

# Deploying I&D solutions in Kubernetes

Nick Glowacki



IBM Cloud

IBM

## Legal Disclaimer

(Sections / text in red need to be customized for each presentation and then removed from the final draft.)

© IBM Corporation 2018. All Rights Reserved.

The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

If the text contains performance statistics or references to benchmarks, insert the following language; otherwise delete:

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

If the text includes any customer examples, please confirm we have prior written approval from such customer and insert the following language; otherwise delete:

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Review text for proper trademark attribution of IBM products. At first use, each product name must be the full name and include appropriate trademark symbols. Subsequent references can drop "IBM" but should include the proper branding (e.g., IBM Bluemix or WebSphere Application Server). Please refer to <http://www.ibm.com/legal/copytrade.shtml> for guidance on which trademarks require the ® or ™ symbol. Do not use abbreviations for IBM product names in your presentation. All product names must be used as adjectives rather than nouns. Please list all of the trademarks that you use in your presentation

If the text/graphics include screenshots, no actual IBM employee names may be used (even your own), if your screenshots include fictitious company names (e.g., Renovations, Zeta Bank, Acme) please update and insert the following; otherwise delete: All references to [insert fictitious company name] refer to a fictitious company and are used for illustration purposes only.

# Contents

[Why](#)

[\*\*Where did Kubernetes come from?\*\*](#)

[Core Concepts](#)

[Things to consider and Challenges](#)

# Why?



Why?

**Wait a minute?  
Was this not one of the purposes of WAS?**

# Benefits of containerization

Build Agility

Team Productivity

Fine-grained Resilience

Scalability and Infrastructure Optimization

Operational Consistency

Component Portability

**Containerization is more than just a re-platforming exercise.** “lift and shift” will not bring the above benefits.  
Requires: *Fine-grained deployment, organizational decentralization, pipeline automation, disposable components...*

# Agile Integration Architecture

## Fine grained deployment

*Improve build independence and production velocity*

Dependency free rapid integration delivery

## Decentralized Ownership

Accelerate agility and innovation

## Cloud native infrastructure

*Dynamic scalability and inherent resilience*

Consumer centric exposure of business APIs

Independent application centric messaging

Business autonomy for integration delivery

Self-administration of API exposure and subscription

Self-provisioning of messaging and event capabilities

Scale and administer integrations with applications

Multi-platform cloud agnostic API management componentry

Cloud scale inherently resilient multi-platform messaging

# Where did Kubernetes come from?



# kubernetes

## Kubernetes

The Kubernetes project was started by Google in 2014.

Open source system for managing containerized applications.



December 17, 2016 – January 17, 2017

---

Excluding merges, **136 authors** have pushed **577 commits** to master and **656 commits** to all branches. On master, **6,096 files** have changed and there have been **343,654 additions** and **301,994 deletions**.

Integral part of the CNCF

# Cloud Native Landscape

v20180525

See the interactive landscape at [l.cncf.io](https://l.cncf.io)

Greyed logos are not open source

### App Definition and Development

### Platforms

### Observability & Analysis

### Scheduling & Orchestration

### Coordination & Service Discovery

### Service Management

### Runtime

### Host Management / Tooling

### Kubernetes Training Partner

### Cloud

### Special

This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well-traveled path.

[l.cncf.io](https://l.cncf.io)

CLOUD NATIVE  
Landscape

CLOUD NATIVE  
COMPUTING FOUNDATION

Redpoint Amplify

# What's Kubernetes

- Open source by Google
- One sized fits all Container orchestration and cluster management
- Self-healing and Failover
- Horizontal and Vertical Scaling
- Allows custom “plugins” to become part of the internal components
  - Does not ship
    - Package Management (Helm)
    - Networking (Calico/Flannel)
    - Storage (CEPH/Gluster)

# How does IBM Cloud Private fit in?

- Takes Kubernetes and a number of “plugins” and makes them fit for enterprise
  - Provides Support
  - Provides Proven patterns
  - Does not stop other 3<sup>rd</sup> party plugins being used.
  - Includes
    - Helm
    - Calico
    - Gluster FS
    - And Many Others

