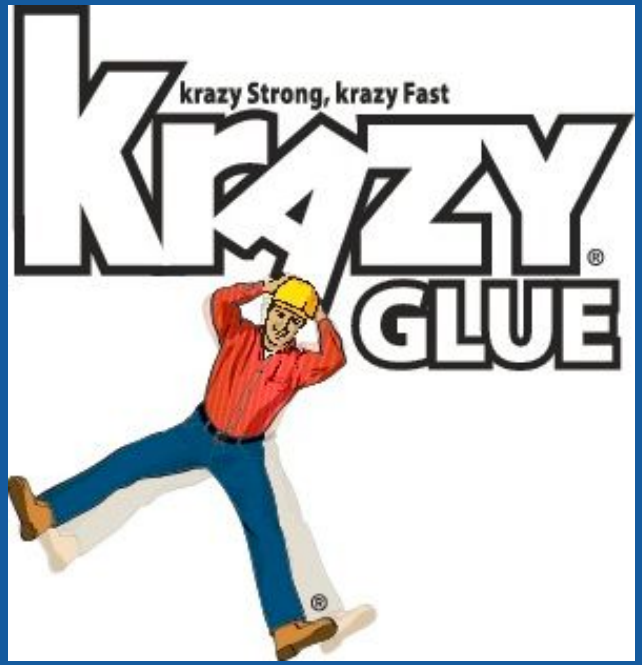
Zero-Configuration Pattern

Provisioning Kubernetes on Unmanaged Infrastructure

Rob @zehicle Hirschfeld, RackN

November, 2017

Hang on to your Hats!



Krazy New Stuff

- Immutable Bootstrap (demo!)
- Node Admission (v1.7)
- Dynamic Kubelet (v1.8)

RackN

Rob Hirschfeld (@zehicle)

Involved in Kubernetes since launch

Co-chair of Cluster Ops SIG

Co-Founder of RackN & Digital Rebar Project

We focus on operations automation for bare metal

RackN

But first... Kubespray

We've been using Kubespray since Kubernetes v1.2

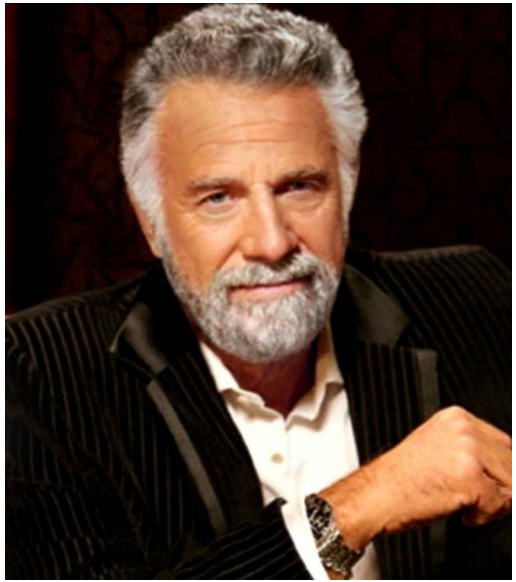
- **Very Solid Ansible Playbook**
- **Strong Community**
- **Amazing Features like HA & Upgrade**

HTTP://bit.ly/SYDkubespray

But....

Why not Kubespray?

I don’t always Ansible, but when I do Ansible, I use Kubespray.



We’d like to do better!

- No Centralized Orchestration
- No Inventory Building
- No SSH
- Immutable Booting
- and, much FASTER

RackN

Let's get Immutable!

What?

- Create, Destroy & Repeat
- Machines recreated, not updated
- Typically "Pre-Baked" images

Why?

- Very repeatable and predictable installation
- Simpler node configuration
- Faster deploy time

RackN

Leveraging Kubeadm

Community converging to single install utility!

