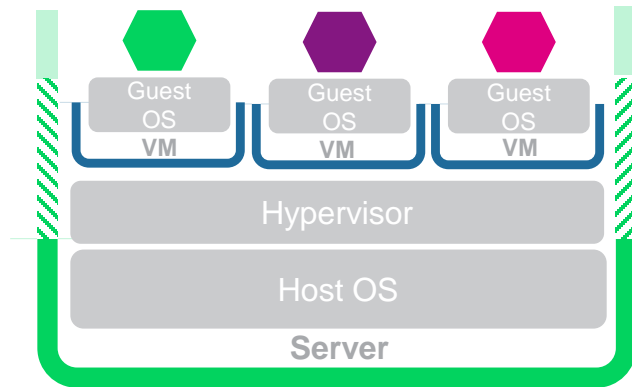
= *Opportunity!*



Containers?

# Build and Deliver Cloud Native Applications

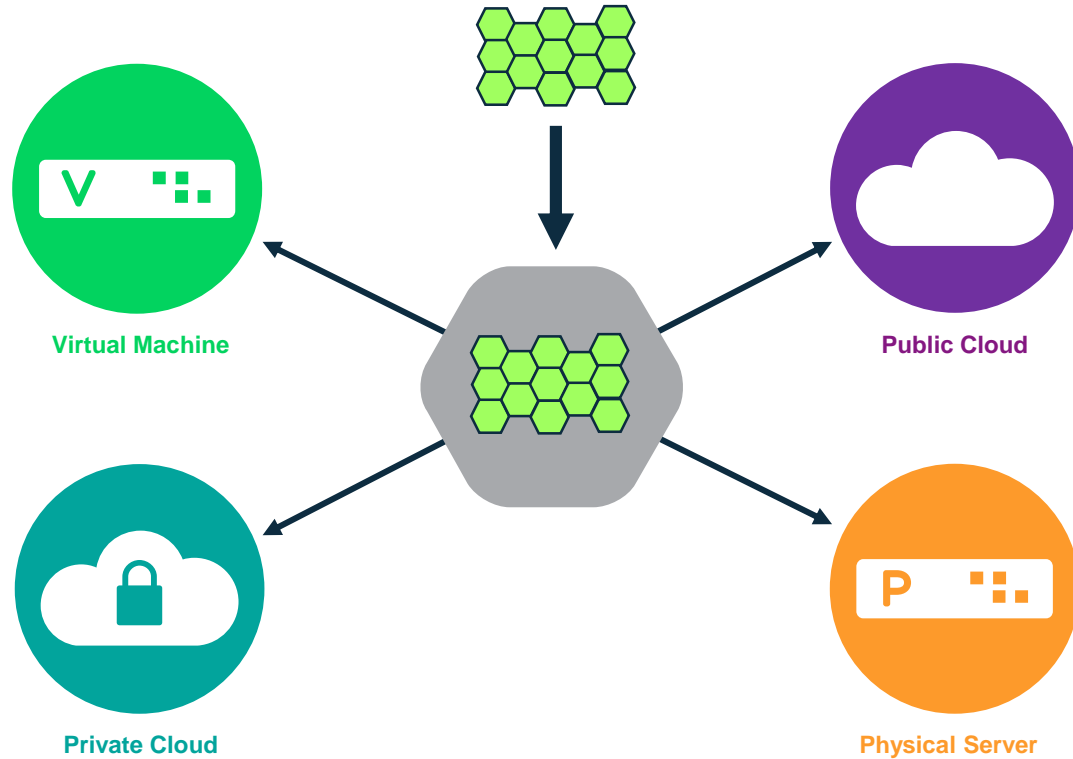
Containers enable microservices model



Large numbers of small containers require efficiency of shared OS Kernel  
Continuous delivery demands fast start and stop capability

