

Redpill
Linpro

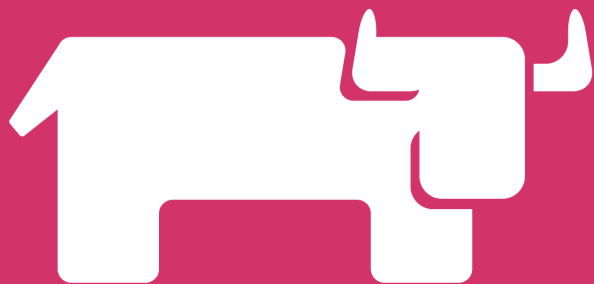


Kubernetes without the cloud

by Christian Nilsson



KUBESPRAY



RANCHER®



K3S

Who am I?

Name: Christian Nilsson

Age: 38

Occupation: DevOps Engineer @ RedPill Linpro

- Worked with IT for about 20 years.
- Developer turned Linux SysAdmin / DevOps Engineer.
- Focusing on Docker, Kubernetes and CI/CD workflows.
- Installed Kubernetes in many different ways.



Kubernetes in the cloud

- Amazon AWS
- Google Cloud Platform
- Microsoft Azure
- Digital Ocean
- IBM Cloud
- Alibaba Cloud
- OpenStack (Magnum)
- ...And more

Tools for Kubernetes in the cloud

\$ cli-tool create cluster mycluster

- Terraform
 - Providers for all major cloud providers.
 - Ansible
 - Modules for all major cloud providers.
 - Cloud provider CLI tools
 - Amazon AWS
 - eksctl
 - Google cloud platform
 - Gcloud
 - Microsoft Azure
 - az
 - OpenStack
 - Magnum plugin for OpenStack CLI
-

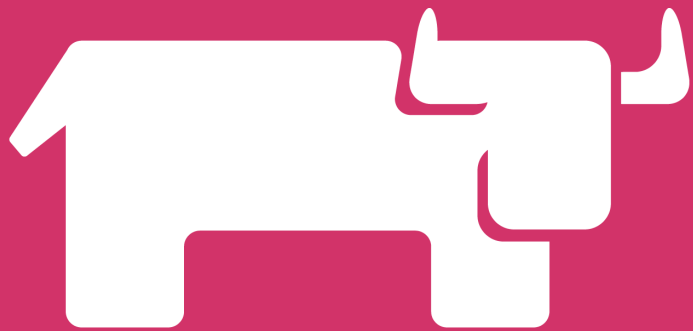
Kubernetes without the cloud

- Datacenter
 - On-prem
 - “Non cloud” VPS
 - Glesys
 - Linode
 - Ipeer
 - etc.
 - Gray box in your mother's basement
 - (...or a cloud VPS)
-

Tools for Kubernetes without cloud

In this talk

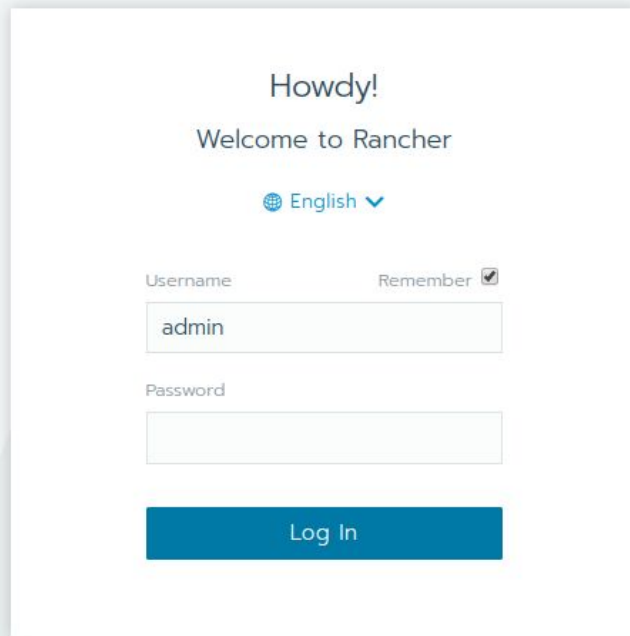
- **Rancher**
 - Web based installer
- **Kubespray**
 - Ansible playbook
- **K3S**
 - Kubernetes distribution



RANCHER®

Rancher

- Web based installer
- Support for both Cloud and non cloud environments.
- Multiple clusters
 - Support existing clusters
- Basic monitoring in web browser
- Basic logging in web browser
- Exec into pods in web browser
- Deploy applications from web interface.