Basic Three Step Cluster Initialization:

1. **Initialize Master**
2. **Retrieve Token from Initialize**
3. **Join Nodes with Token**

Still requires coordination / synchronization

RackN

But First, Kubeadm Prereqs

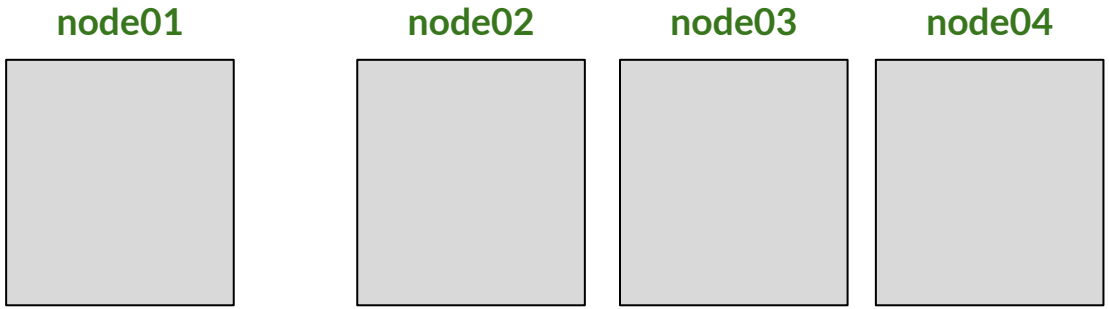
We need to build underlay infrastructure

Basic Three Step Underlay:

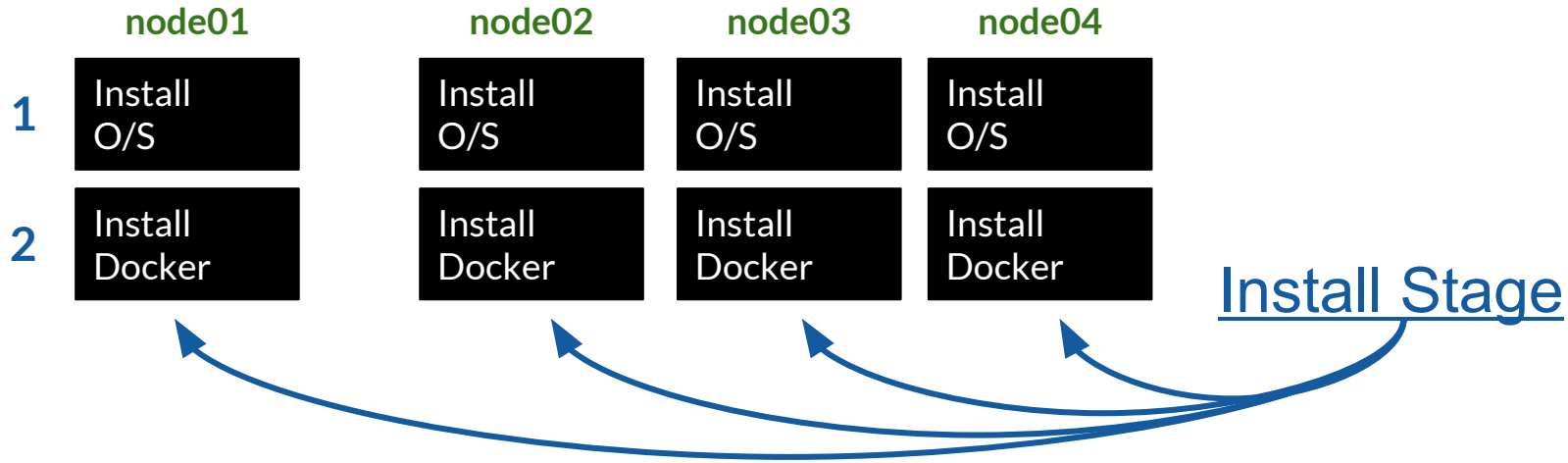
1. **install operating system *with network access***
2. **attach disks (optional?!)**
3. **install Docker on the machine**

Oh, and we need to have some control mechanism on the nodes too.