# Ease Application Transformation

(Re-)deploy to virtual or cloud infrastructure



# Why a Container Orchestration Tool?

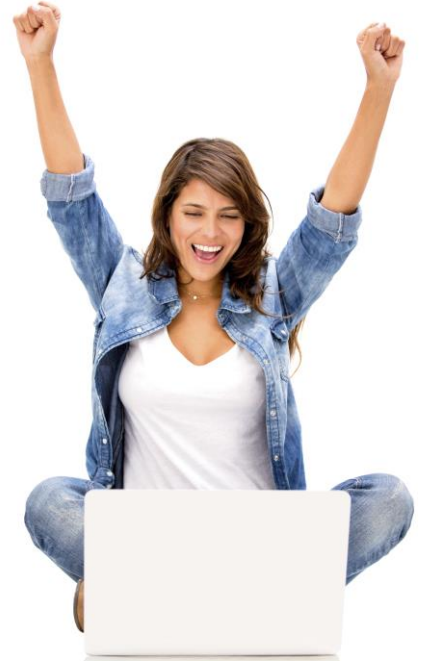
# We've already got it covered for Docker Open Source Project support....



+



=



Docker Open Source Container Engine

SLES

# Deploying at Scale Requires Automation

## Orchestration

- Scheduling
  - Service discovery
- 

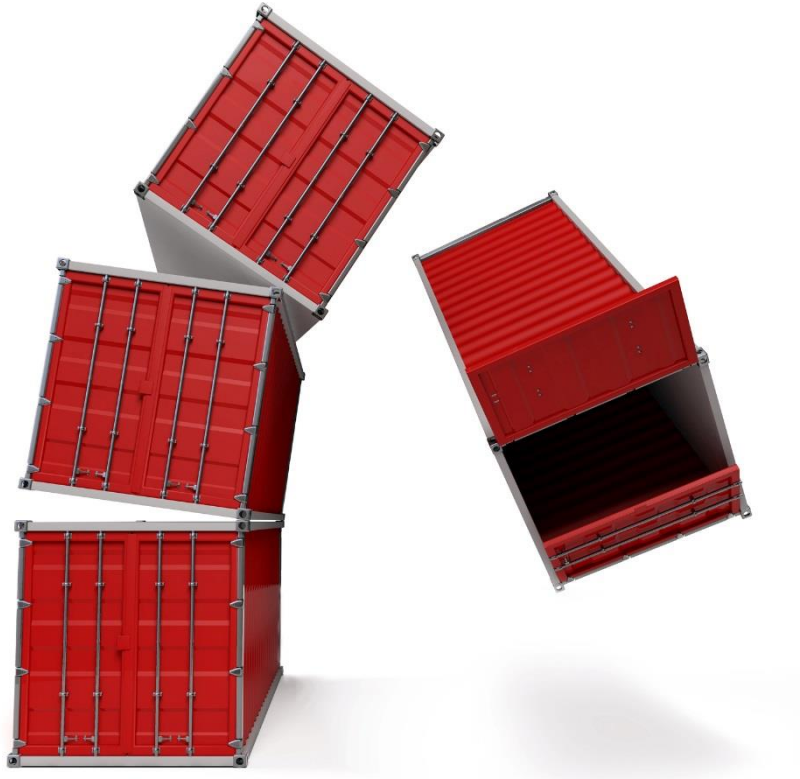
## Performance and availability

- Scaling
  - Load balancing
  - Self-healing
  - Monitoring
- 

## Maintenance

- Rollout
- Rollback





“Building a container stack from the ground up is not for everyone.”

# Organizations Need More Consumable Container Management Solutions



- Easier to get up and running quickly
- Simpler to manage and maintain
- Less risky to embrace



