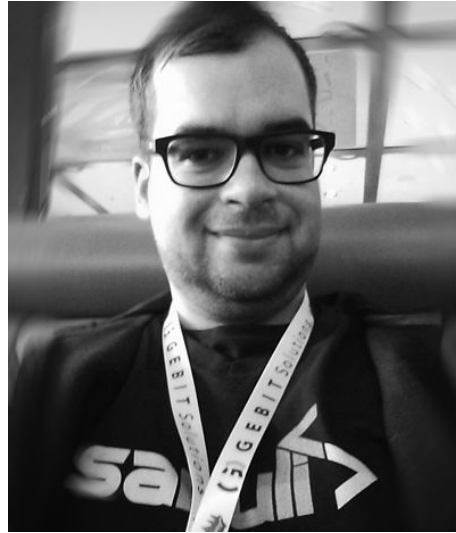




KubeOne

Kubernetes Cluster Lifecycle Management Tool

Who am I?



Tobias Schneck

Software Engineer @ Loodse



@toschneck

Who are the Developer?



Marko Mudrinić
Software Developer @ Loodse

  @xmudrii



Artiom Diomin
Software Developer @ Loodse

  @kron4eg

Operational Excellence for Your Cloud Native Applications



Our Expertise

Loodse is a leading expert for container and cloud native technologies.



Our Team

We are a team of 40+ employees across Europe and the US.



Locations

Founded in 2016, our headquarters are located in Hamburg, Germany.

Agenda



- ▶ Introduction to KubeOne
- ▶ Core concepts and architecture
- ▶ Demo: Manage Kubernetes HA cluster on AWS
- ▶ Demo: Configure the cluster and explore KubeOne features



Introduction



What is KubeOne?

- ▶ A tool for **managing** Kubernetes **cluster lifecycle**
 - ▶ Installs and provisions Kubernetes, upgrades, un-provisions the cluster
- ▶ **Open source** and **vendor neutral**
- ▶ Works on the most popular **cloud providers**, on **on-prem** and on **bare metal**
- ▶ Supports **1.13+ Highly-Available** clusters



Why we built KubeOne?

- ▶ Kubernetes brought us a new way for managing our workload...
- ▶ but managing Kubernetes clusters is still a hard task.
- ▶ We want to apply lessons learned managing workload to clusters.

In a search for a **feature-complete** solution, we decided to build **KubeOne**



Why KubeOne?

- ▶ Uses the **latest technologies** to bring **many features** in an **easy to consume** manner
- ▶ Brings **declarative** cluster representation
- ▶ Provides **ready to use** cluster
- ▶ Optionally configures various features on the provisioning time:
 - ▶ PodSecurityPolicy, DynamicAuditLog, metrics-server and more
- ▶ Ability to integrate KubeOne with **infrastructure provisioning tools**



Supported providers

- ▶ KubeOne is supposed to work on **any** provider, including on-prem and bare metal
- ▶ Officially supported providers enjoy **additional features** such as:
 - ▶ Support for managing worker nodes using Kubermatic machine-controller
 - ▶ Automatically deploy cloud provider specific features like external CCM
 - ▶ Use Terraform integration to pick up information about infrastructure from the Terraform state
- ▶ Officially supported providers include AWS, GCE, DigitalOcean, Hetzner, Packet, OpenStack and VMware vSphere
- ▶ Microsoft Azure will be supported as of the upcoming v0.9 release



Create cluster on AWS



Create cluster on AWS

- ▶ Step 1: Create instances and infrastructure to be used by Kubernetes
 - ▶ KubeOne comes with example Terraform scripts that can be used to get started
- ▶ Step 2: Build KubeOne configuration manifest
 - ▶ Defines what Kubernetes version will be installed, what machines will be used, how the cluster will be provisioned...
- ▶ Step 3: Run `kubeone install` command
- ▶ Step 4: Enjoy!