# Key concepts of **Kubernetes**

ACE  
Container

# Key concepts of **Kubernetes**

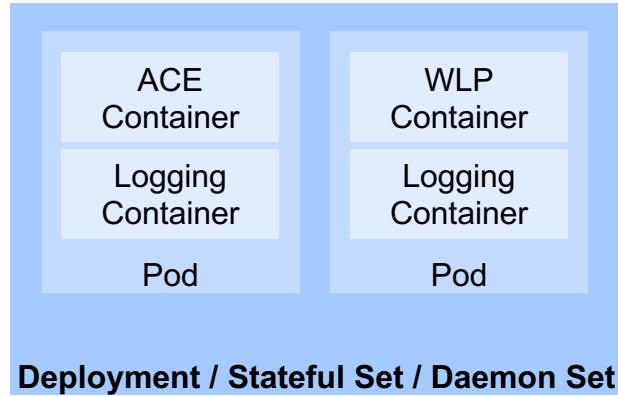
Deployable Bar Files

Global Configuration

ACE Binaries

Operating System - Ubuntu

# Key concepts of **Kubernetes**



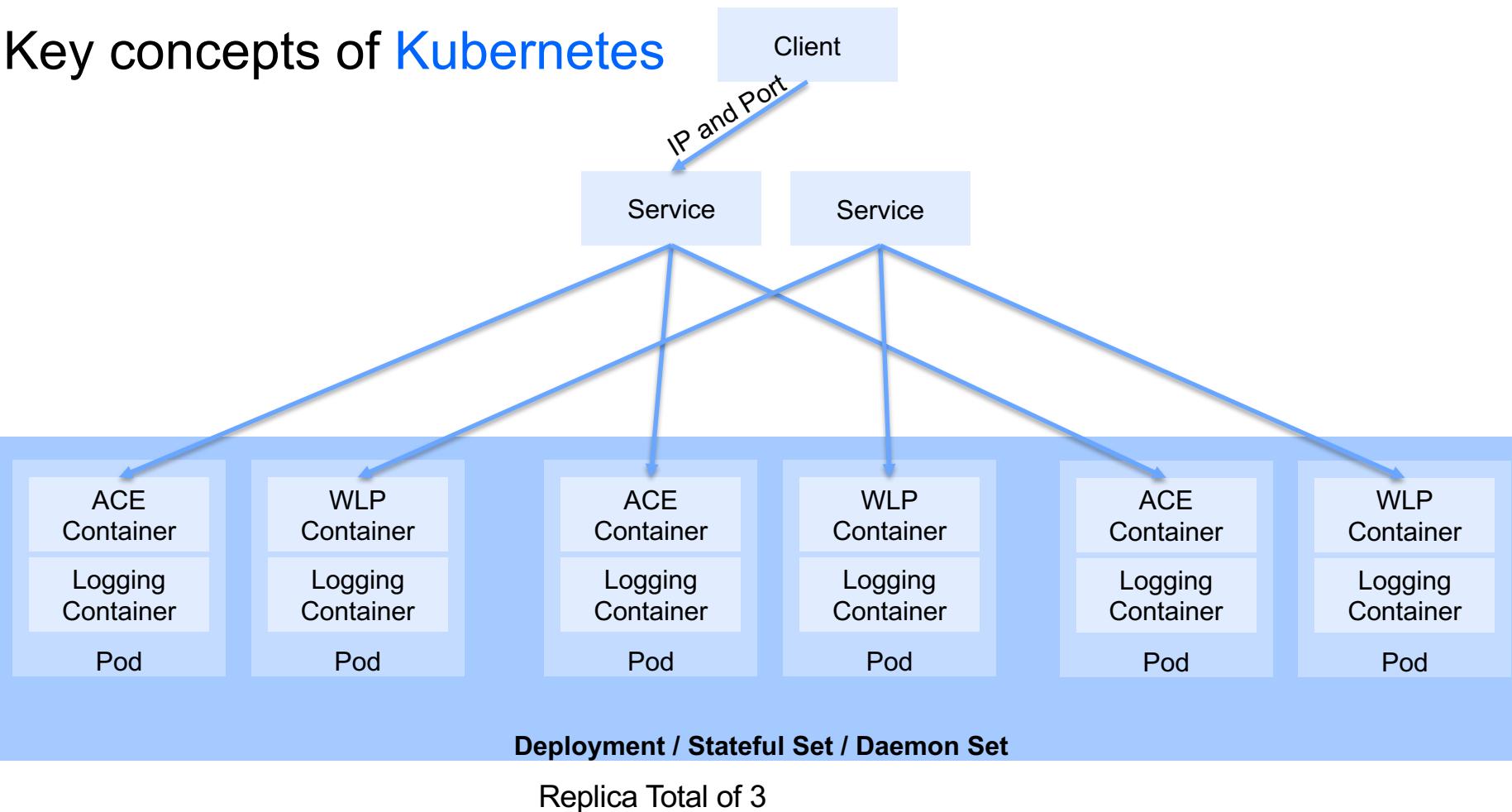
# Key concepts of **Kubernetes**



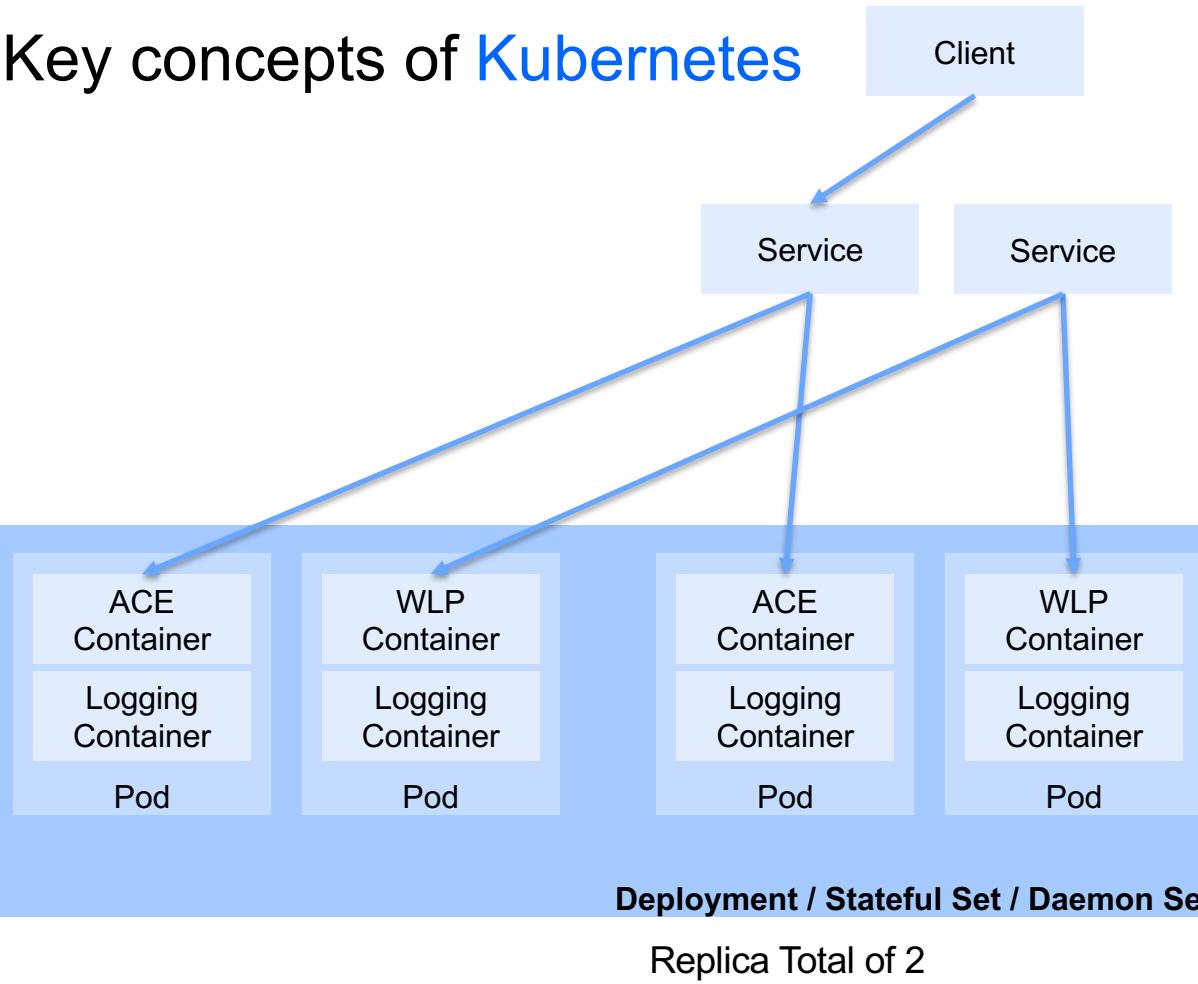
**Deployment / Stateful Set / Daemon Set**

