



12 PAŹDZIERNIKA 2021

Rancher Rodeo w Polsce

Warsztaty on-line i stacjonarne

Introduction



Jan Bruder
Field Engineer,

Rancher/SUSE



Jarosław Biniek
Presales Engineer

SUSE



Objects for the day

1. Explain Docker and Kubernetes concepts
2. Provision HobbyFarm user accounts
3. Install and Demo Rancher Server
4. Deploy a Kubernetes cluster
5. Deploy a hello-world application into the cluster
6. Deploy WordPress into the cluster
7. Technical Discussion, Questions



Forewarning

- This Rodeo Presentation is designed for those that are new to Kubernetes and Rancher
- We will cover some slightly advanced techniques (if we have time)
- What is presented in here is not necessarily how you should run something in production, but should help give you a basis to put together the right tools to be able to run production workloads



Prerequisites

- Up-to-date browser
 - Google Chrome
 - Mozilla Firefox
 - Other Chromium based browsers



Have comments or questions?

Talk to the team by:

- Use the Questions Feature in the GTM Panel
- Joining the Teams meeting during the break



Learning Platform URL

<https://learn.eu.hobbyfarm.io>



Containers & Kubernetes

The Basics



Containers are great!

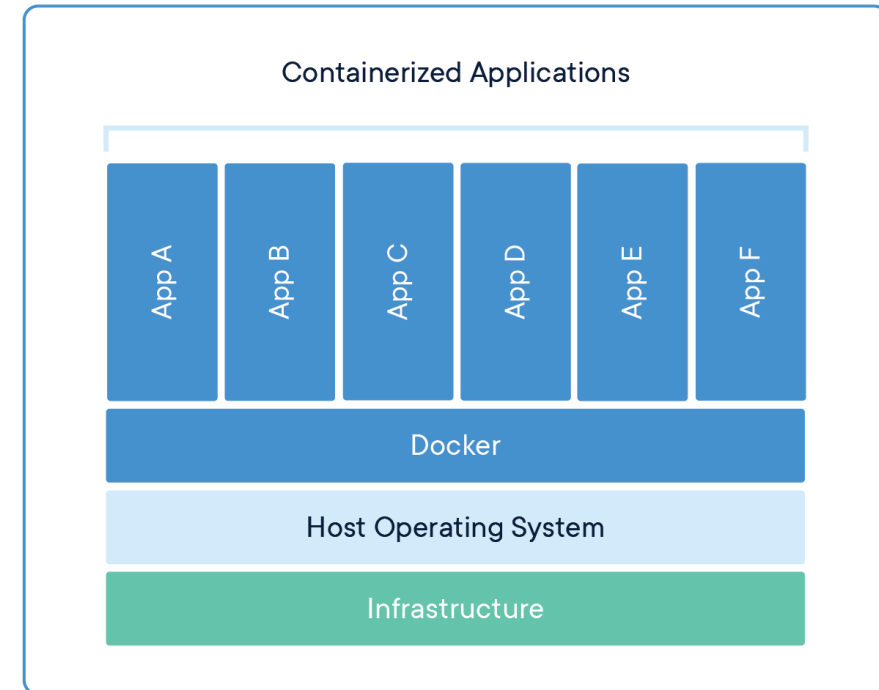
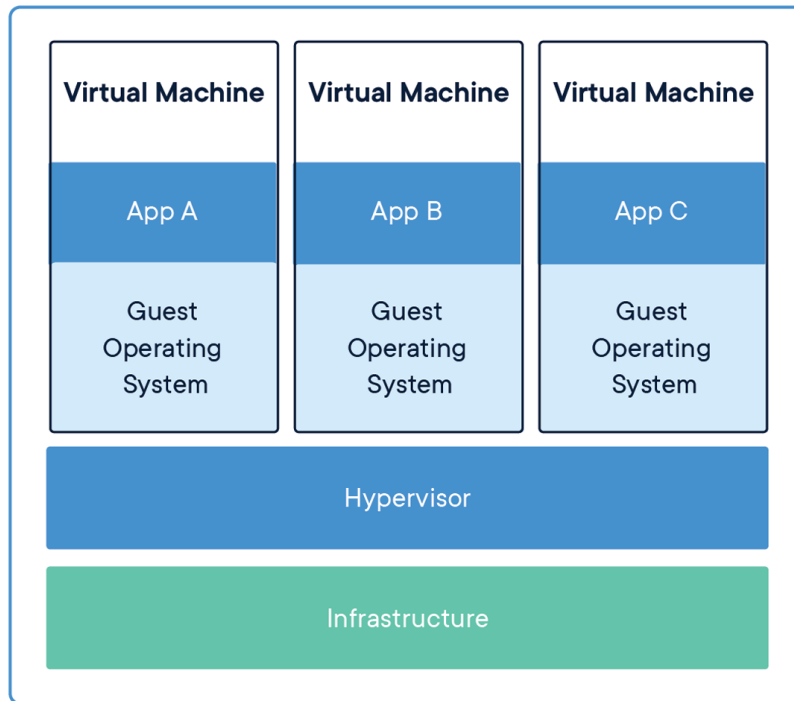


Containers

- A running instance of a container image
- Portable – You build your application once, and you run it anywhere.
- You have a clean, isolated runtime environment for your application.
- You can scale your application horizontally by spawning more containers. Each container is guaranteed to be identical.



Containers



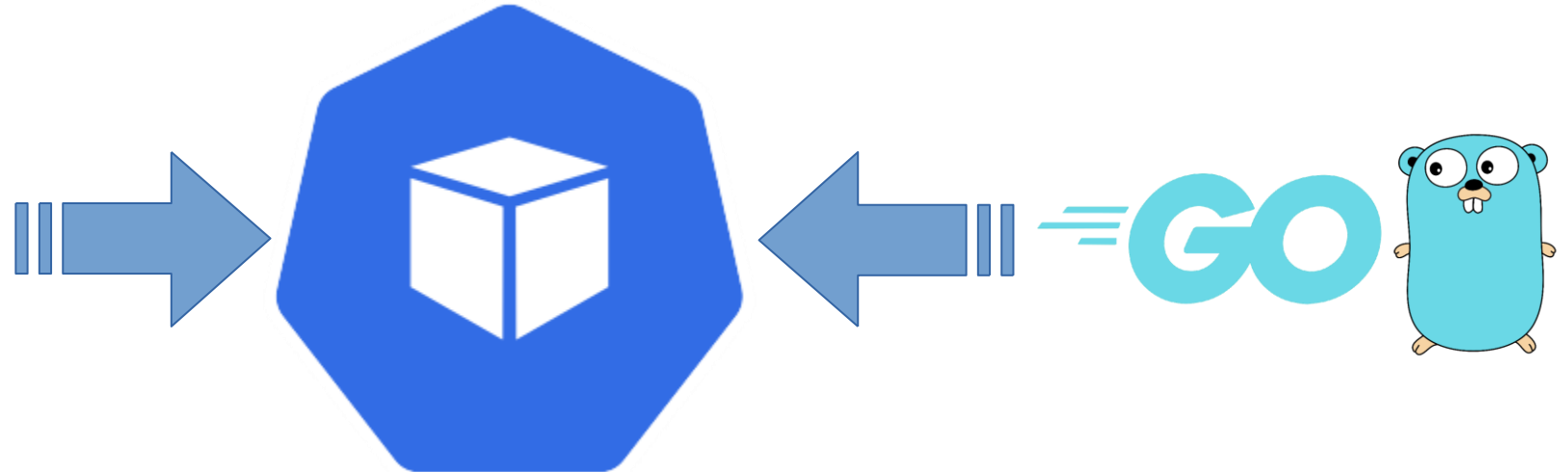
Source : <https://www.docker.com/resources/what-container>