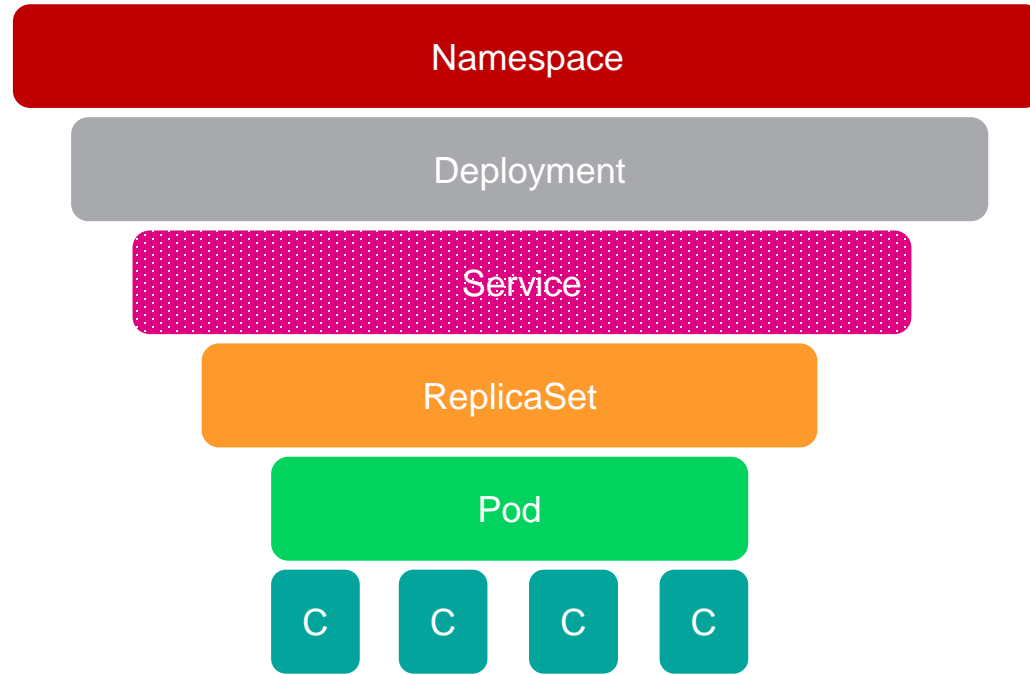


Along Came **kubernetes**



# Kubernetes Technology Deep Dive

# Logical Hierarchy



# Controller/Control Loops

Drive current state -> desired state

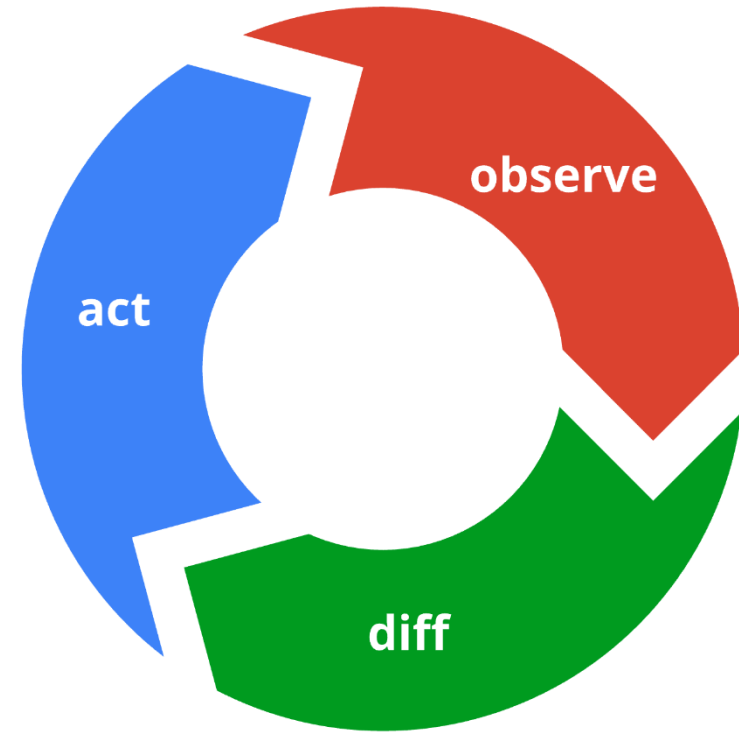
Act independently

APIs – no shortcuts, back doors or hacks

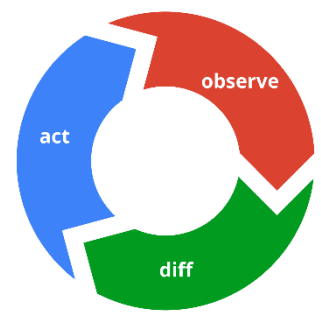
Observed state is truth

Recurring pattern in the system

Example: ReplicationController



# ReplicaSet (formally ReplicationController)



Type of *controller* (control loop)