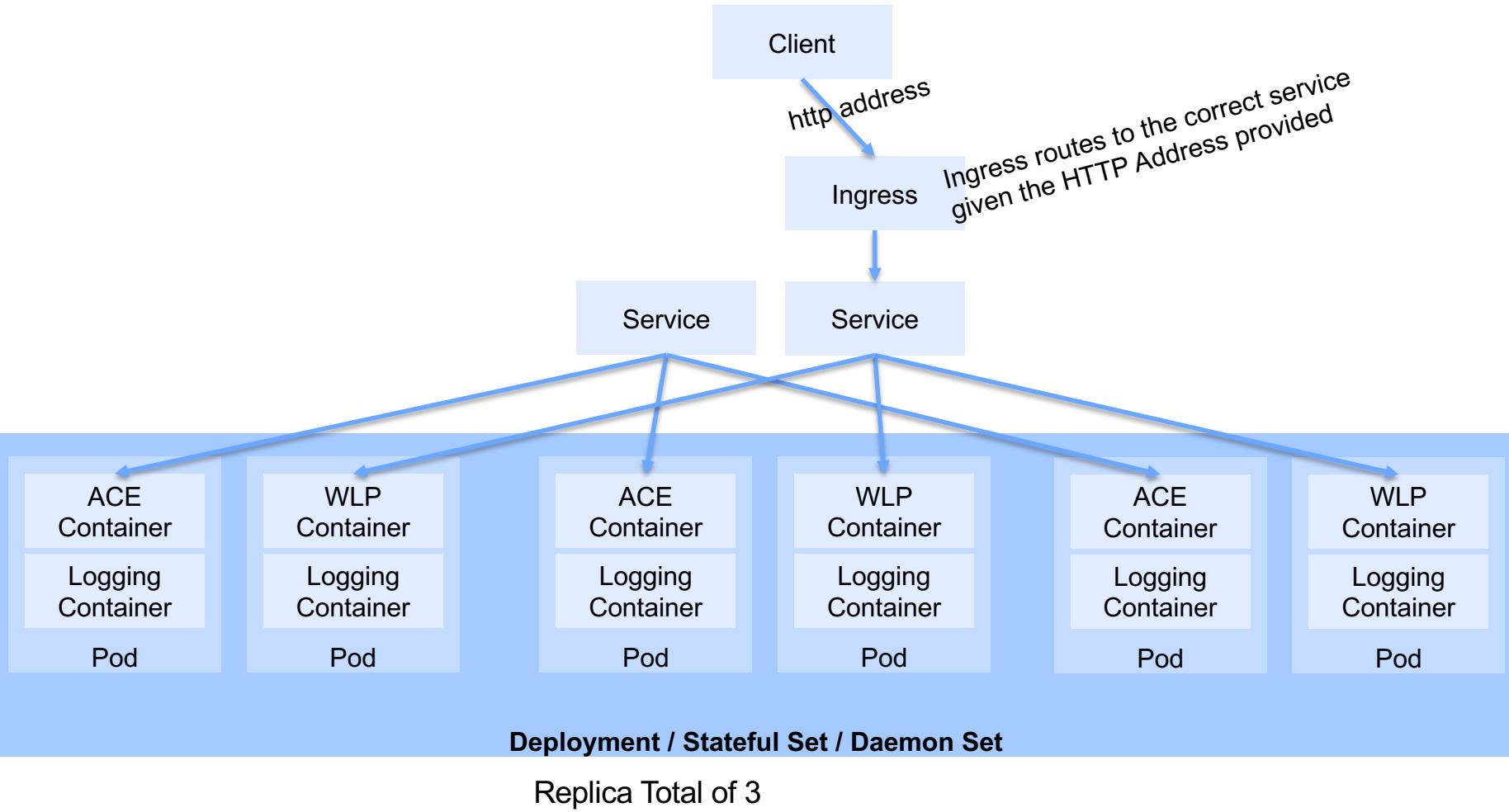
Replica Total of 3

# Key concepts of Kubernetes

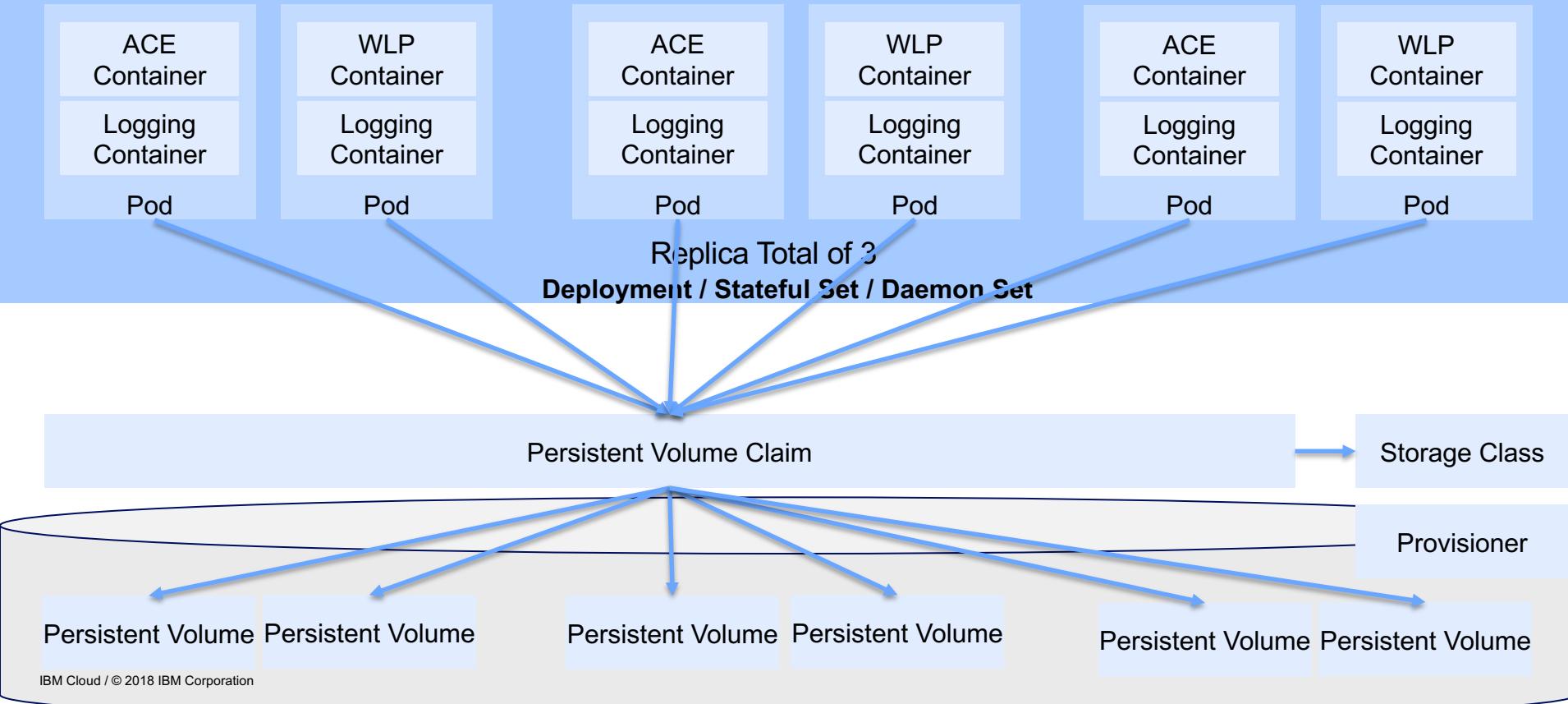


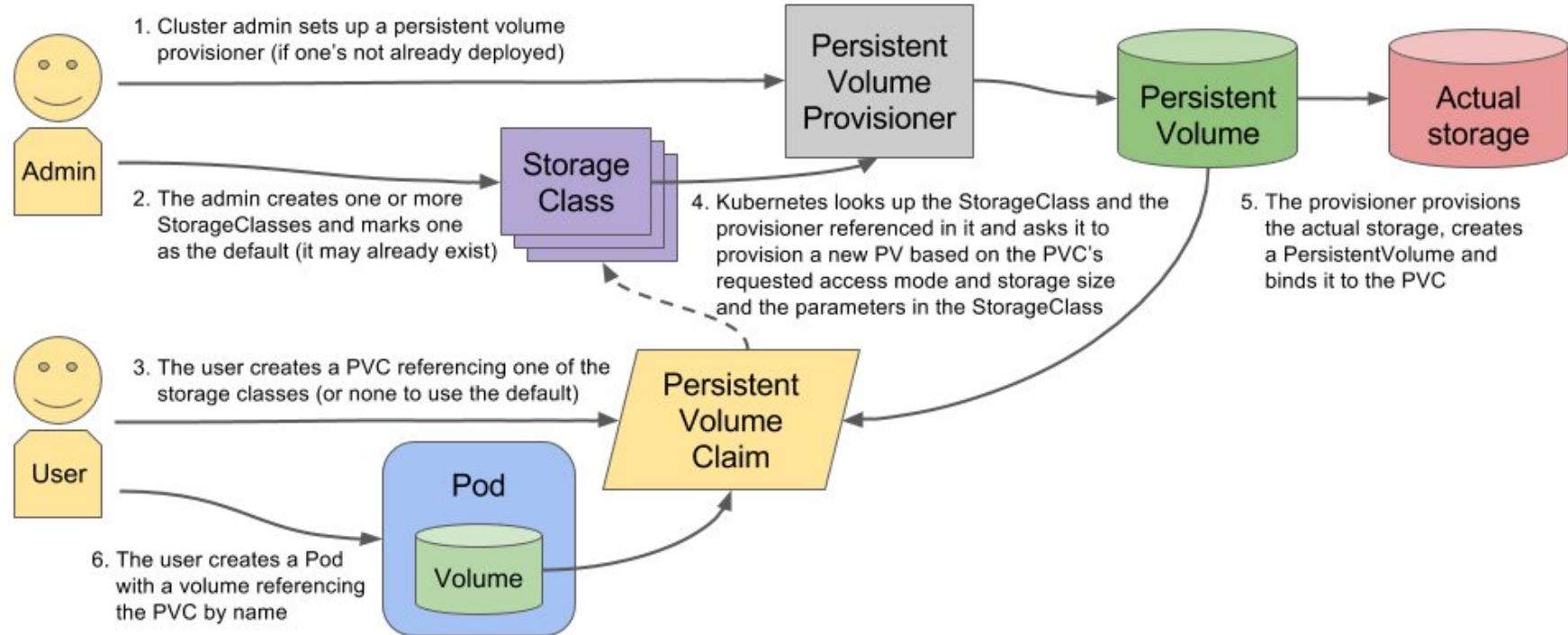
# Key concepts of Kubernetes





# Key concepts of Kubernetes





# What we have covered so far?

- Containers
- Pods
- Deployments
- Services
- Ingress
- PVC
- PV
- StorageClass

ANY Questions so far?

# What to consider when using Kubernetes

# High Availability Deployments



kubernetes



kubernetes



kubernetes



kubernetes



kubernetes



kubernetes



kubernetes