Container Images

```
1 package main
2
3 import (
4     "fmt"
5     "net/http"
6 )
7
8 func hello(w http.ResponseWriter, req *http.Request) {
9     fmt.Fprintf(w, "Hello, world!\n")
10 }
11
12 func headers(w http.ResponseWriter, req *http.Request) {
13     for name, headers := range req.Header {
14         for _, h := range headers {
15             fmt.Fprintf(w, "%v: %v\n", name, h)
16         }
17     }
18 }
19
20 func main() {
21     http.HandleFunc("/hello", hello)
22     http.HandleFunc("/headers", headers)
23     http.ListenAndServe(":80", nil)
24 }
```

Application
Code

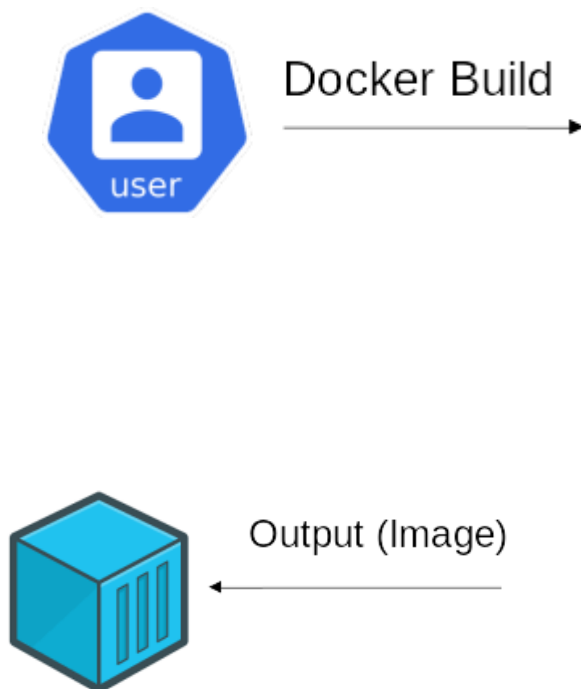


Container
Image

Application
Dependencies



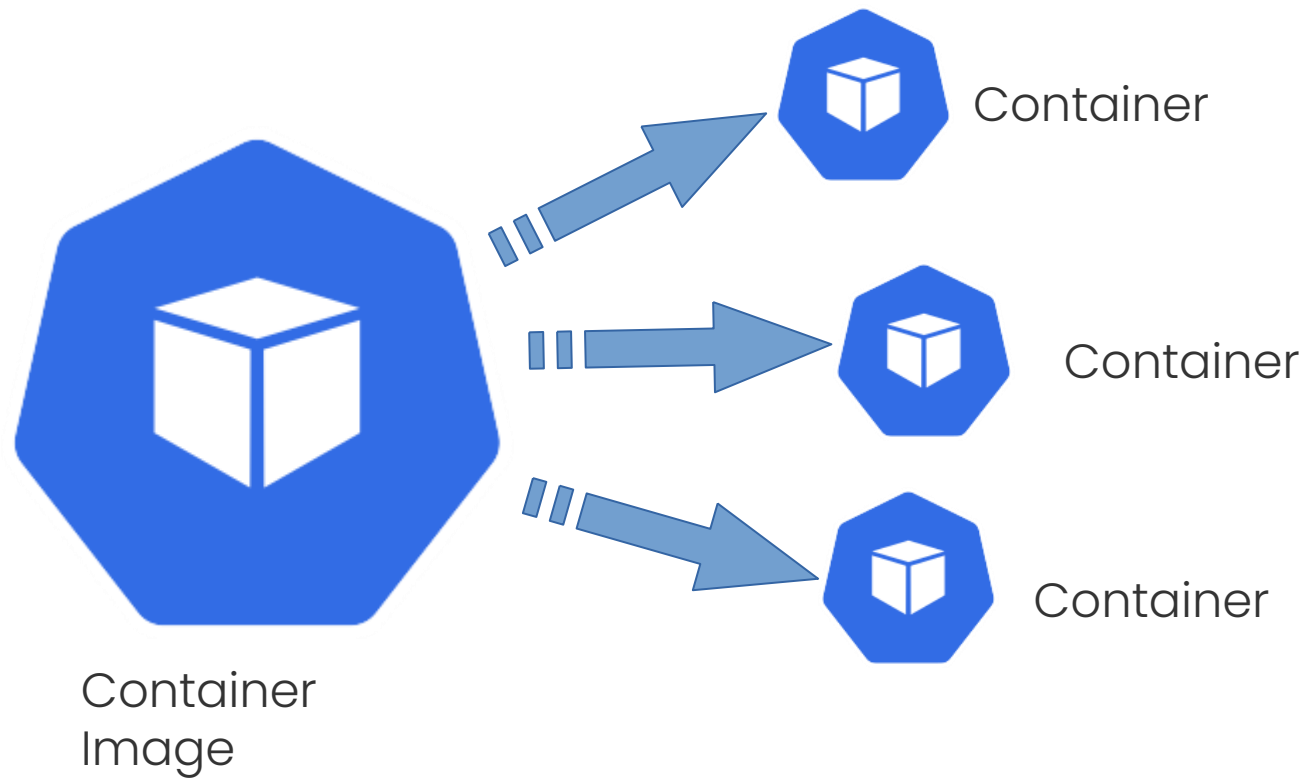
Building an image



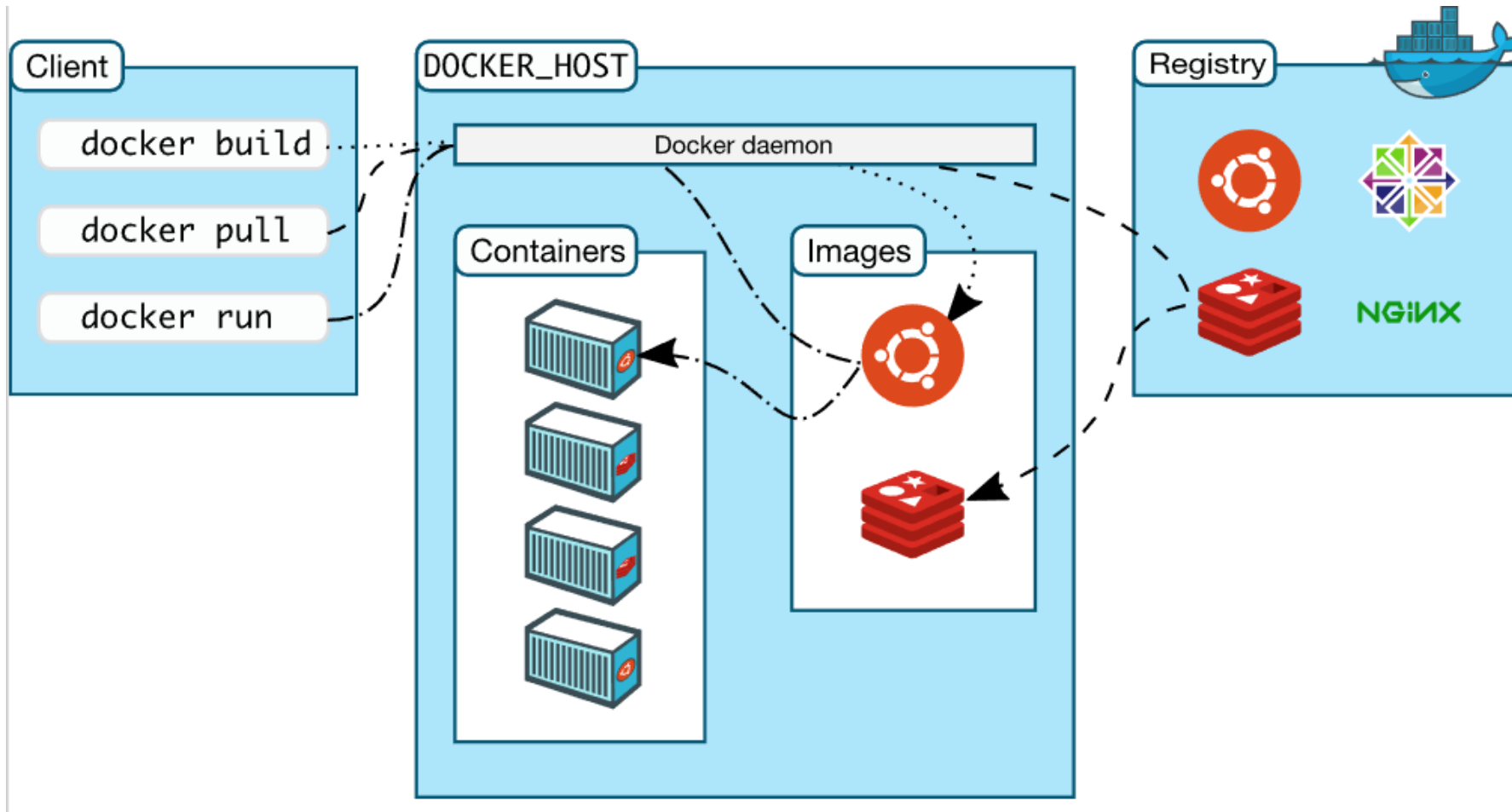
```
1 FROM ubuntu:latest
2
3 LABEL maintainer="someone@someorg.com"
4
5 RUN \
6     apt-get update && \
7     apt-get -y upgrade && \
8     apt-get install -y apache2
9
10 COPY /home/user/src/* /var/www/html
11
12 WORKDIR /root
13
14 CMD ["usr/sbin/apache2ctl", "-D", "FOREGROUND"]
15
16 EXPOSE 80
```



Running an image in a container



Distributing a Container Image

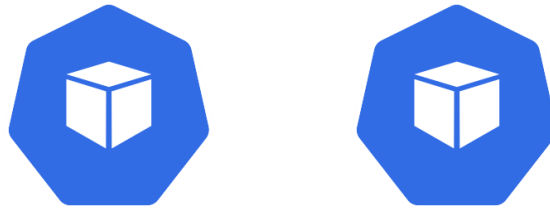