The image shows the Rancher login page. At the top, it says "Howdy!" followed by "Welcome to Rancher". Below this is a language selector showing "English" with a dropdown arrow. The login form has two fields: "Username" and "Password". The "Username" field contains the text "admin". To the right of the "Username" field is a "Remember" checkbox which is checked. Below the "Password" field is a "Log In" button.

Howdy!

Welcome to Rancher

English ▼

Username Remember ☒

admin

Password

Log In

Rancher

Requirements

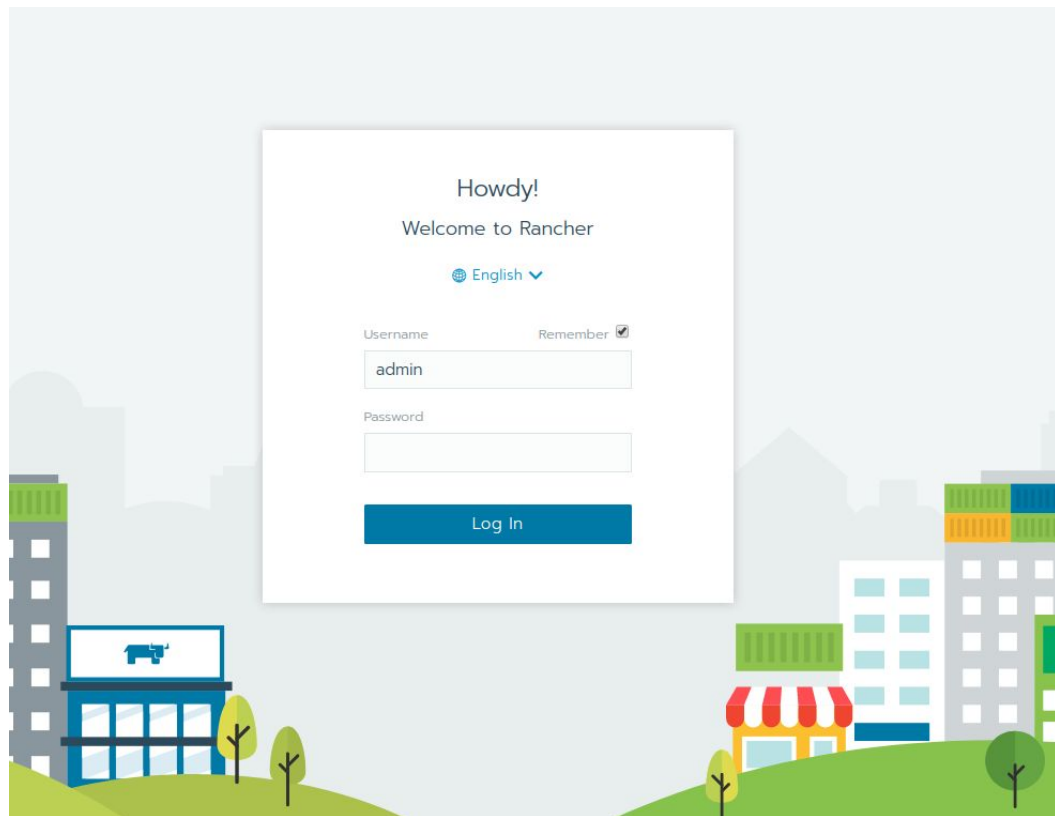
- 1 server for Rancher
 - Docker
 - Port requirements
 - X nr of servers for your Kubernetes cluster.
 - Docker
 - Port requirements
-

Rancher

```
docker run -d --restart=unless-stopped -p 80:80 -p 443:443 rancher/rancher:latest
```



Rancher



Rancher



Add Cluster - Select Cluster Type



From existing nodes (Custom)

Create a new Kubernetes cluster using RKE, out of existing bare-metal servers or virtual machines.



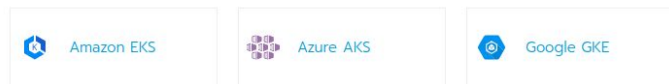
Import an existing cluster

Import an existing Kubernetes cluster. The provider that created it will continue to manage the provisioning and configuration of the cluster.

With RKE and new nodes in an infrastructure provider:



With a hosted Kubernetes provider:



Cancel

Rancher



Add Cluster - Custom

Cluster Name * [Add a Description](#)

mycluster

Member Roles

Control who has access to the cluster and what permission they have to change it.

Labels & Annotations

Configure labels and annotations for the cluster.

None

Cluster Options

[Edit as YAML](#) 

[Expand All](#)

Kubernetes Options

Customize the kubernetes cluster options.