What is the hardest problem with High Availability?

# Managing State



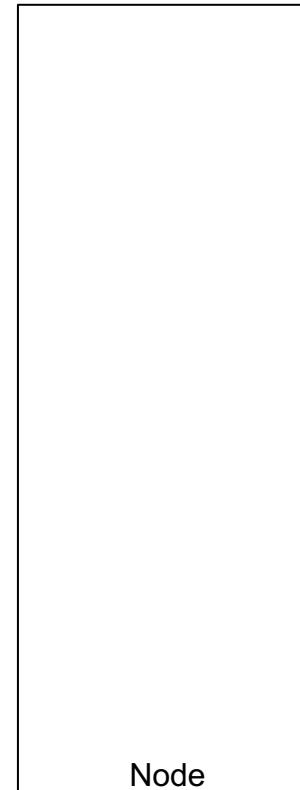
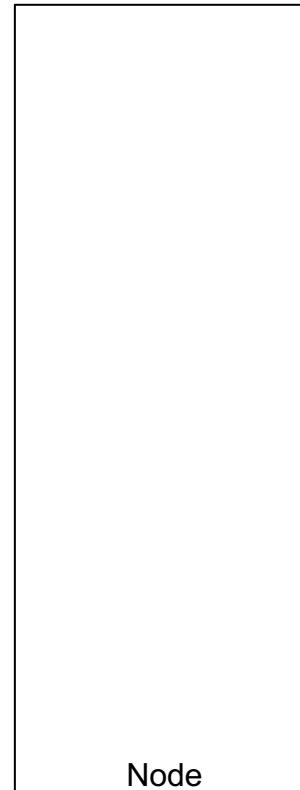
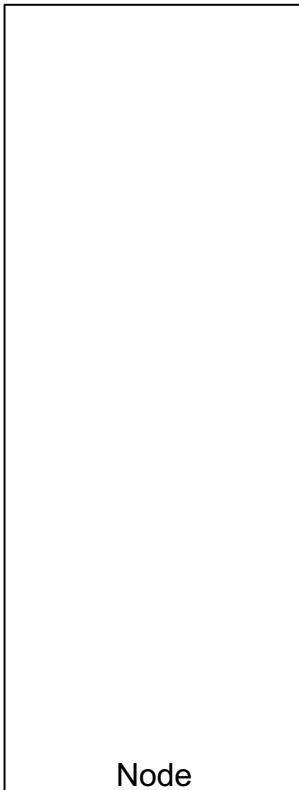
# Kubernetes delegates the HA of Application State to the storage solution

	Local				Cloud
	Gluster	CEPH Block	NFS	Host Path	EBS/ABS/IBS
<b>API Connect</b>	No	Yes	No	Limitation	Yes
<b>Event Streams</b>	Yes	Yes	Yes	Limitation	Yes
<b>ACE</b>	Yes	Yes	Yes	Limitation	Yes
<b>MQ</b>	Yes	Yes	Yes	Limitation	Yes

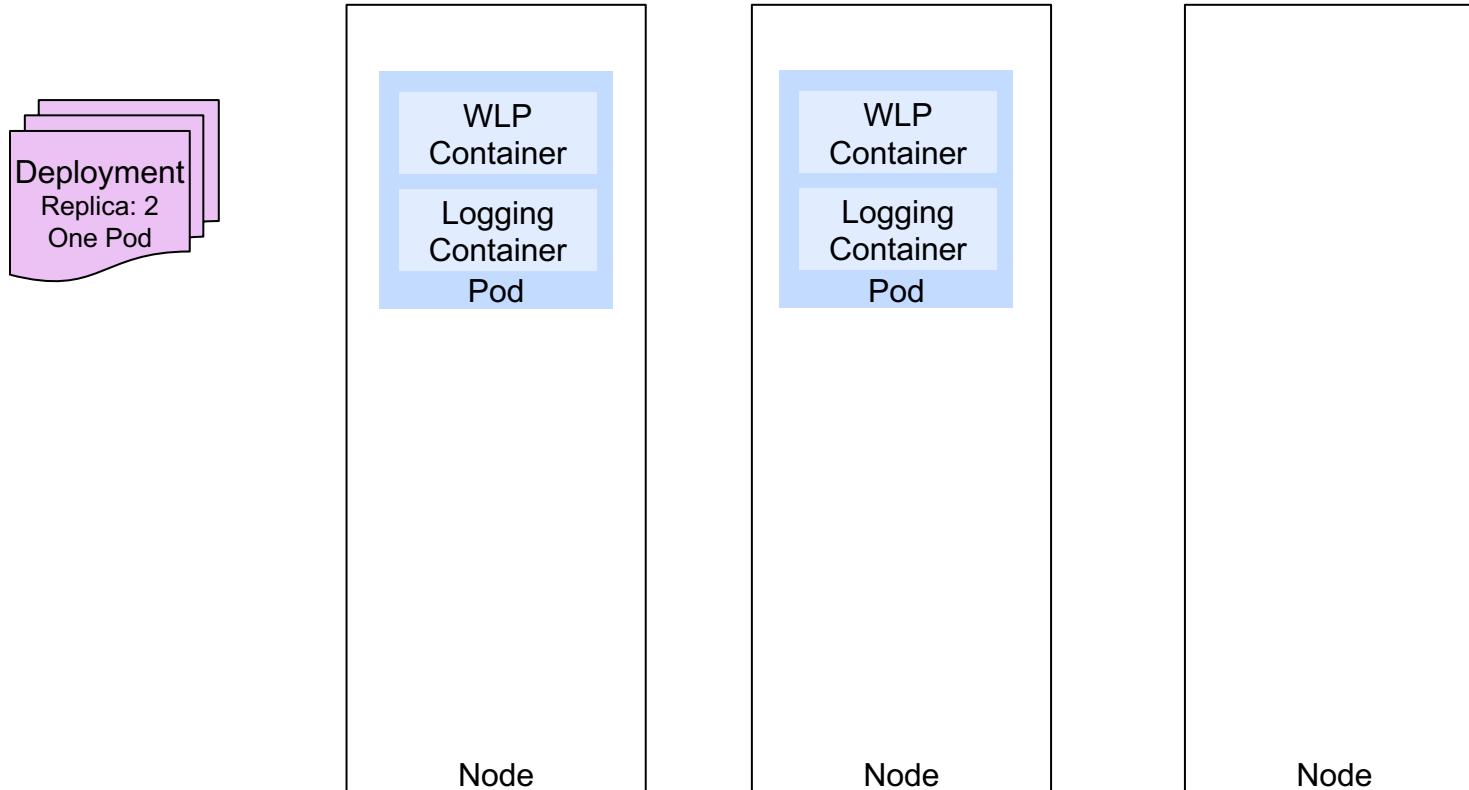
# Kubernetes HA – Three Types of Deployment