Container Registries

- Central place to store images
 - Public
 - Private
 - On-prem / Cloud
- Examples:
 - Amazon Elastic Container Registry
 - Azure Container Registry
 - Docker Hub
- Some are extended with other capabilities:
 - Vulnerability scanning
 - LDAP / AD Support
 - Auditing
 - Replication
 - ...



Containers are great.....but



Managing a couple – no problem



Containers are great.....but



How about managing
many?

How do we address:

Networking, Security, Scheduling, Automation, etc?





kubernetes



Kubernetes is a powerful open-source platform for managing containerized workloads and services



Declarative Configuration



Everything in Kubernetes is a an API Resource



You can create/retrieve/update/delete resources



Pods



- Smallest unit of deployment
 - Similar to what a VM is in the hypervisor world
- Can contain one or more containers
- Has its own resources:
 - Storage (i.e., shared volumes)
 - Networking
- Scheduled on to a Worker node



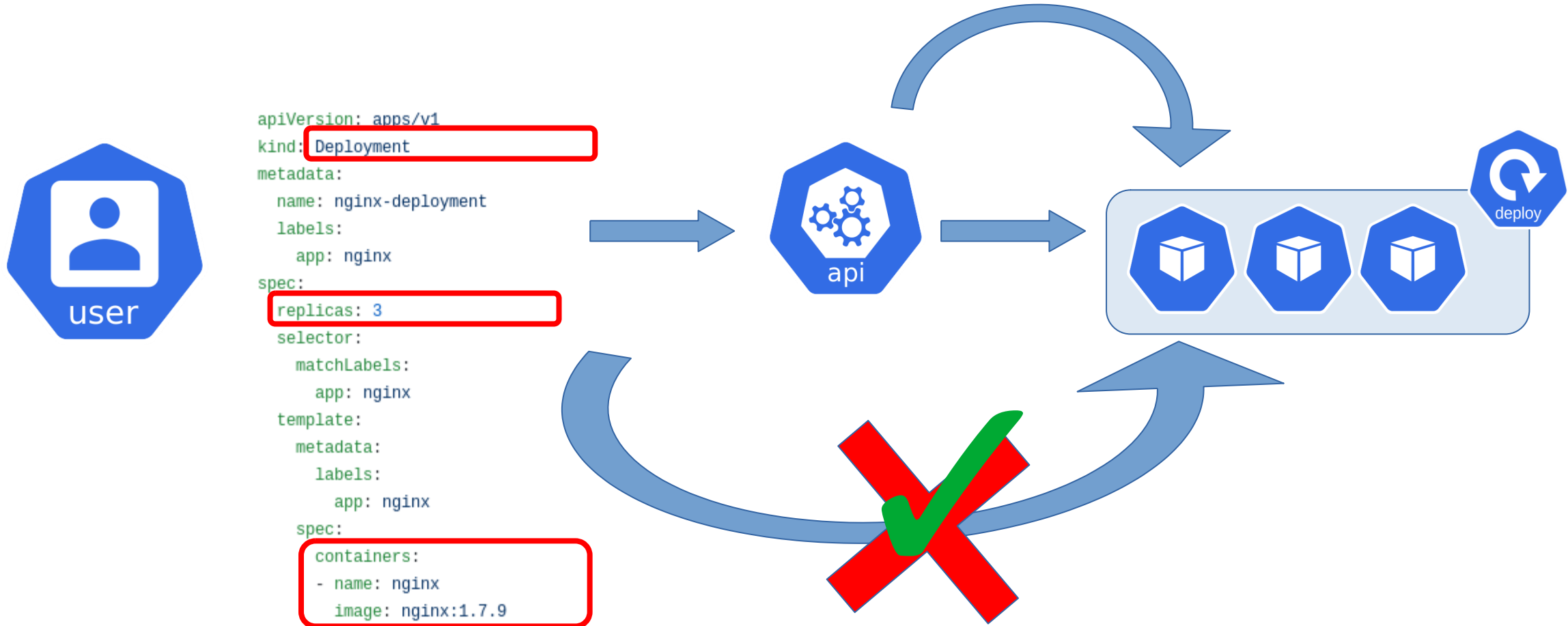
Deployments



- Abstraction for managing a number of identical pods
- Manages these pods as a group
 - Updating
 - Scaling
 - Rollback
- Manages desired state vs actual state