Kubernetes Version

v1.16.3-rancher1-1

Network Provider

Canal (Network Isolation Available)

Windows Support

☐ Enabled

☒ Disabled

Project Network Isolation

☐ Enabled

☒ Disabled

Available for Kubernetes 1.15 with Flannel network provider.

Rancher



Global ▾ Clusters Apps Users Settings Security ▾ Tools ▾



Add Cluster - Custom

Cluster Options

Edit as YAML

Customize Node Run Command

Editing node options will update the command you will run on your existing machines

1

Node Options

Choose what roles the node will have in the cluster

Node Role

☐ etcd

☒ Control Plane

☐ Worker

[Show advanced options](#)

2

Run this command on one or more existing machines already running a supported version of Docker.

```
sudo docker run -d --privileged --restart=unless-stopped --net=host -v /etc/kubernetes:/etc/kubernetes -v /var/run:/var/run rancher/rancher-agent:v2.3.3 --server https://0.0.0.0 --token bnc5kgwp8wjz5vbqzcgz9v2x74jkgd4t7fdgrltpk85jntllsm5bk --ca-checksum 677c19cf56e8ca6f3bac97390e4cb54cd19ff172bc848fe86198b1bce09e32b5 --controlplane
```



Done

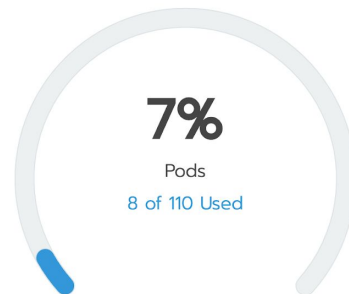
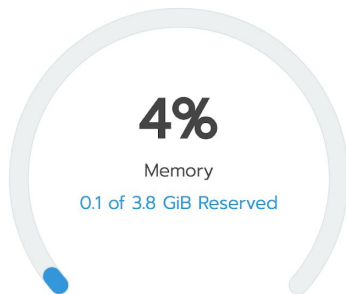
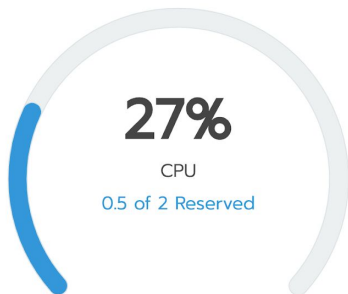
Rancher

Dashboard: custom

≥ [Launch kubectl](#) [Kubeconfig File](#) ⋮

Provider: Custom	Version: v1.13.5	Nodes: 3
CPU: 2 Cores	Memory: 3.8 GiB	Created: 3:42 PM

[Enable Monitoring to see live metrics](#)



✓ Etcd

✓ Controller Manager

✓ Scheduler

✓ Nodes



K3S



K3S

- Slimmed down Kubernetes distribution.
- Requires only 512mb of memory
- Cloud components removed
- Support for ARM
- ...can even run on a Raspberry PI!

Lightweight Kubernetes

The certified Kubernetes distribution built for IoT & Edge

Great For

Edge

IoT

CI

ARM

K3S

Requirements

- Docker
- Port dependencies
- At least 1 server for your Kubernetes cluster.

K3S - Install Kubernetes master / 1 node cluster

```
curl -sfL https://get.k3s.io | sh -
```



K3S - Install Kubernetes worker

```
curl -sfL https://get.k3s.io | K3S_URL=https://myserver:6443  
K3S_TOKEN=mynodetoken sh -
```





KUBESPRAY

Kubespray

- Ansible playbook
- Support for both cloud and non cloud environments.
- Addons
 - Storage (Ceph, Local path)
 - Ingress (Nginx, Traefik)
 - Cert manager
- Support for multiple clusters
- Configuration files in in YAML.
 - Docker
 - Kubernetes
 - Network plugins
 - Cloud integration

```
1 all:
2   vars:
3     kubelet_cgroup_driver: cgroupfs
4   hosts:
5     localhost:
6       ansible_python_interpreter: /usr/bin/python
7     node1:
8       ansible_host: [REDACTED]
9       ip: [REDACTED]
10      access_ip: [REDACTED]
11     node2:
12       ansible_host: [REDACTED]
13       ip: [REDACTED]
14       access_ip: [REDACTED]
15     node3:
16       ansible_host: [REDACTED]
17       ip: [REDACTED]
18       access_ip: [REDACTED]
19     node4:
20       ansible_host: [REDACTED]
21       ip: [REDACTED]
22       access_ip: [REDACTED]
23   children:
24     kube-master:
25       hosts:
26         node1:
27     kube-node:
28       hosts:
29         node1:
30         node2:
31         node3:
32         node4:
33     etcd:
34       hosts:
35         node1:
```