A bootstrapping illustration



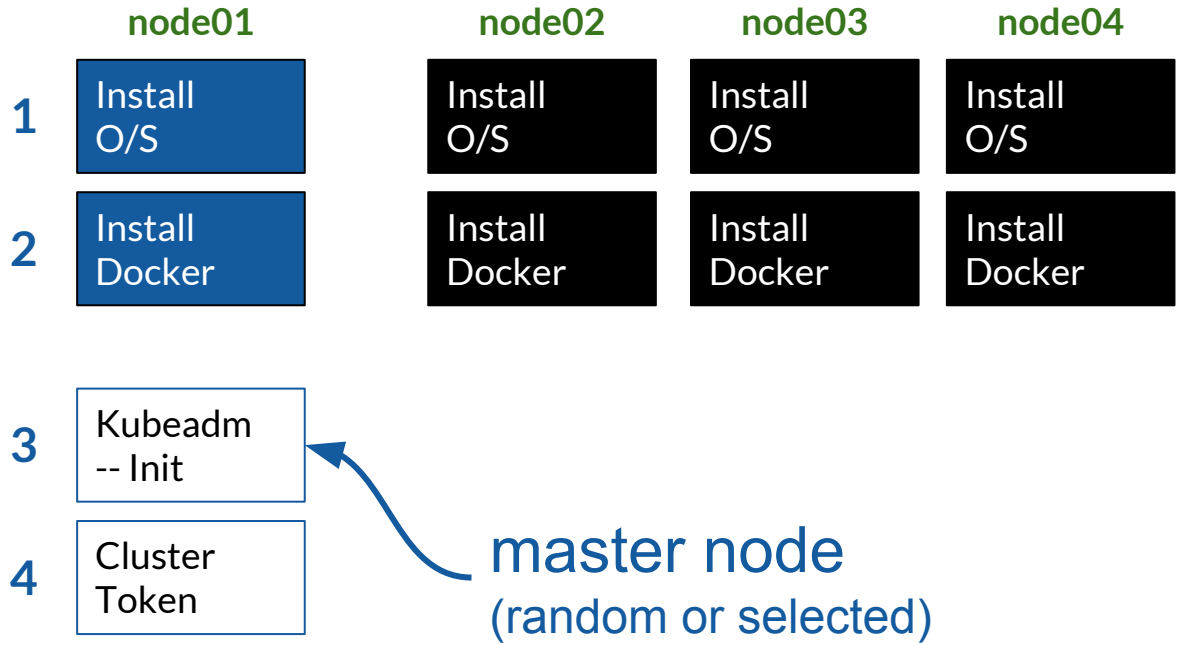
A bootstrapping illustration



RackN

#KubeCon - @zehicle

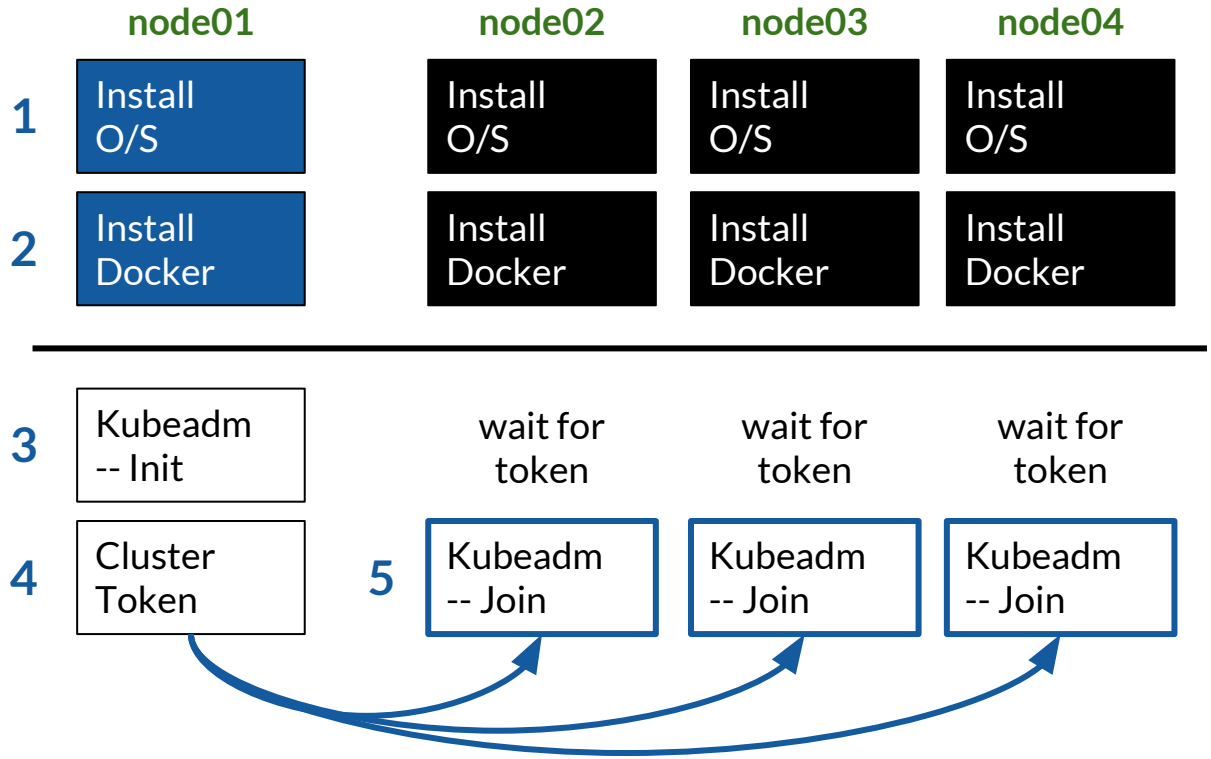
A bootstrapping illustration



RackN

#KubeCon - @zehicle

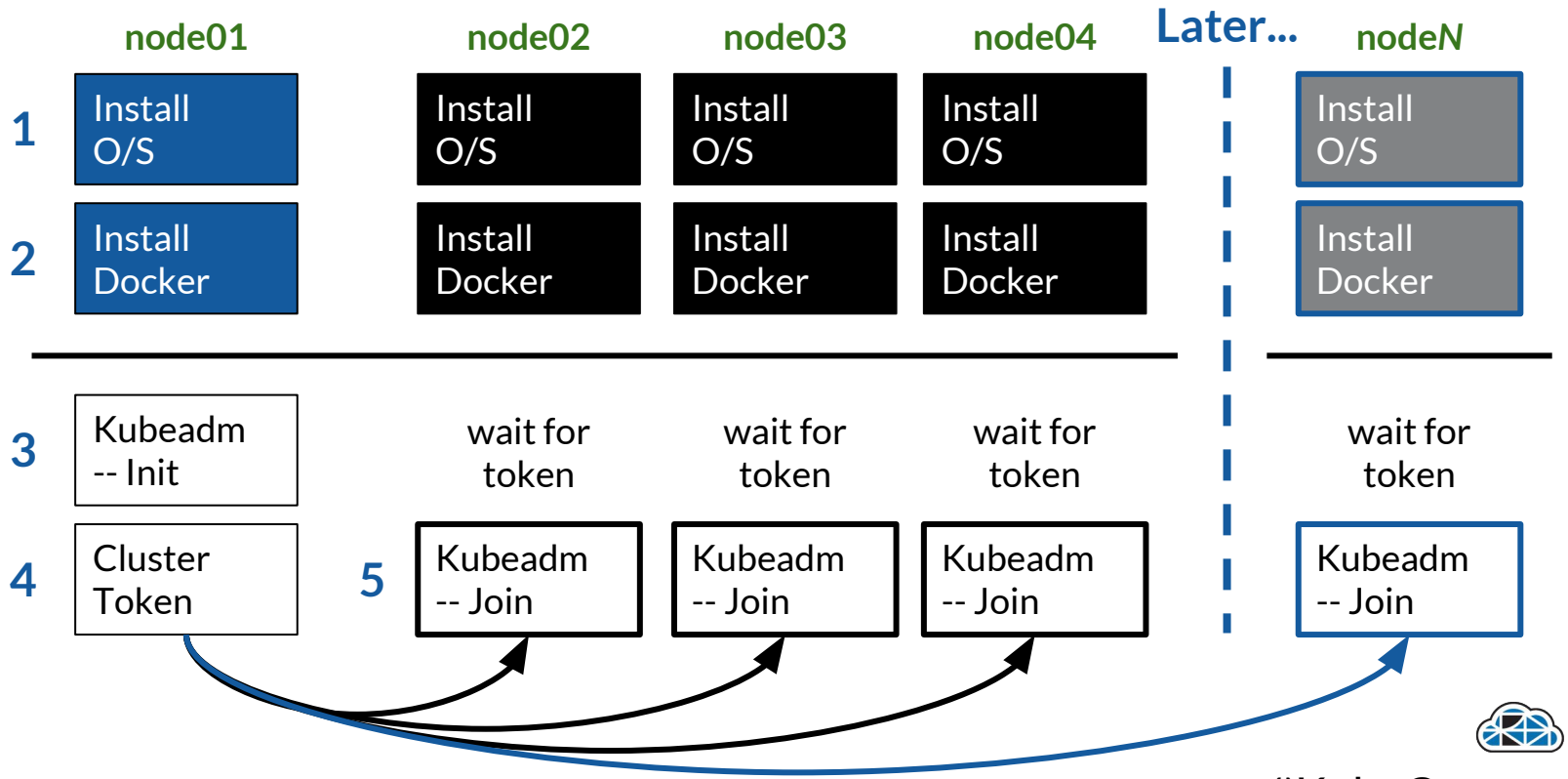