- Deployment
  - Pods are deployed to any node with capacity.
  - Scenario – Most Scenarios
- Stateful Set
  - Pod Replicas Start one at a time
  - Scenario – Storage
- Daemon Set
  - Pod Replicas have one on each node, no more no less.
  - Scenario – Critical Dependency that cannot afford to go down.

# Kubernetes HA - Deployment

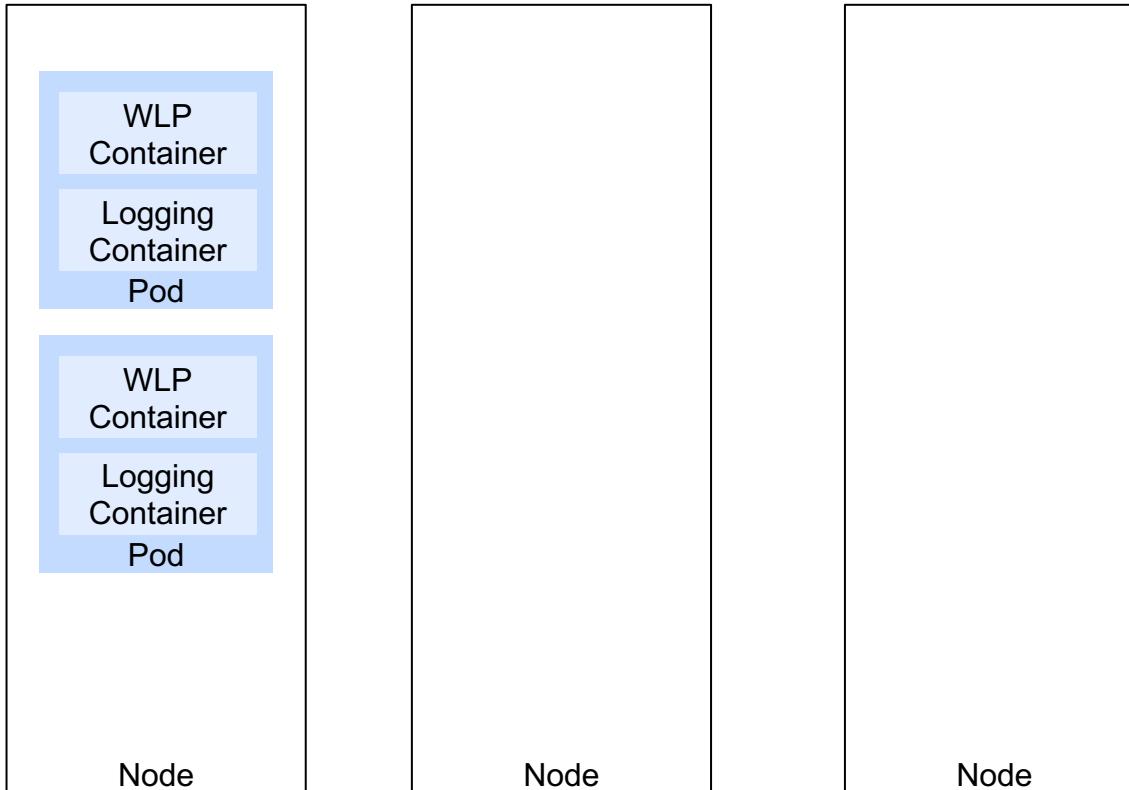


# Kubernetes HA - Deployment



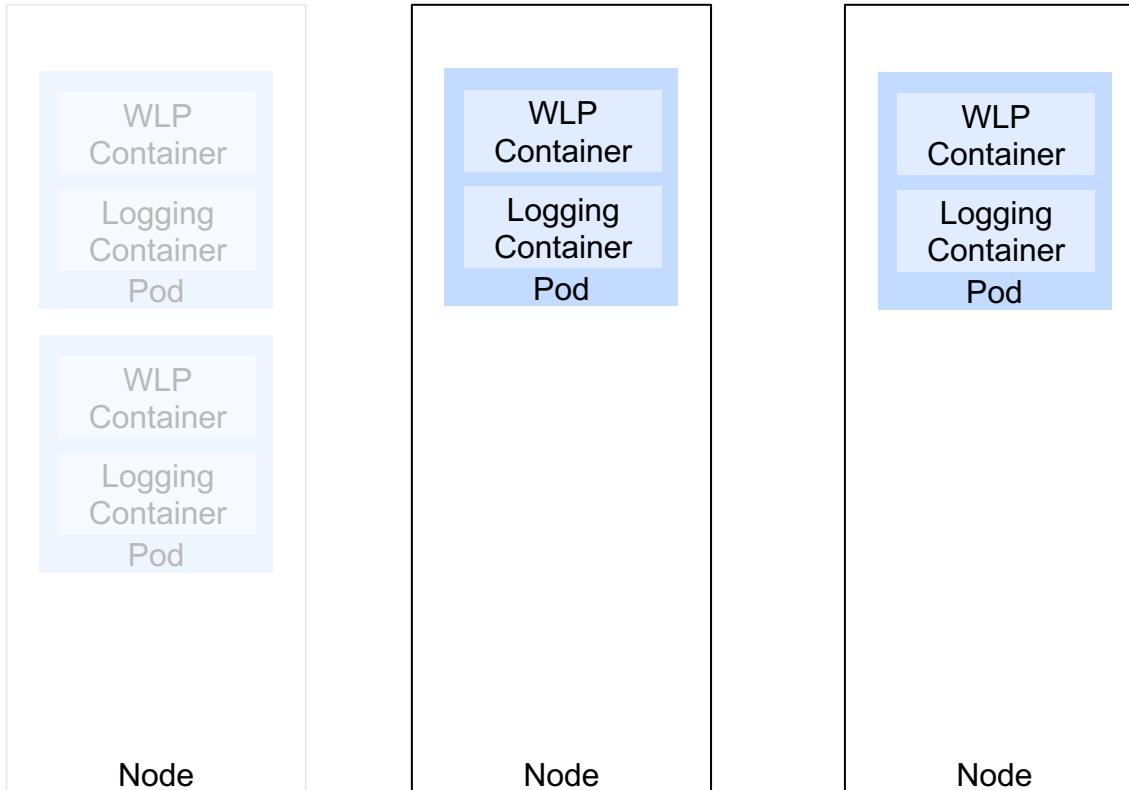
# Kubernetes HA - Deployment

Deployment  
Replica: 2  
One Pod