Deployments



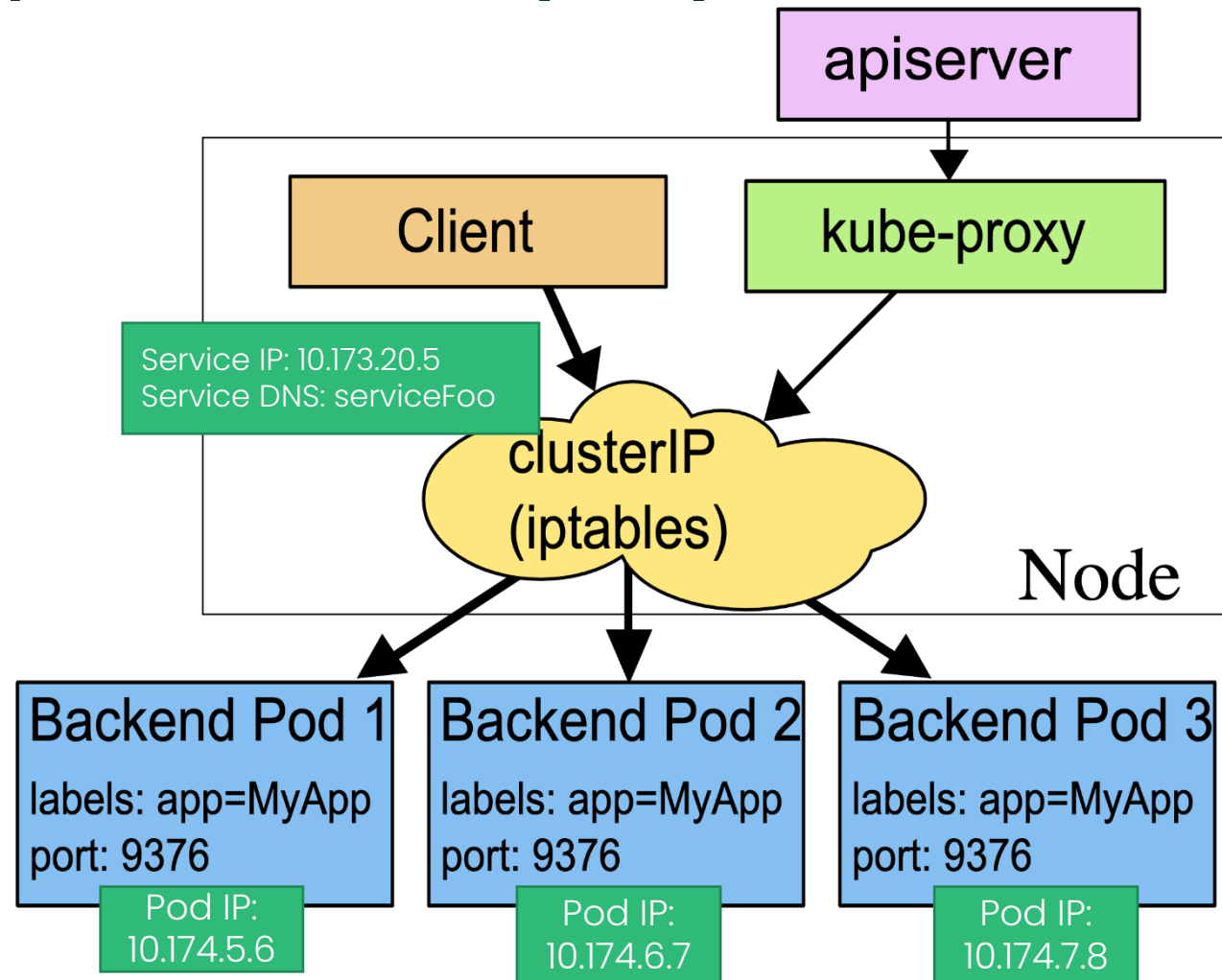
Services



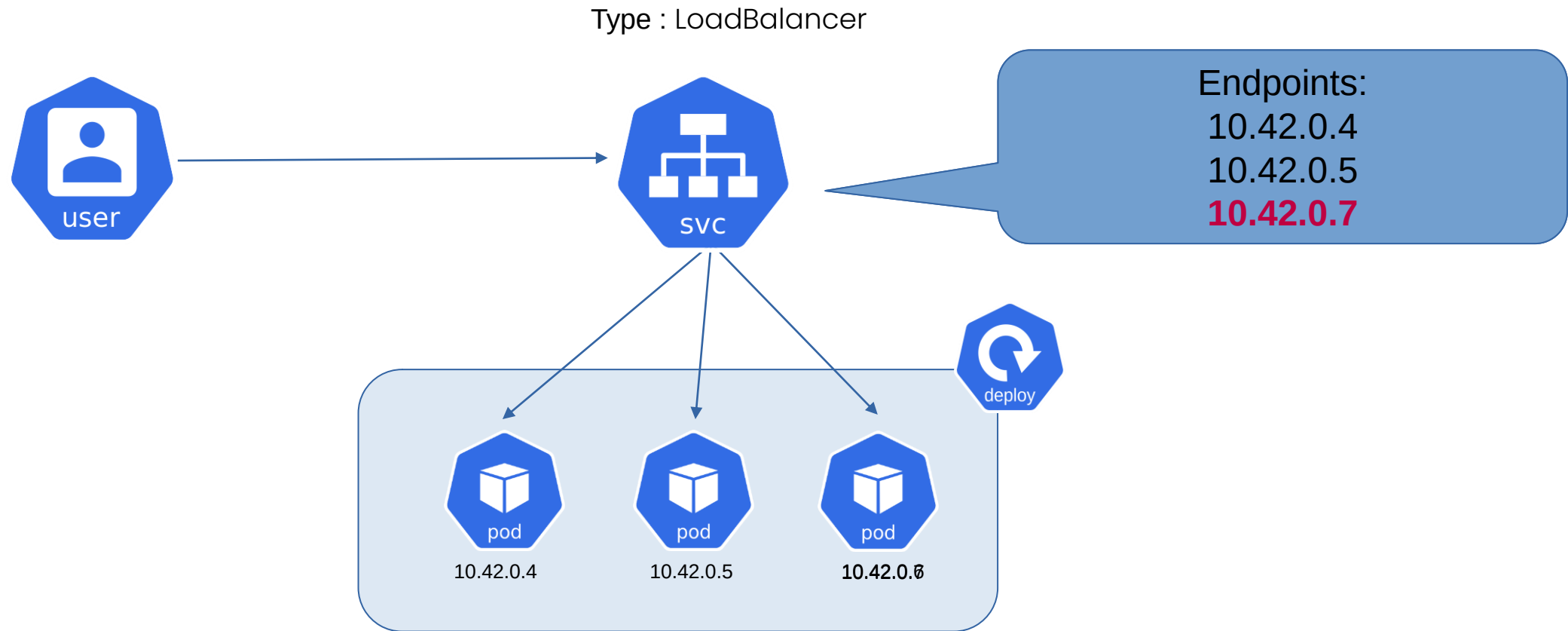
- Exposes an application running on a set of pods
- Reliable, static reference for objects such as deployments
- Has its own IP address / DNS record
- Pods are **mortal** – they have their own life-cycle
 - Don't address them directly
- Services can be Internal (ClusterIP) or External facing (NodePort & Loadbalancer)



Service Type ClusterIP (VIP)