A bootstrapping illustration



RackN

#KubeCon - @zehicle

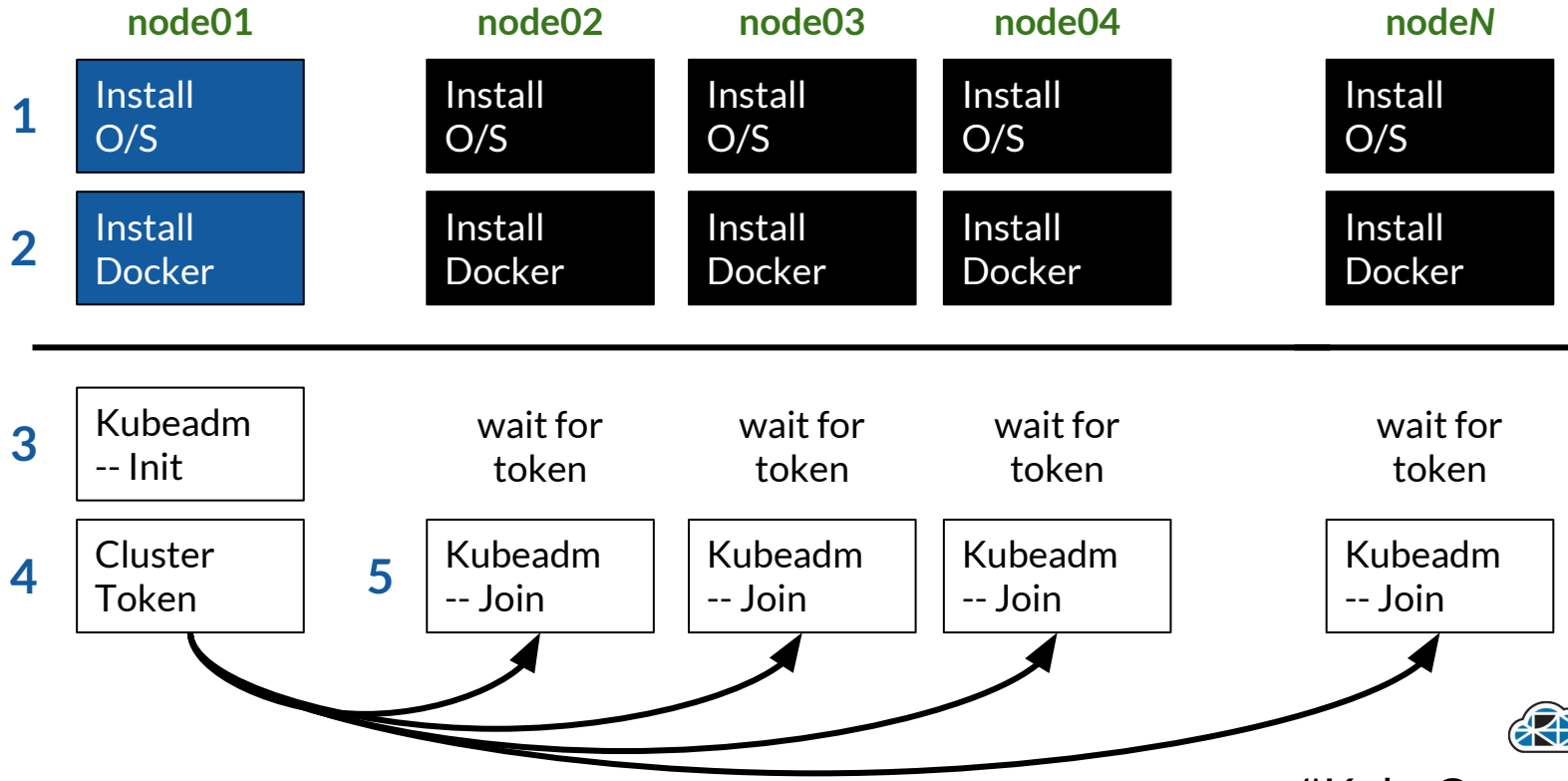
A bootstrapping illustration



RackN

#KubeCon - @zehicle

A bootstrapping illustration



RackN

#KubeCon - @zehicle

PSA: *THIS IS NOT A NEW INSTALLER*

At RackN, we push back against the distro installer wars *(ala OpenStack)*.

We believe that Kubernetes install tooling should be a shared community investment.

RackN

Demo!

Kubeadm
Rebar
Immutable
Bootstrap

RackN

Pretty Cool! But...

There is more to do

- Adding Nodes requires Token
- Adding Kubelet requires Configuration
- Cluster API (Orchestrating Update)

RackN

Node Admission

https://kubernetes.io/docs/admin/admission-controllers/

Benefits!

1. Immutable Configuration
2. Auto Scaling
3. Faster Node Install
4. Centralized Configuration of Cluster
5. Coordinated Upgrades