Ensure N copies of a pod always running

Cleanly layered on top of the core

- All access is through public APIs

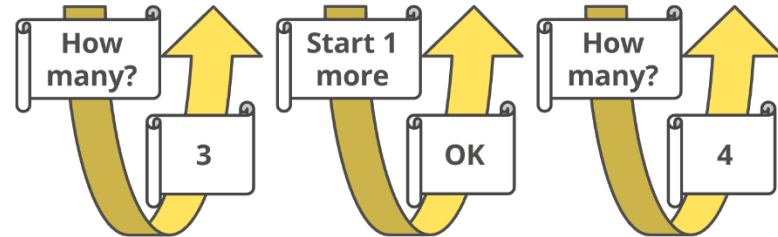
Replicated pods are fungible

- No implied order or identity

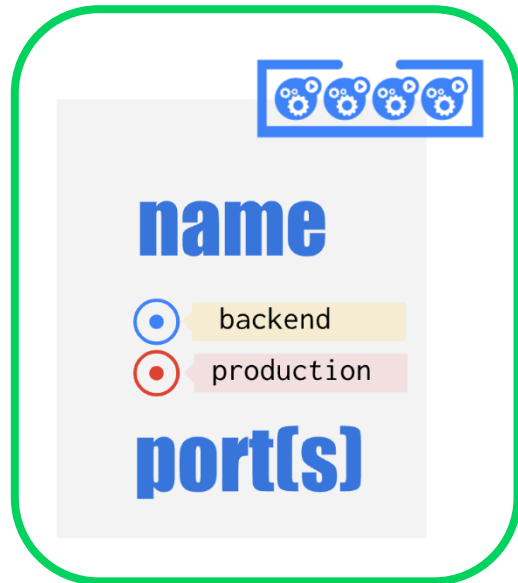
Other kinds of controllers coming soon

## Replication Controller

- Name = "nifty-rc"
- Selector = {"App": "Nifty"}
- PodTemplate = { ... }
- NumReplicas = 4