Service Type LoadBalancer



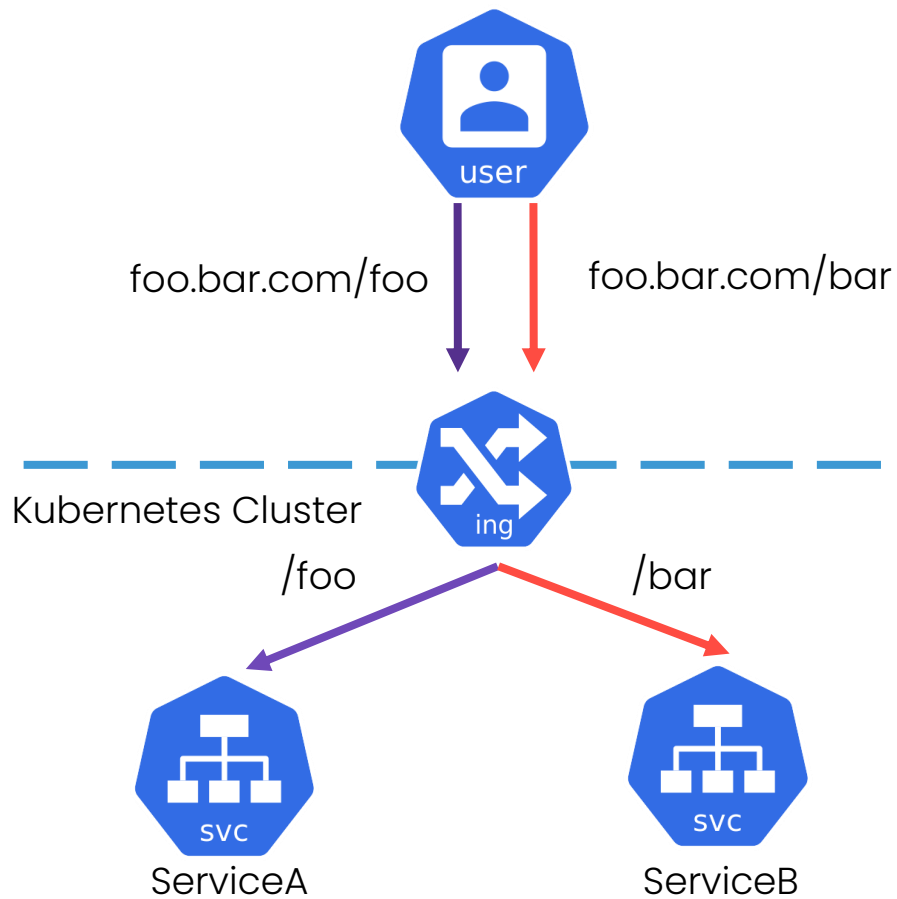
Ingresses



- Manages external access to services in a cluster
 - Commonly HTTP
- Gives services externally-reachable URLs
- Load Balance traffic
- Terminate SSL/TLS
- Facilitate name-based virtual hosting



Ingresses



```
1  apiVersion: networking.k8s.io/v1beta1
2  kind: Ingress
3  metadata:
4    name: simple-fanout-example
5  spec:
6    rules:
7    - host: foo.bar.com
8      http:
9        paths:
10         - path: /foo
11           backend:
12             serviceName: serviceA
13             servicePort: 4200
14         - path: /bar
15           backend:
16             serviceName: serviceB
17             servicePort: 8080
```

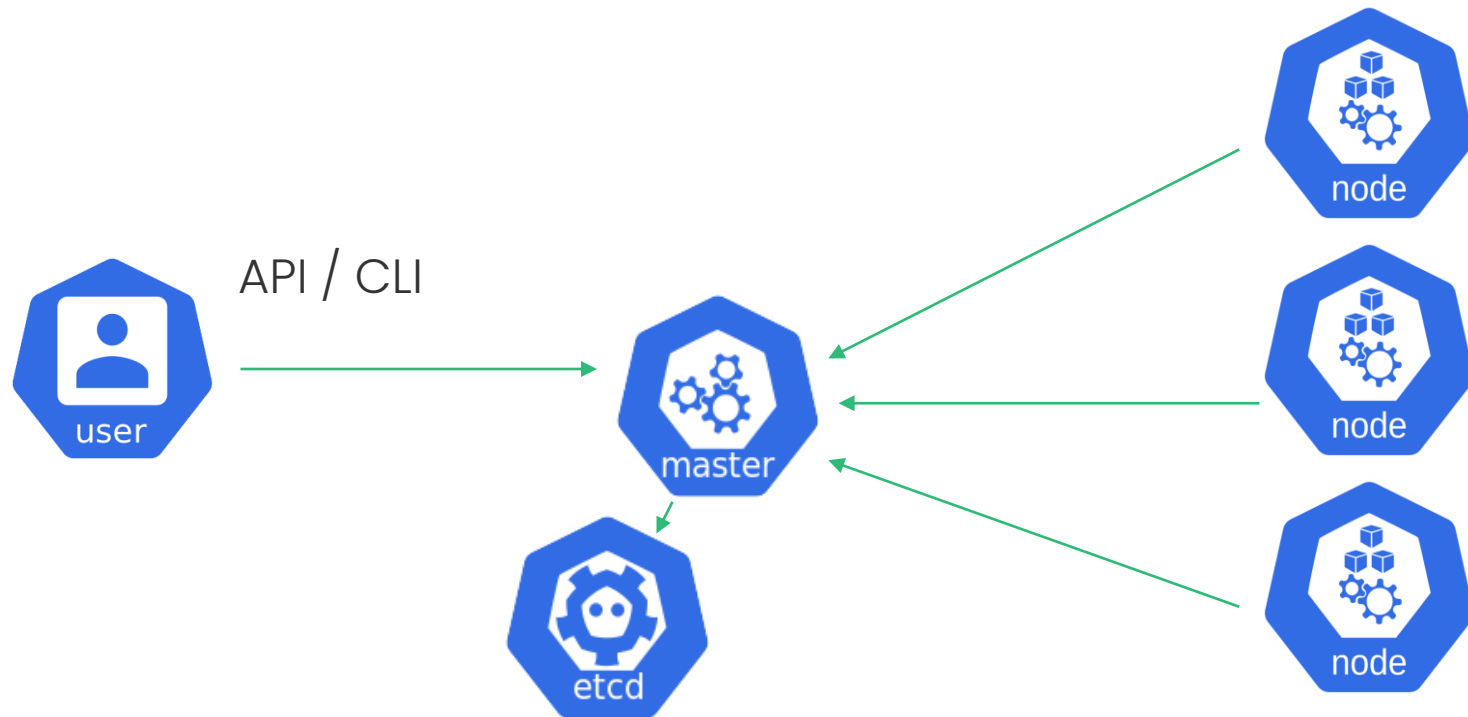


Many more resources ...

- ConfigMaps
- Secrets
- PersistentVolumes
- HorizontalPodAutoscalers
- CronJobs
- DaemonSets
- StatefulSets
-



Kubernetes architecture



- Controlplane: Manages the cluster and exposes an API for control
- Etcd: a key value store used as Kubernetes' backing store for all cluster data.
- Worker: Runs workloads and all of the supporting components.