Still requires coordination / synchronization

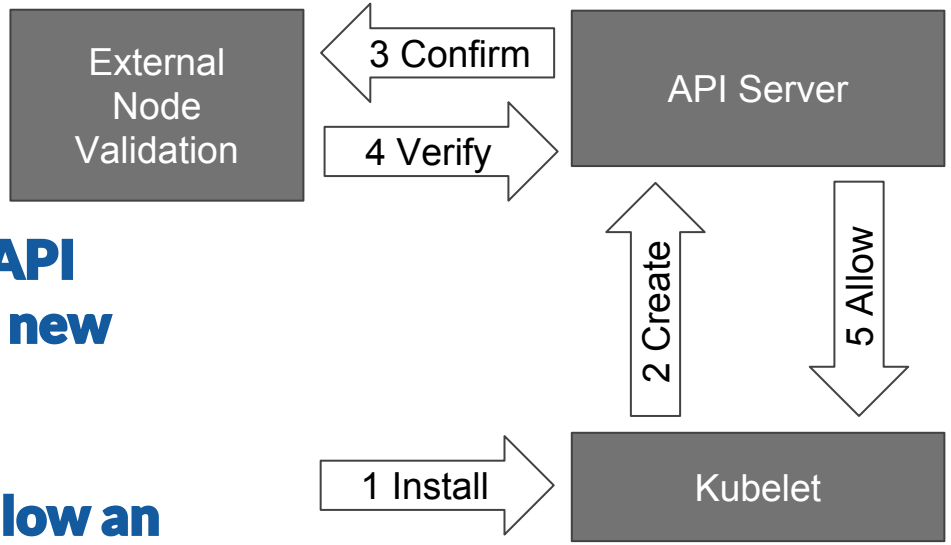
Node Admission

HSM: Hardware Signing Module

NOT Node specific!

Admission control provides an API mechanism to block creation of new objects.

In this case, Admission would allow an external system to validate that new nodes are known and trusted.



RackN

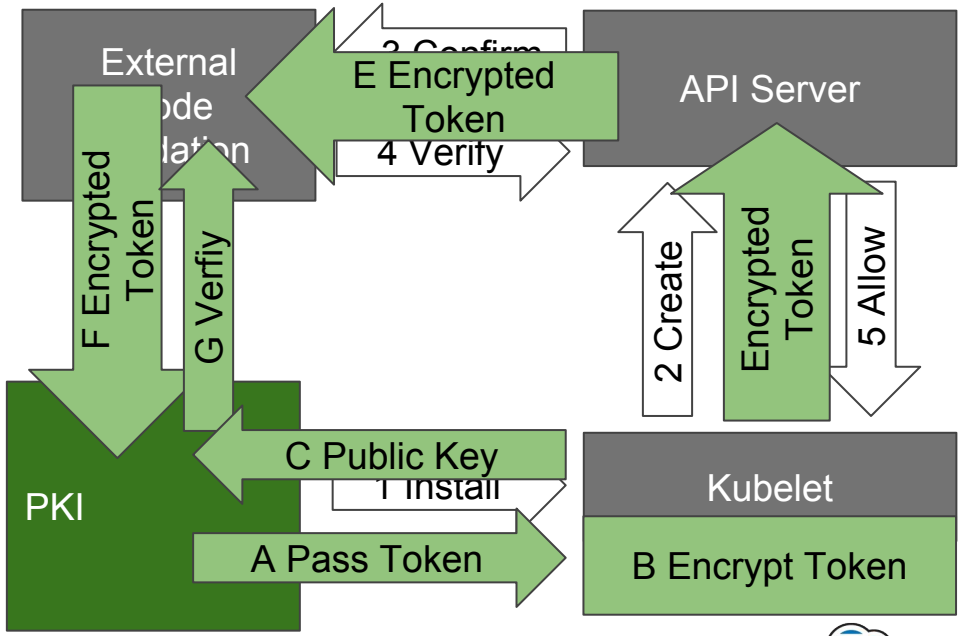
#KubeCon - @zehicle

Node Admission with HSM

HSM: Hardware Signing Module

HSM ensures unique identy of machine by signing secret token.

Only token creater (PKI) and machine know the secret. API Server cannot read or validate internally.



RackN

#KubeCon - @zehicle

Is Node Admission Needed?

Frankly, RackN is on the fence.

If injecting a join cluster token then the external system has already verified the new node.