Building KubeOne Cluster manifest



```
apiVersion: kubeone.io/v1alpha1
kind: KubeOneCluster
versions:
  kubernetes: 1.14.2
cloudProvider:
  name: aws
```

Building KubeOne Cluster manifest



```
apiVersion: kubeone.io/v1alpha1
```

```
kind: KubeOneCluster
```

```
versions:
```

```
  kubernetes: 1.14.2
```

```
cloudProvider:
```

```
  name: aws
```

Building KubeOne Cluster manifest



```
apiVersion: kubeone.io/v1alpha1
```

```
kind: KubeOneCluster
```

```
versions:
```

```
  kubernetes: 1.14.2
```

```
cloudProvider:
```

```
  name: aws
```

Building KubeOne Cluster manifest



```
apiVersion: kubeone.io/v1alpha1
```

```
kind: KubeOneCluster
```

```
versions:
```

```
  kubernetes: 1.14.2
```

```
cloudProvider:
```

```
  name: aws
```




Demo time!

Architecture

Architecture



- ▶ KubeOne uses many tools/solutions as **building blocks**
 - ▶ kubeadm is used to provision and join **control plane nodes** and handle **cluster upgrades**
 - ▶ Kubermatic machine-controller based on Cluster-API is used to manage **worker nodes**
- ▶ The environment is prepared over SSH
 - ▶ Including installing and upgrading binaries, configuring components and running kubeadm
- ▶ client-go is used for deploying various cluster features such as CNI