# Kubernetes HA – Deployment

Deployment  
Replica: 2  
One Pod

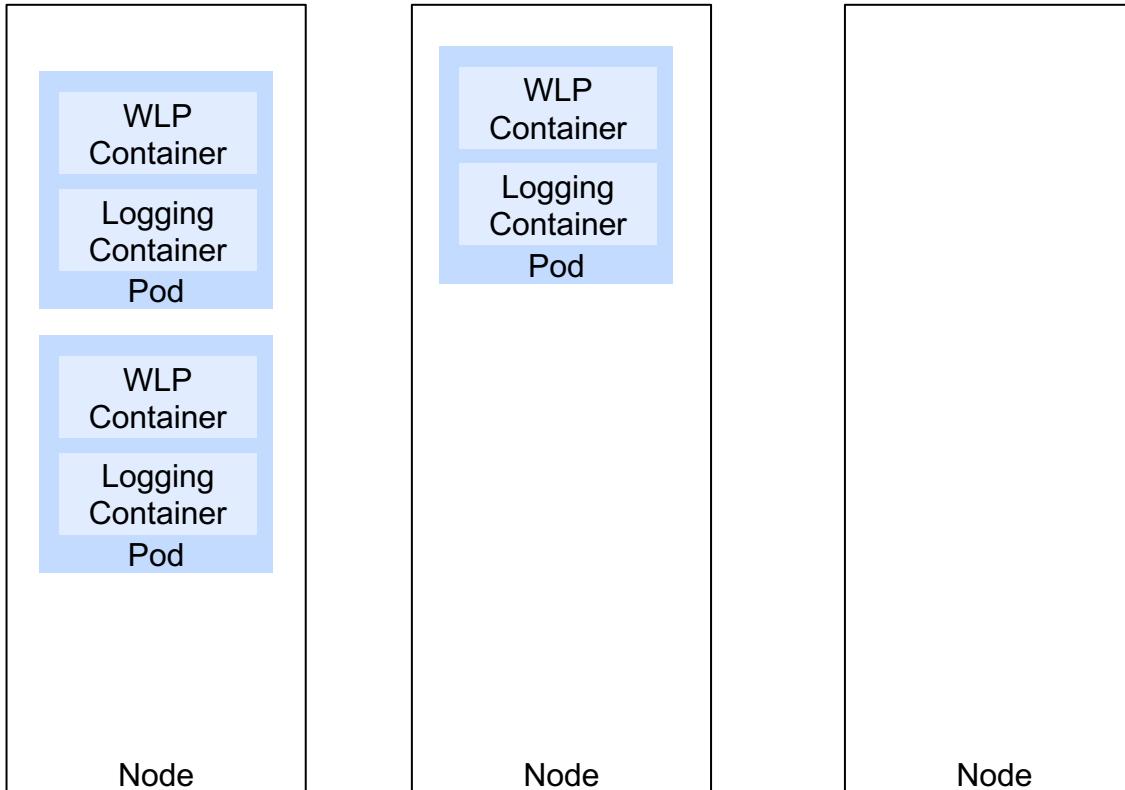


# Kubernetes HA – Node Affinity and Anti-Affinity

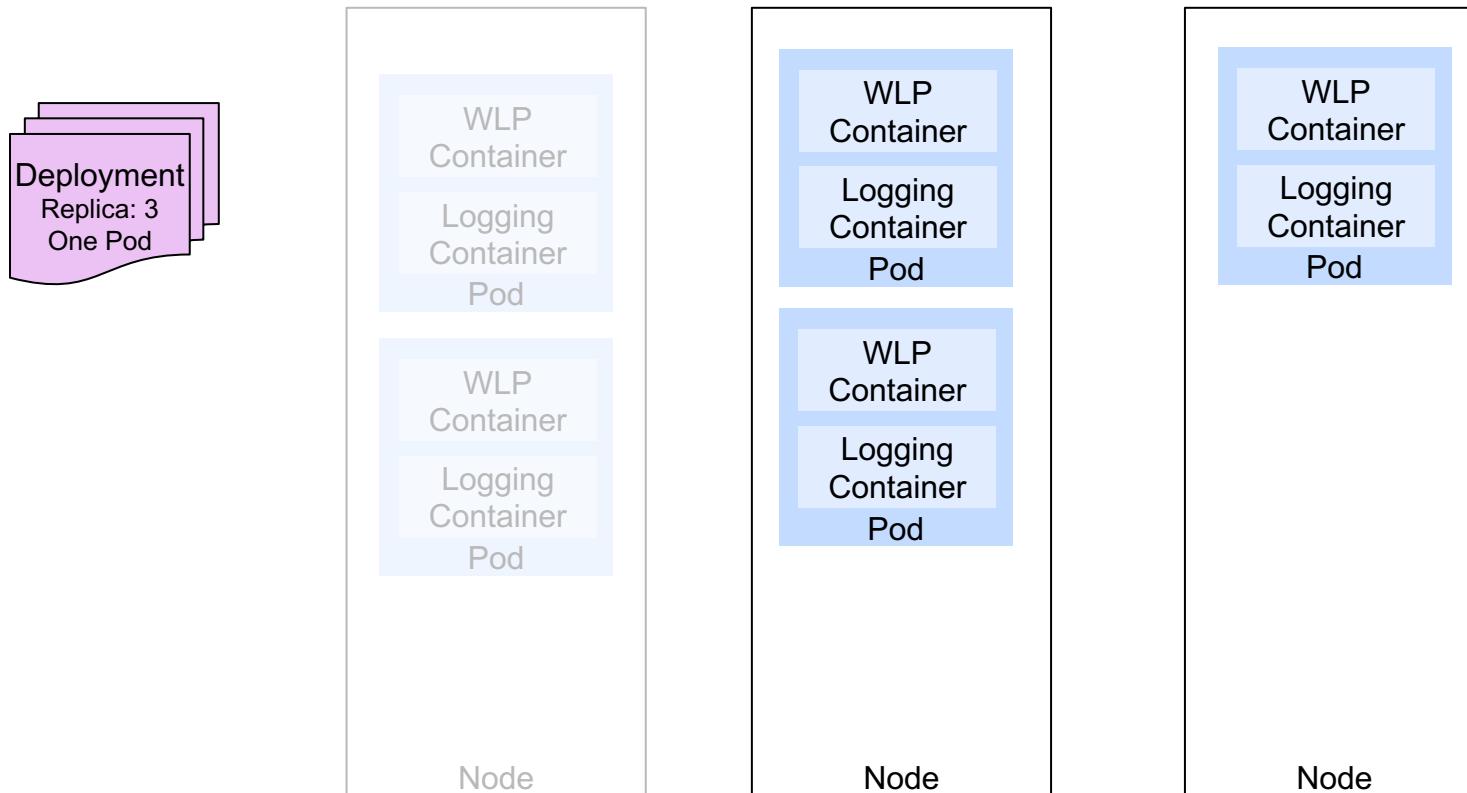
- Node Affinity
  - A set of rules the pod must follow at deploy time
    - E.g. Hostname
    - Custom Label (Can be used to identify availability zones)
- Node Anti Affinity
  - A set of rules the pod must not follow at deploy time
    - E.g. Hostname
    - Stopping it deploying on a node that already has an instance.