RackN

Kubelet Dynamic Configuration

https://kubernetes.io/docs/tasks/administer-cluster/reconfigure-kubelet/

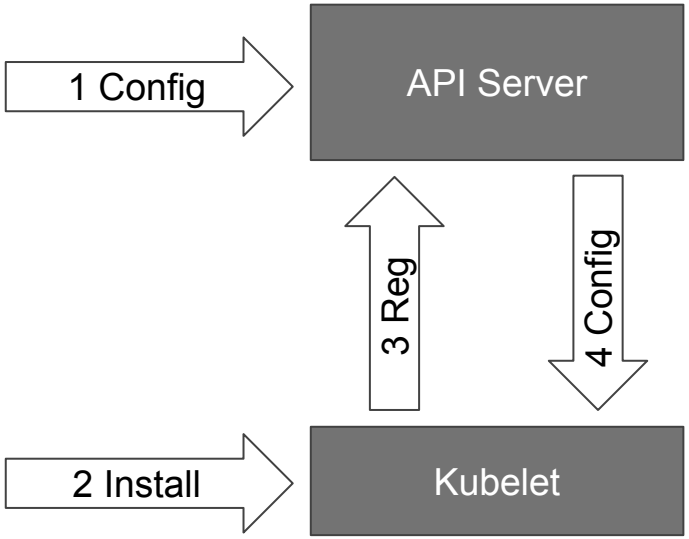
We want to eliminate external configuration tools.

Kubernetes is already a system configuration database!
Can't we just use that capability to bootstrap the system?

Then we have fewer tools to learn and managed!
(IMHO, this is known as a the bootstrap fallacy)

Ideally, it would be like this...

1. **Centrally Configurate**
2. **Install Kubelet**
3. **Allow Kubelet to Register**
4. **Kubelet Configures itself**



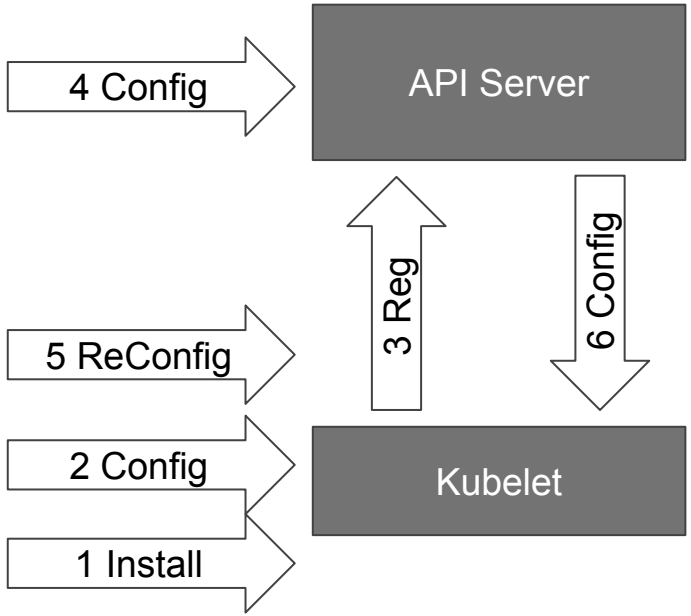
RackN

#KubeCon - @zehicle

Kubelet Dynamic Configuration

https://kubernetes.io/docs/tasks/administer-cluster/reconfigure-kubelet/

1. **Install Node and Kubelet**
2. **Configure Kubelet**
3. **Allow Kubelet to Register**
4. **Register Configuration in API**
5. **Reconfigure Kubelet to use configuration from API**
6. **Manage configuration from API**



RackN

#KubeCon - @zehicle

Is Dynamic Configuration Needed?

Frankly, RackN is on the fence.

Since we have to boostrap a node with *some* configuration, there is not much difference between some and all configuration.

We have not eliminated configuration.

RackN

We're Making Great Progress!

We can automatically bootstrap a cluster using open community tools with minimal configuration.

And we have room to improve.

RackN

Thank you

Join In! http://rebar.digital

Follow:

- Rob Hirschfeld > @zehicle
- RackN > @rackngo
- Digital Rebar > @digitalrebar
- Cluster Ops SIG > http://bit.ly/k8sclops

RackN