Installation process

Installation process

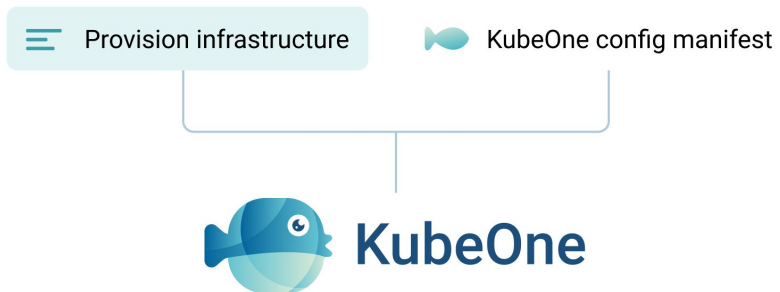


☰ Provision infrastructure

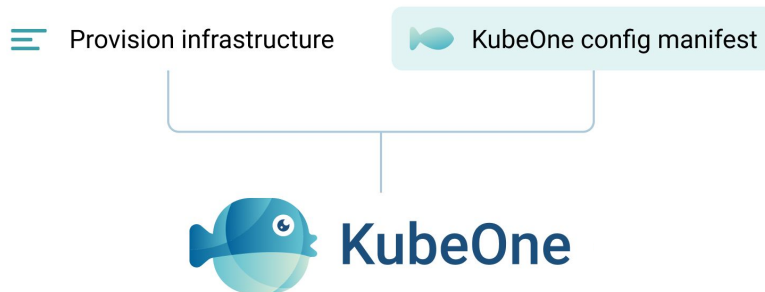
🐟 KubeOne config manifest



Installation process



Installation process



Installation process



☰ Provision infrastructure 🐟 KubeOne config manifest



KubeOne



Provision cluster



Download KubeConfig



Configure and deploy



Create worker nodes

Installation process



☰ Provision infrastructure 🐟 KubeOne config manifest



KubeOne



Provision cluster



Download KubeConfig



Configure and deploy



Create worker nodes

Installation process



☰ Provision infrastructure 🐟 KubeOne config manifest



KubeOne



Provision cluster



Download KubeConfig

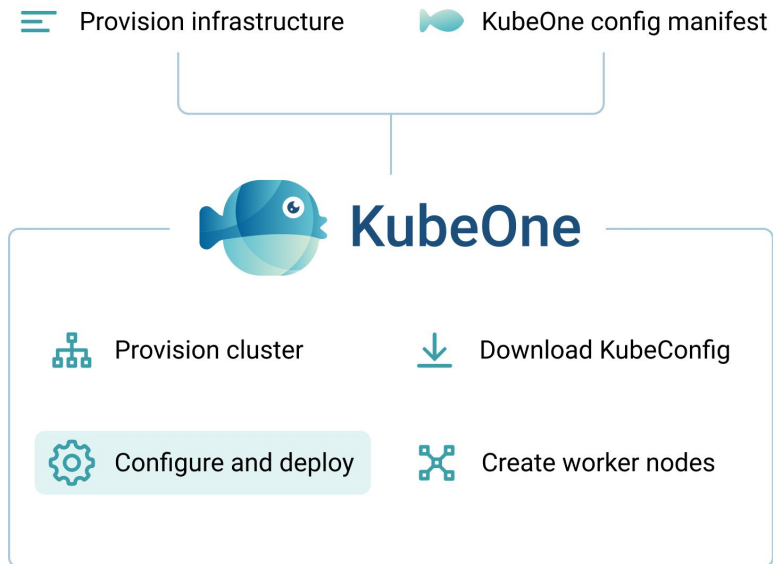


Configure and deploy



Create worker nodes

Installation process



Installation process



☰ Provision infrastructure 🐟 KubeOne config manifest



KubeOne



Provision cluster



Download KubeConfig



Configure and deploy



Create worker nodes

Installation process



☰ Provision infrastructure 🐟 KubeOne config manifest



KubeOne



Provision cluster



Download KubeConfig



Configure and deploy



Create worker nodes





Managing worker nodes



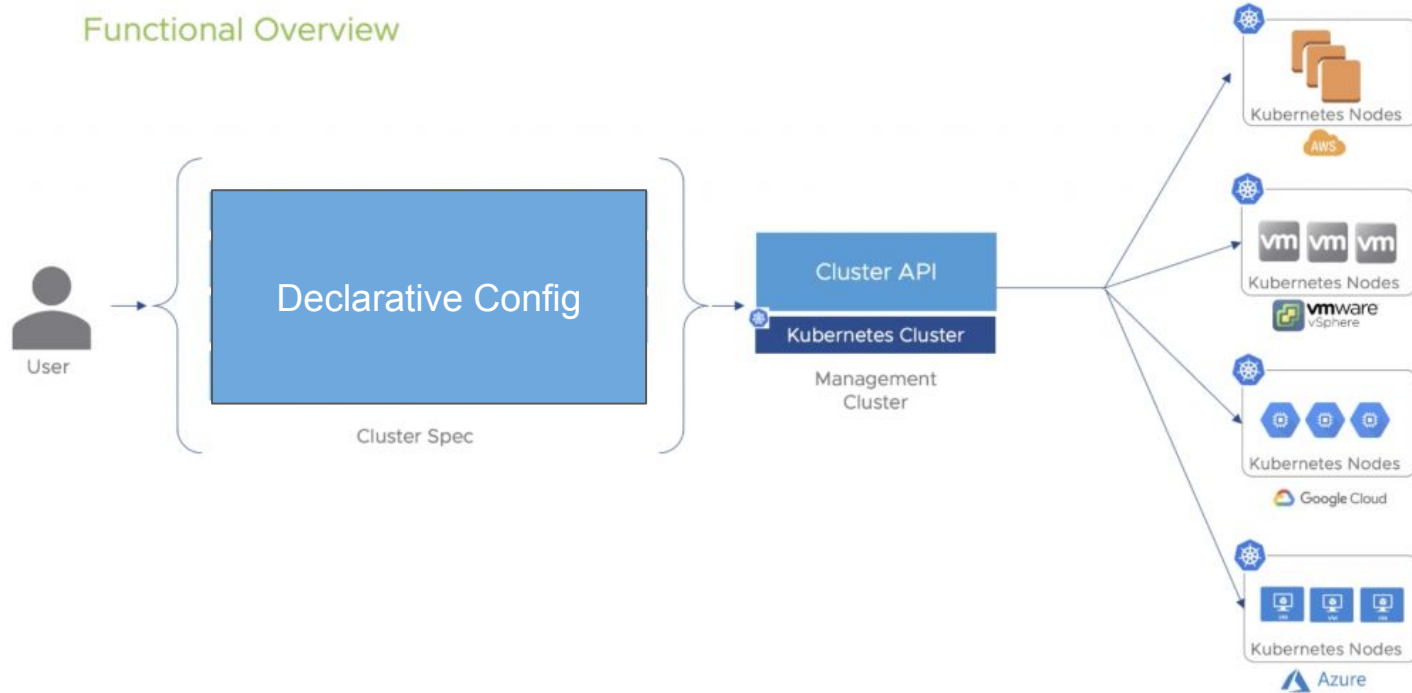
Managing worker nodes

- ▶ Worker nodes are managed using **Kubermatic machine-controller**
- ▶ **machine-controller** is an open source **Cluster-API** implementation
- ▶ Cluster-API is a **declarative, Kubernetes-style** API for cluster and machine **creation, configuration** and **management**
- ▶ You define what you want, controller creates it for you
 - ▶ In KubeOne case, machine-controller provisions and configures machines

What is Cluster API?