Setting up Kubernetes is hard

kelseyhightower / [kubernetes-the-hard-way](#) Watch 903

<> Code Issues 53 Pull requests 45 Actions Projects Wiki Security Insights

master [kubernetes-the-hard-way / docs /](#) Go to file Add file ...

 kelseyhightower Update to Kubernetes 1.18.6 ca96371 on 18 Jul 2020 History

..		
images	Update to Kubernetes 1.10.2 and add gVisor support	3 years ago
01-prerequisites.md	Update to Kubernetes 1.18.6	9 months ago
02-client-tools.md	Update to Kubernetes 1.18.6	9 months ago
03-compute-resources.md	Update to Kubernetes 1.18.6	9 months ago
04-certificate-authority.md	Update to Kubernetes 1.15.3	2 years ago
05-kubernetes-configuration-files.md	Update to Kubernetes 1.15.3	2 years ago
06-data-encryption-keys.md	update docs	4 years ago
07-bootstrapping-etcd.md	Update to Kubernetes 1.18.6	9 months ago
08-bootstrapping-kubernetes-controllers.md	Update to Kubernetes 1.18.6	9 months ago
09-bootstrapping-kubernetes-workers.md	Update to Kubernetes 1.18.6	9 months ago
10-configuring-kubectrl.md	Update to Kubernetes 1.18.6	9 months ago
11-pod-network-routes.md	Update to Kubernetes 1.18.6	9 months ago
12-dns-addon.md	Update to Kubernetes 1.18.6	9 months ago
13-smoke-test.md	Update to Kubernetes 1.18.6	9 months ago
14-cleanup.md	Update to Kubernetes 1.15.3	2 years ago



You don't compile Linux from scratch, you use a distribution



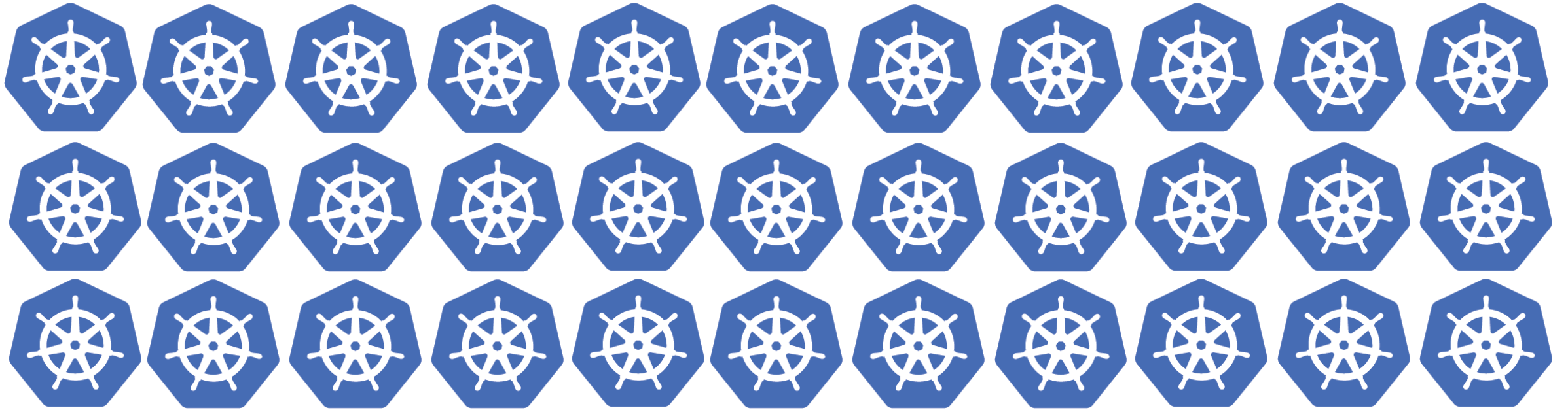
Rancher Kubernetes Engine

- 100% Upstream Kubernetes
- CNCF certified
- Easy installation
- Zero-downtime upgrades
- Backup & Disaster Recovery
- Air gapped installation support



RKE

Kubernetes clusters are great.....but



How about managing
many?

- Different environments
- Different teams
- Different hardware
- Different locations
- Edge devices