# Kubernetes HA - Deployment – Without any additional rules

Deployment  
Replica: 3  
One Pod

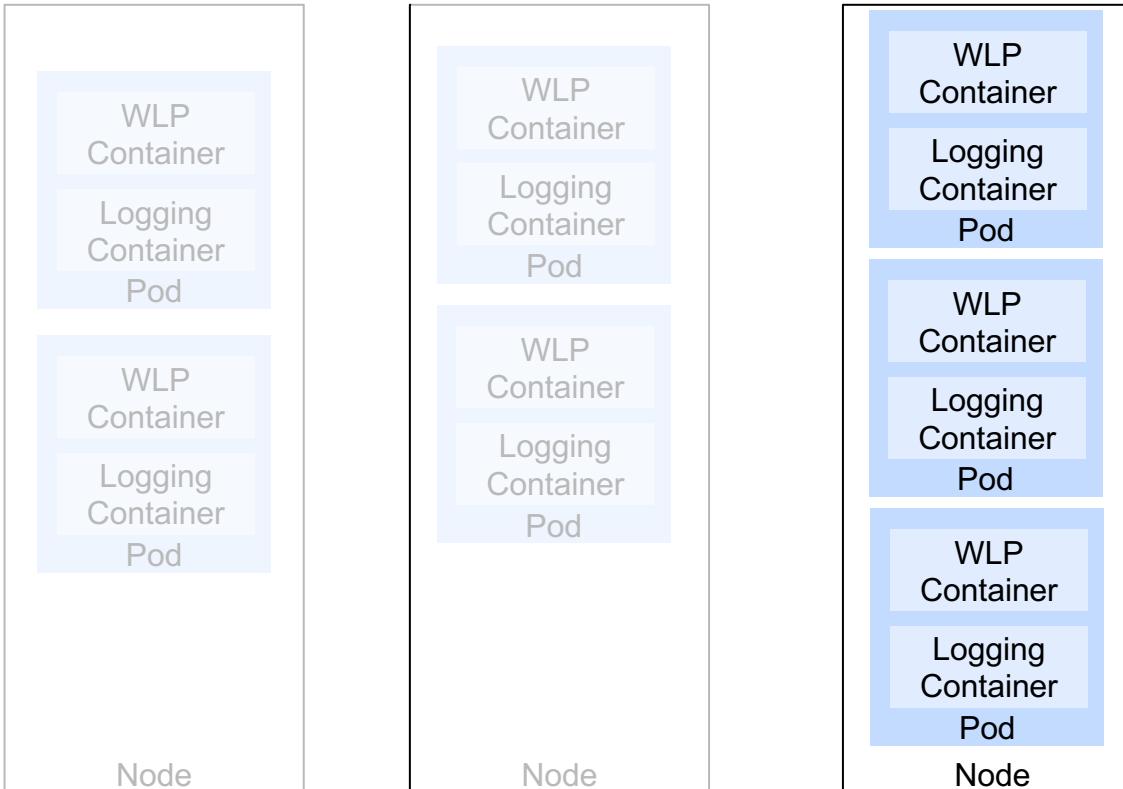


# Kubernetes HA - Deployment – Without any additional rules

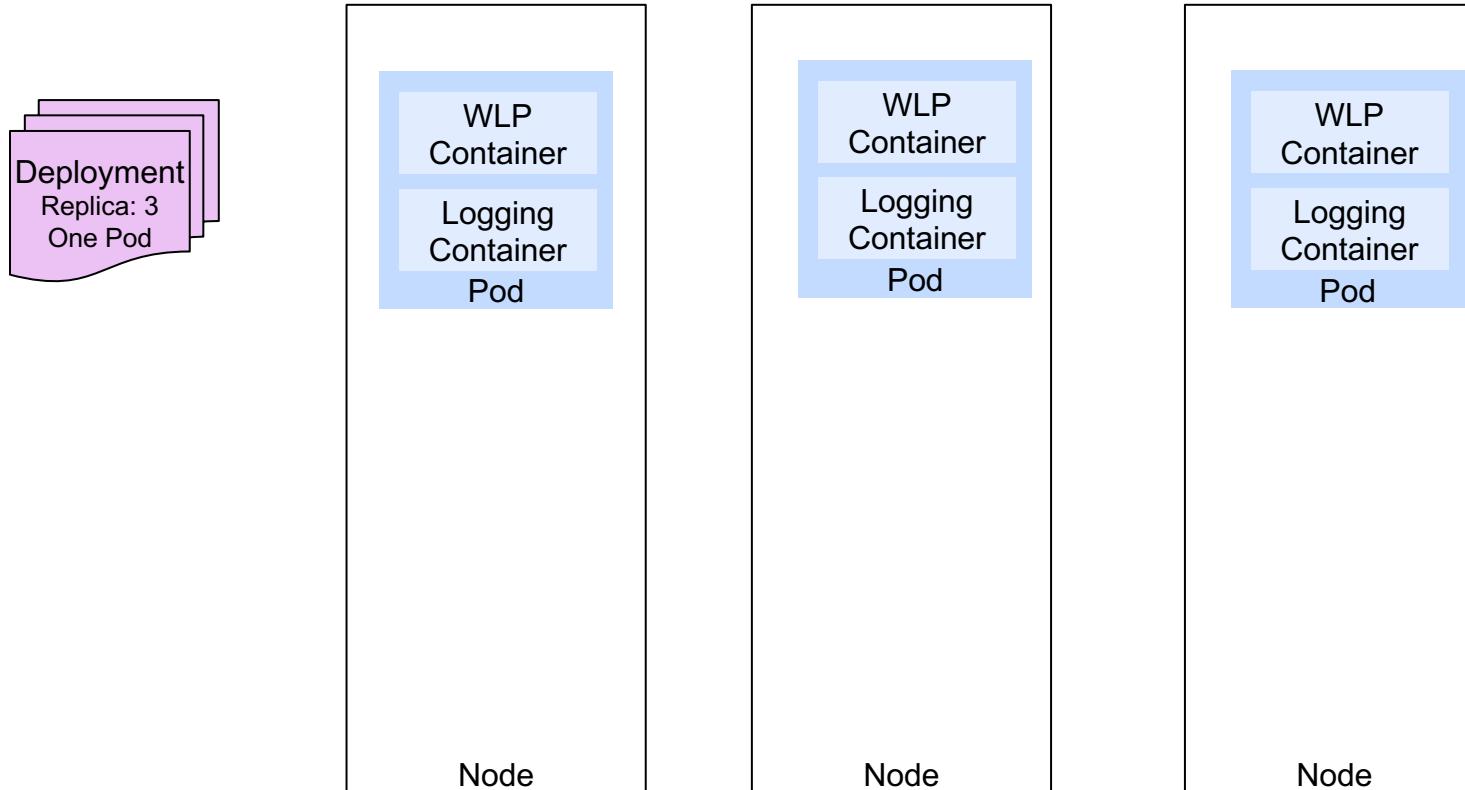