Cluster API Functional Overview





Managing worker nodes

- ▶ In KubeOne config manifest, user provides number of **replicas** and **cloud provider specification**
- ▶ Based on provided information, KubeOne creates **MachineDeployment** object
- ▶ machine-controller creates **MachineSet** and **Machine** objects, which trigger creation and provisioning of cloud instances
- ▶ machine-controller watches machines all the time
 - ▶ If machine/node becomes unavailable/unhealthy machine will be recreated
 - ▶ If machine is changed (e.g. upgraded), all machines in the MachineDeployment will be rolled-out

Managing worker nodes



MachineDeployment



MachineSet



Machine



Deployment



ReplicaSet



Pod





Upgrade process



Demo time!




Upgrade process

- ▶ The control plane nodes are upgraded **in-place**
- ▶ Upgrading control plane nodes include upgrading:
 - ▶ Kubernetes binaries
 - ▶ core Kubernetes components
 - ▶ all components deployed by KubeOne
- ▶ Worker nodes are upgraded by **rolling out** MachineDeployment

Upgrade process




 KubeOne config manifest



Upgrade process




 KubeOne config manifest



Upgrade process



 KubeOne config manifest




KubeOne

✓ Preflight checks


↑ Upgrade Cluster

✓ Check components

 Roll out worker nodes

Upgrade process



 KubeOne config manifest




KubeOne

✓ Preflight checks


↑ Upgrade Cluster

✓ Check components

 Roll out worker nodes

Upgrade process




 KubeOne config manifest




KubeOne

✓ Preflight checks


 Upgrade Cluster

✓ Check components

 Roll out worker nodes

Upgrade process



 KubeOne config manifest




KubeOne

✓ Preflight checks


↑ Upgrade Cluster

✓ Check components

 Roll out worker nodes

Upgrade process



 KubeOne config manifest




KubeOne

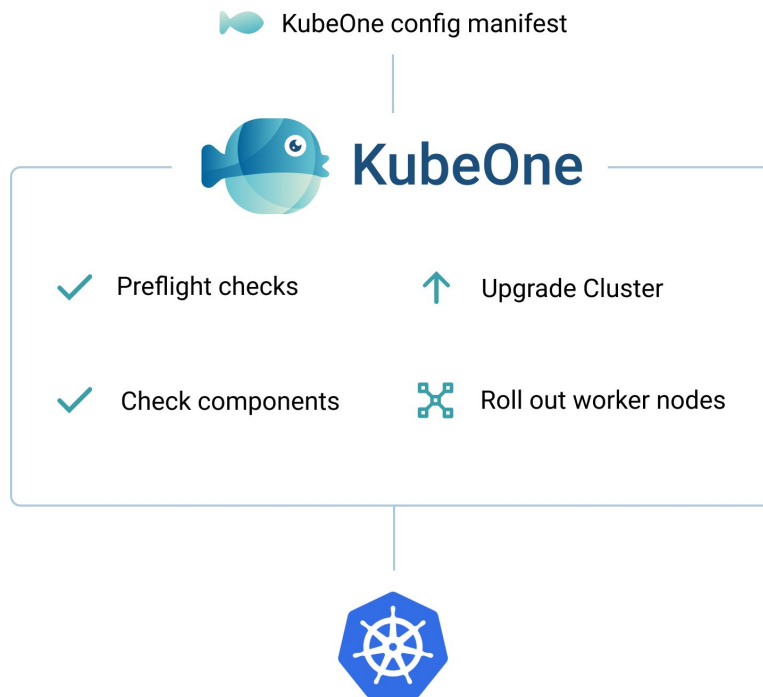
✓ Preflight checks

↑ Upgrade Cluster

✓ Check components

 Roll out worker nodes

Upgrade process





Questions? Remarks?

Thank you for your time!



- ▶ Find KubeOne on GitHub: <https://github.com/kubermatic/kubeone>
- ▶ Follow us on Twitter: [@Loodse](#), [@xmudrii](#), [@kron4eg](#), [@toschneck](#)
- ▶ Check out Loodse blog: <https://loodse.com/blog>
- ▶ Join `#kubeone` on Kubermatic Slack: <http://slack.kubermatic.io>