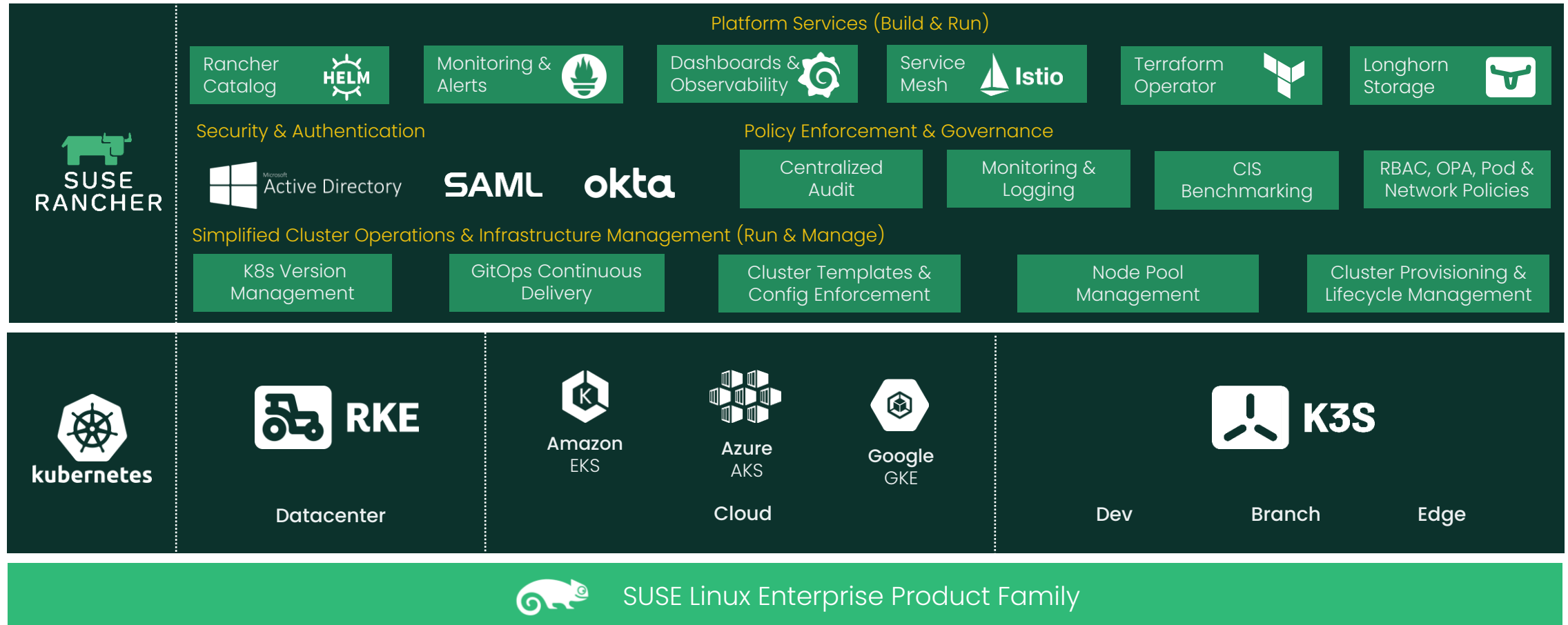


SUSE Rancher – Manage Kubernetes Everywhere

Applications 1

Applications 2

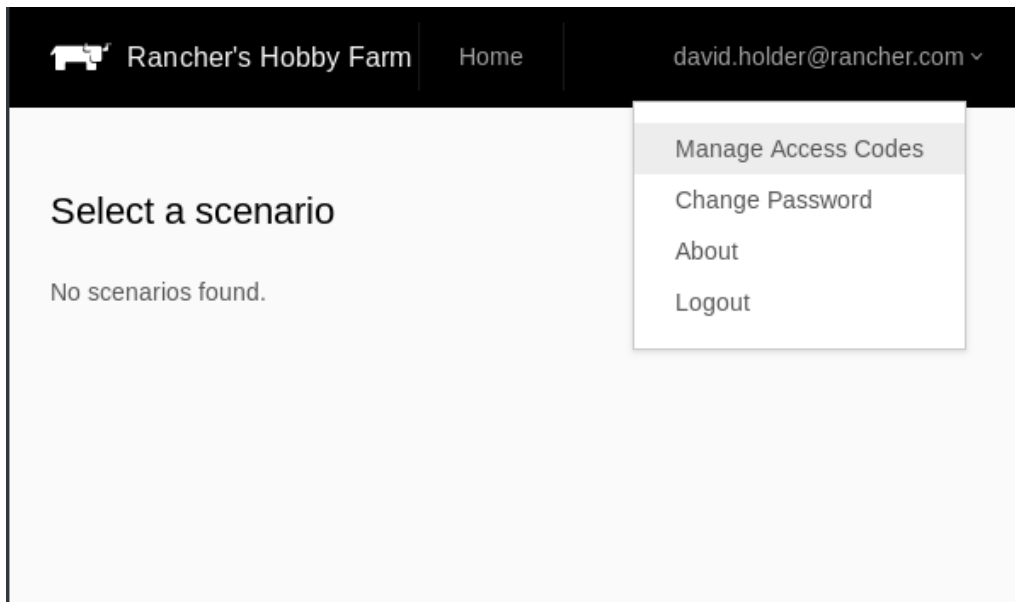
Applications 3



Let's try it out



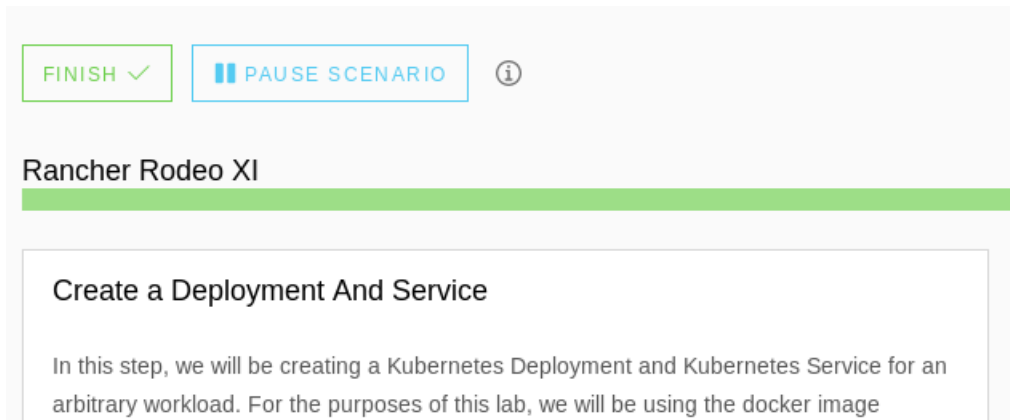
HobbyFarm



- Navigate to learn.eu.hobbyfarm.io
- Register with your e-mail and password
- Access Code: **rodeo1012**
- Note: Browser refresh may be required
- If no scenario shows up, check access code



HobbyFarm Scenario Agenda



Part 1: Rancher Installation

- Generating SSH Keys
- Installing required tools (RKE, Kubectl, Helm)
- Creating the management cluster
- Deploying Cert-Manager and Rancher

Pause for the coffee break at the “Accessing Rancher” step



Coffee break

Until 11:15



Watch on-demand

Online Meetup

SUSE Rancher 2.6 Launch in Poland

Premierowy pokaz SUSE
Rancher 2.6

Czwartek, 23 września 2021 | Spotkanie on-line (meet-up)

godz. 10.00 – 11.00

Rejestracja



Jarosław Biniak
Presales Engineer

SUSE



Marcin Madey
Country Manager

SUSE

Available on-demand: <https://linkd.pl/paedc>



Copyright © SUSE 2021



[WHY RANCHER?](#)

[PRODUCTS](#)

[CUSTOMERS](#)

[RESOURCES](#)

[COMPANY](#)

[GET STARTED](#)

ONLINE EVENTS

Live training options

Introduction to Kubernetes and Rancher online...

Free, online training for Kubernetes and Rancher. Our popular sessions show you the basics of running Kubernetes, including deployments, pods,...

[Register here](#)

Online Meetups

Free, online meetups aimed at users new to Kubernetes, or those interested in using Rancher to easily deploy and manage Kubernetes.

[Learn more](#)

Kubernetes Master Classes

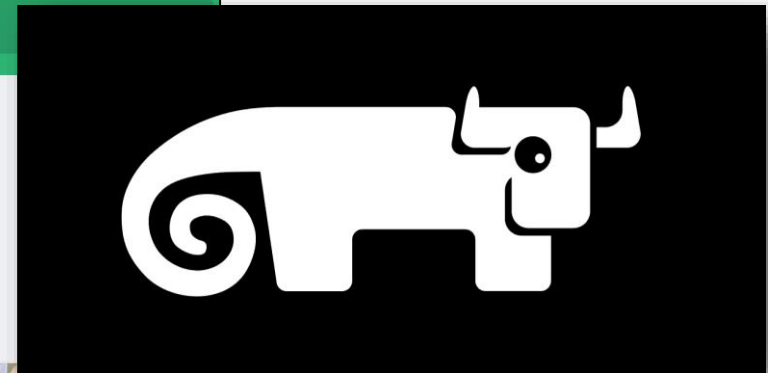
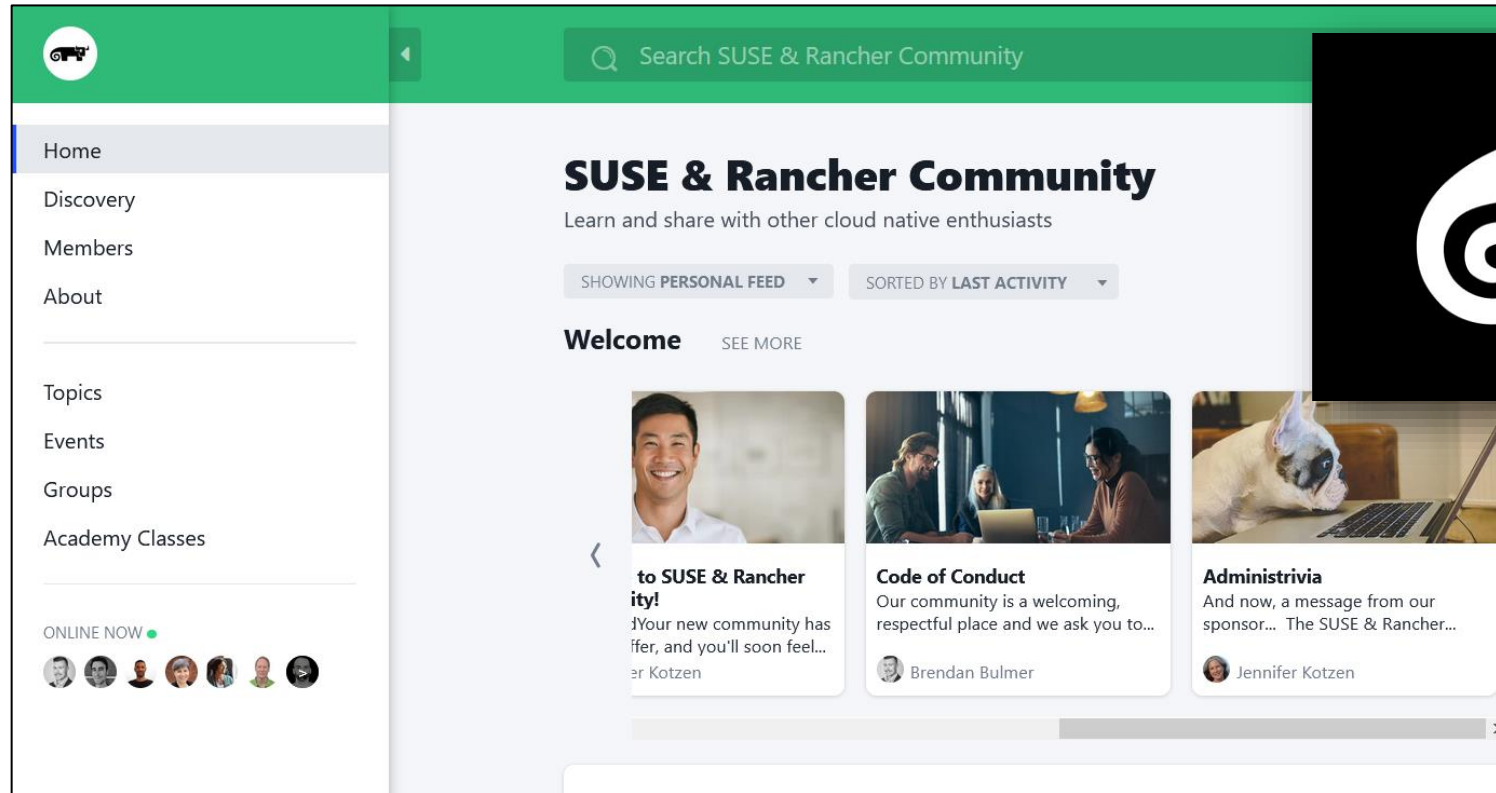
Live, online training sessions on Kubernetes, Rancher and other integrations with presentations from our customers, users, partners, a...