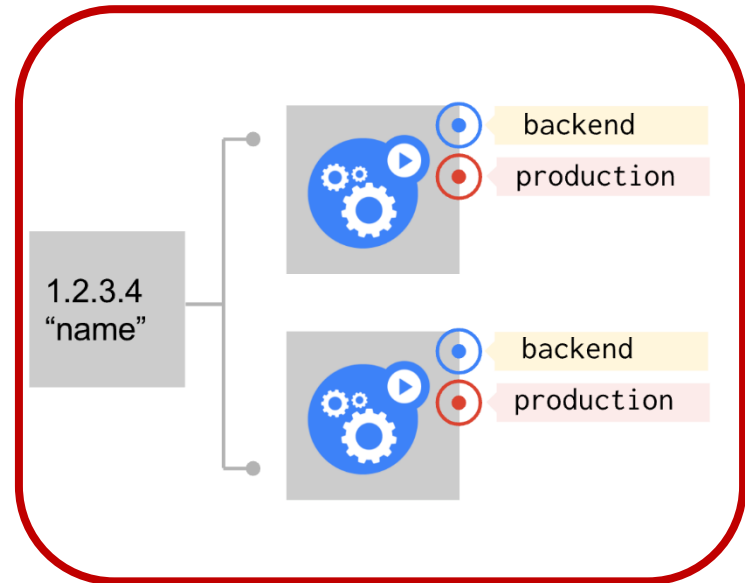
# Services



"Exposed" Service

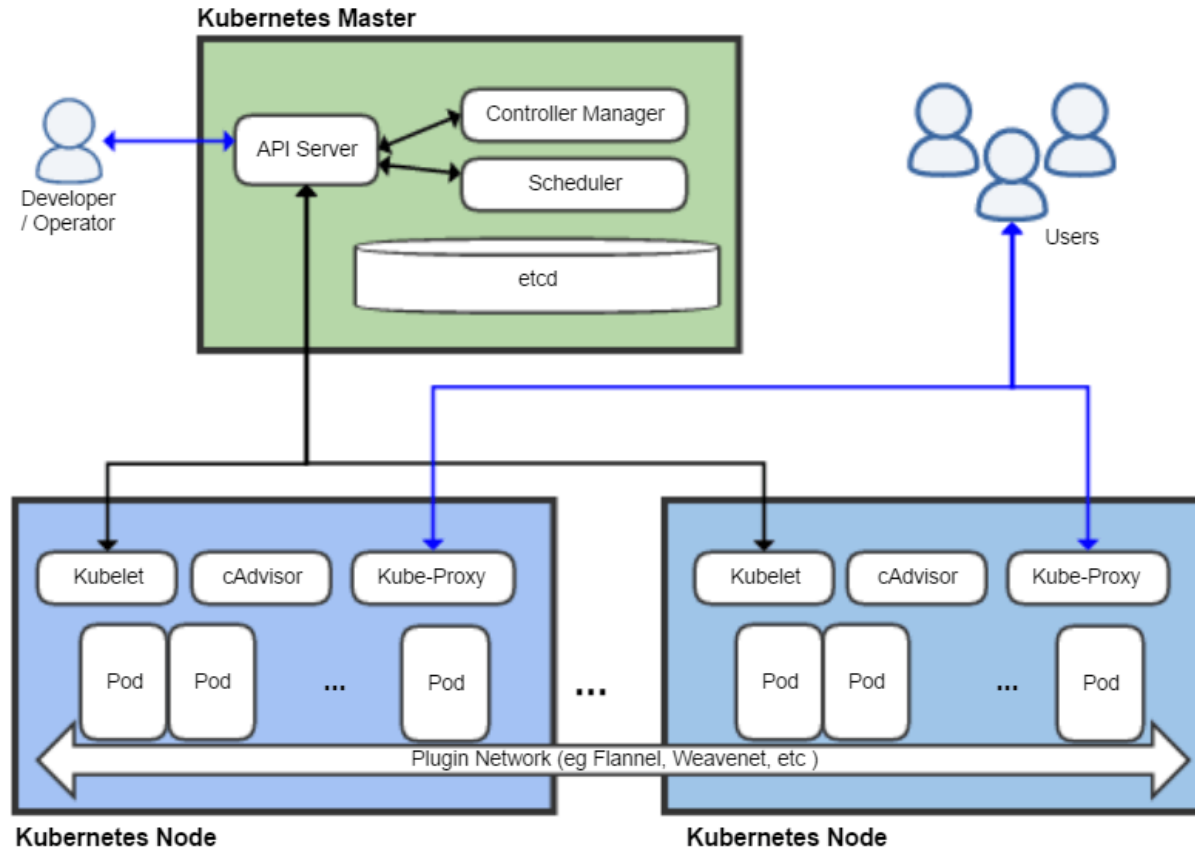


Sends Traffic To >



"Hidden" Underlying Pods

# Component Hierarchy









# SUSE CaaS Platform



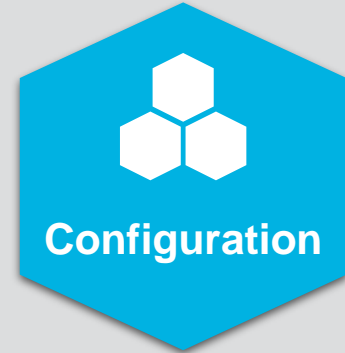
# SUSE CaaS Platform has 3 key components



**Kubernetes**



**SUSE MicroOS**



**Salt,  
Container Engines**

# 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

# Helm

## Package management for Kubernetes

### What is Helm?



- Tool to manage Kubernetes application
- Streamlines installation and management
- It's like 'zypper' for Kubernetes
- Helm has two parts: a client (helm) and a server (tiller).
- Tiller runs inside of Kubernetes cluster, and manages releases (installations) of charts\*
- During **SUSE CaaS Platform** set up the server can be installed on the Kubernetes cluster and then Helm can be used to deploy containerized applications.

---

### Why Helm?

- Ability to deploy applications from SUSE maintained Helm charts or from 3rd party sources
- Official tool to deploy containerized products such as **SUSE Cloud Application Platform**
- Easy to integrate with **SUSE CaaS Platform**

\* "Helm Chart" is the Kubernetes equivalent of an RPM file



# Install and Configure SUSE CaaS Platform



## 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

System Information

Partitioning

\* Standard

Booting

\* Boot Loader Type: GRUB2

\* Enable Trusted Boot: no

\* Status Location: /dev/sda2 ("/")

Network Configuration

\* DHCP / eth0

Kdump

\* Kdump status: enabled

Help

Release Notes...

Install

# 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

Log in

Remember me





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

## Log in to Your Account

Login



# Demo time!