Kubespary

Requirements

- Python
- X nr of servers for your Kubernetes cluster.
 - Docker
 - Port requirements

Install Kubernetes with Kubespray

Inventory file

```
1 all:
2   vars:
3     kubelet_cgroup_driver: cgroupfs
4   hosts:
5     localhost:
6       ansible_python_interpreter: /usr/bin/python
7   node1:
8     ansible_host: [REDACTED]
9     ip: [REDACTED]
10    access_ip: [REDACTED]
11   node2:
12     ansible_host: [REDACTED]
13     ip: [REDACTED]
14     access_ip: [REDACTED]
15   node3:
16     ansible_host: [REDACTED]
17     ip: [REDACTED]
18     access_ip: [REDACTED]
19   node4:
20     ansible_host: [REDACTED]
21     ip: [REDACTED]
22     access_ip: [REDACTED]
23   children:
24     kube-master:
25       hosts:
26         node1:
27     kube-node:
28       hosts:
29         node1:
30         node2:
31         node3:
32         node4:
33   etcd:
34     hosts:
35     node1:
```

Install Kubernetes with Kubespray

Configuration files

```
1  ---
2  # Kubernetes configuration dirs and system namespace.
3  # Those are where all the additional config stuff goes
4  # the kubernetes normally puts in /srv/kubernetes.
5  # This puts them in a sane location and namespace.
6  # Editing those values will almost surely break something.
7  kube_config_dir: /etc/kubernetes
8  kube_script_dir: "{{ bin_dir }}/kubernetes-scripts"
9  kube_manifest_dir: "{{ kube_config_dir }}/manifests"
10
11 # This is where all the cert scripts and certs will be located
12 kube_cert_dir: "{{ kube_config_dir }}/ssl"
13
14 # This is where all of the bearer tokens will be stored
15 kube_token_dir: "{{ kube_config_dir }}/tokens"
16
17 # This is where to save basic auth file
18 kube_users_dir: "{{ kube_config_dir }}/users"
19
20 kube_api_anonymous_auth: true
21
22 ## Change this to use another Kubernetes version, e.g. a current beta release
23 kube_version: v1.15.3
24
25 # kubernetes image repo define
26 kube_image_repo: "gcr.io/google-containers"
27
28 # Where the binaries will be downloaded.
29 # Note: ensure that you've enough disk space (about 1G)
30 local_release_dir: "/tmp/releases"
31 # Random shifts for retrying failed ops like pushing/downloading
32 retry_stagger: 5
33
34 # This is the group that the cert creation scripts chgrp the
35 # cert files to. Not really changeable...
36 kube_cert_group: kube-cert
37
38 # Cluster LogLevel configuration
39 kube_log_level: 2
40
41 # Directories to store logs, all logs will be stored in this directory
```

Install Kubernetes with Kubespray

```
ansible-playbook -i inventory/prod/hosts.yml cluster.yml -b -v --user=pixelpiloten  
--private-key=~/.ssh/id_rsa --ask-sudo-pass
```



Rancher

Pros / Cons

Pros

- Ease of use
- Support for both cloud and non cloud environments.
- Manage multiple clusters.
- Monitoring, Logging and Exec into pods.

Cons

- Limited configuration options.
 - Upgrading Kubernetes?
-

K3S

Pros / Cons

Pros

- Ease of use.
- Low memory.
- Possible to have a 1 node cluster.
- Support for a HA cluster. (since v1.0)

Cons

- No cloud integration.
- Not many configuration options.

Kubespary

Pros / Cons

Pros

- Support for both Cloud and non Cloud environments.
- A lot of configuration options.
- Addons
- Upgradeability

Cons

- Requires some manual steps
 - A lot of configuration options.
-

What, when and where?

- **Rancher**
 - If you have multiple clusters.
 - More cloud centric.
 - **K3S**
 - If you have limited resources.
 - Great for IOT and CI/CD Runners.
 - **Kubespary**
 - Great for “On prem”.
 - If you need to have a granular control over your Kubernetes/Docker settings.
-

Thank you!

DevOps guide:

[www.redpill-linpro.com/resources/
devops-guide](http://www.redpill-linpro.com/resources/devops-guide)

IT Talks Podcast:

[Ittalks.libsyn.com](http://ittalks.libsyn.com) (February the 14)

Blog: www.pixelpiloten.se

LinkedIn: pixelpiloten