[Find out more](#)



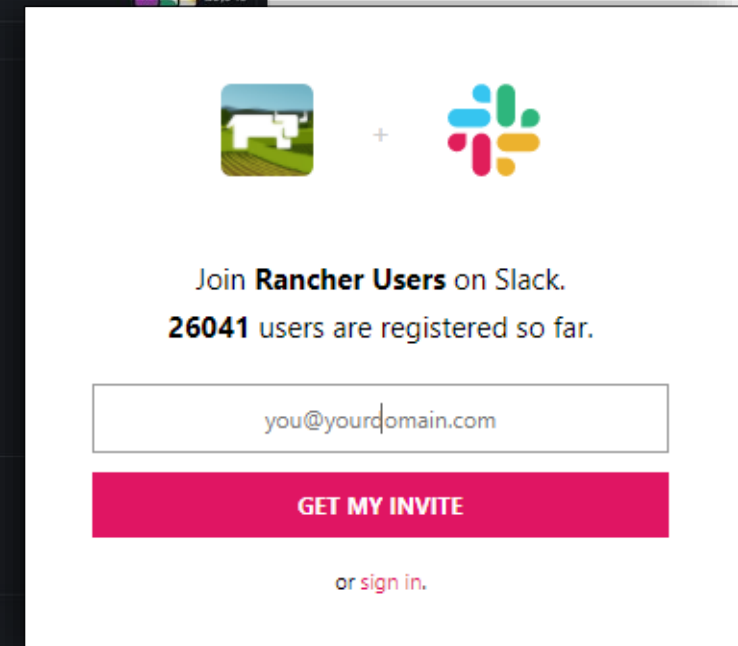
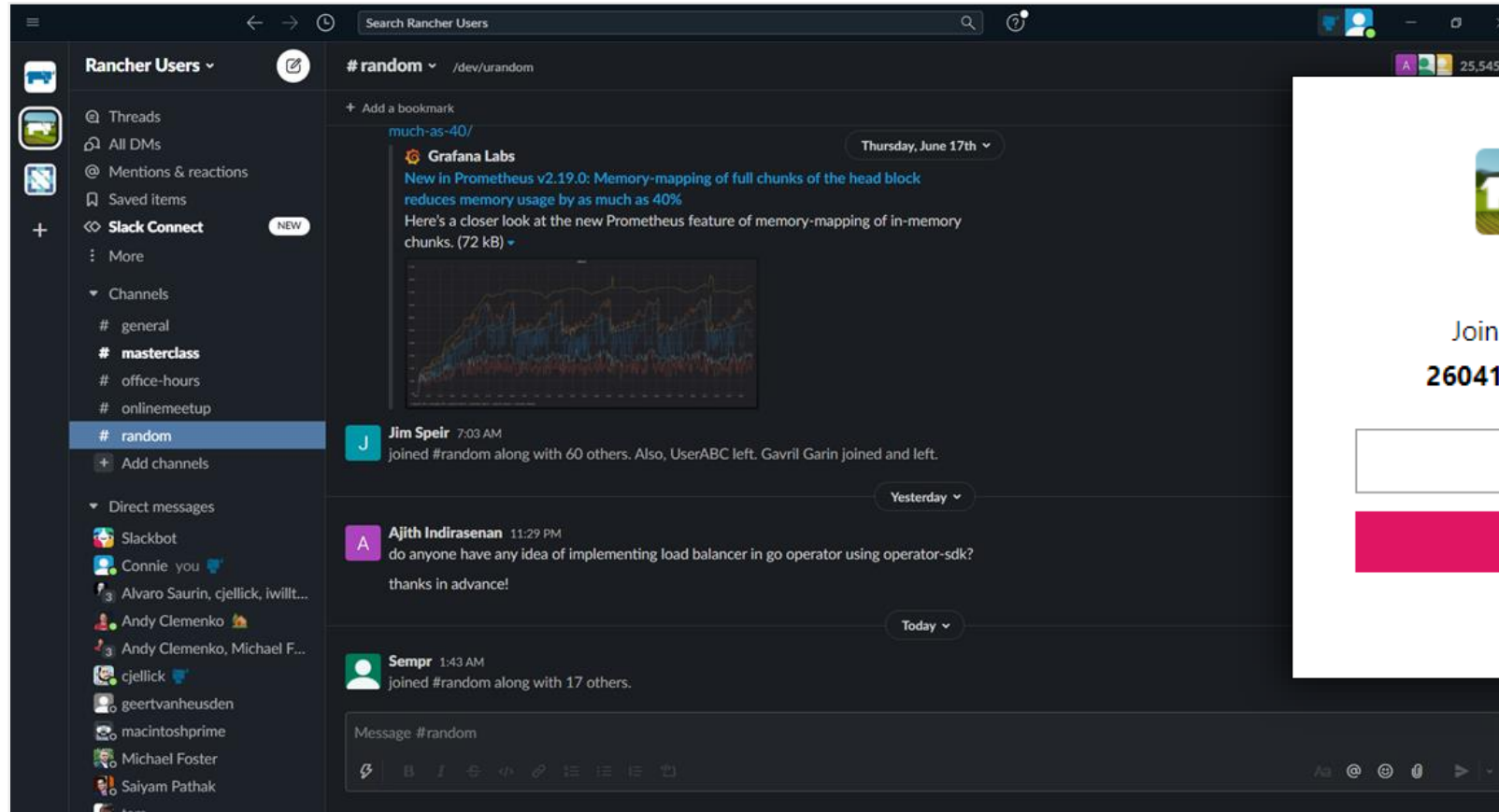
Join the SUSE & Rancher Community

<https://community.suse.com>



Join the Rancher Slack Network

<http://slack.rancher.io>



Copyright © SUSE 2021

 #RancherMeetup



Thank you

For more information, contact SUSE at:

+48 22 537 5020 (Poland)

+49 (0)911-740 53-0 (Worldwide)

SUSE Polska Sp. z o.o.

Postępu 21, 02-676 Warszawa

www.suse.com

© 2021 SUSE LLC. All Rights Reserved. SUSE and the SUSE logo are registered trademarks of SUSE LLC in the United States and other countries. All third-party trademarks are the property of their respective owners.