# Kubernetes HA - Deployment – Without any additional rules

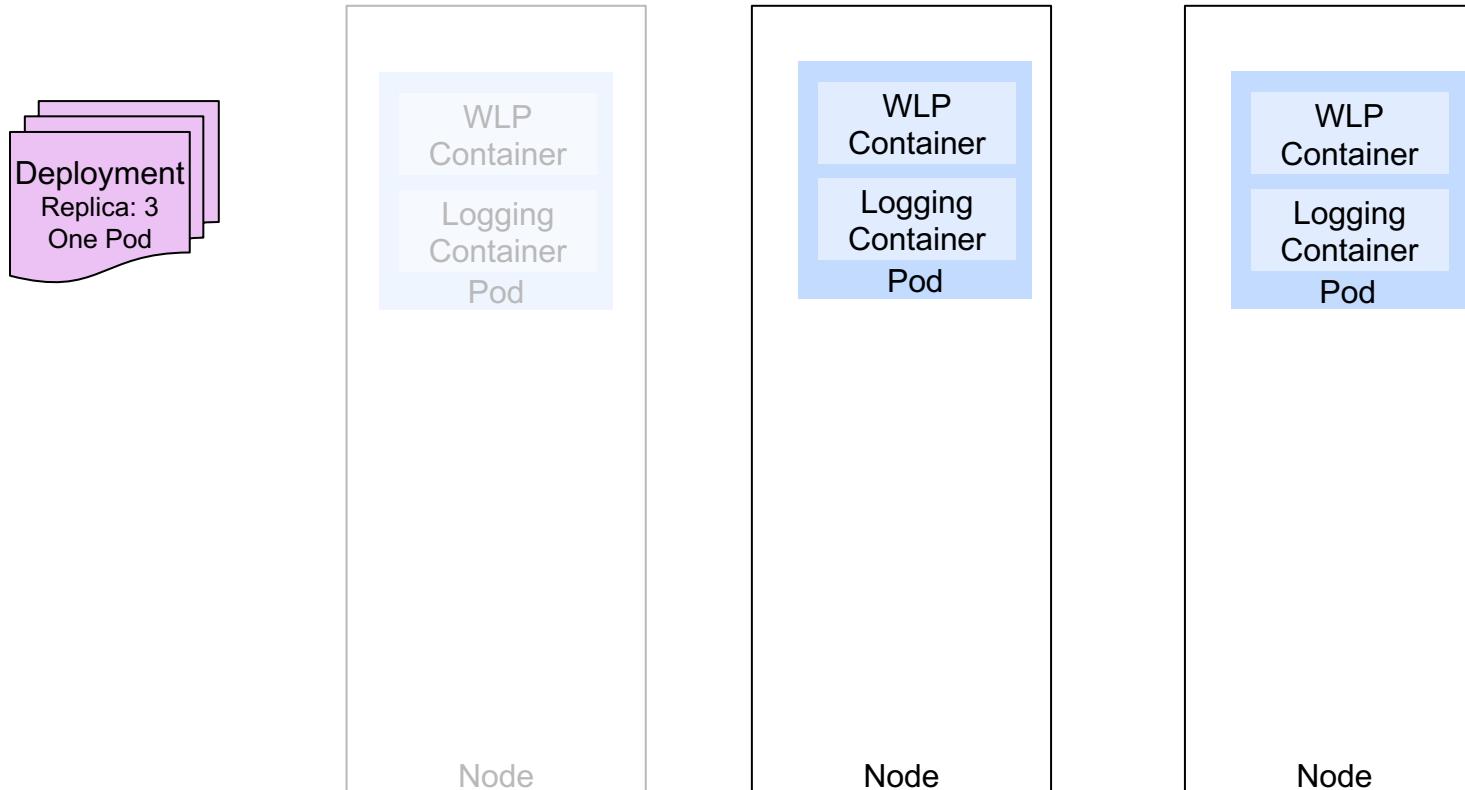
Deployment  
Replica: 3  
One Pod



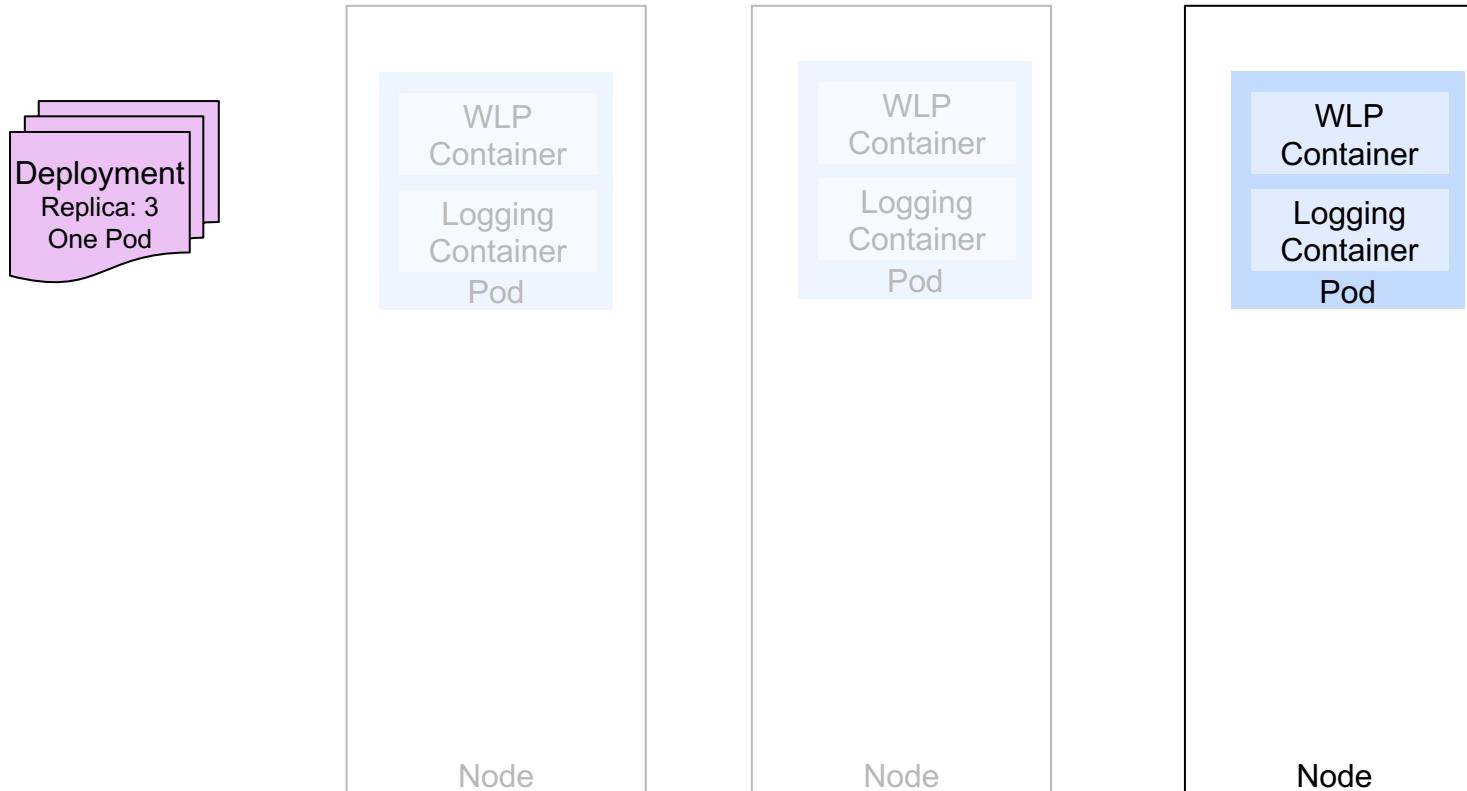
# Kubernetes HA - Deployment – With additional rules to ensure that no node has multiple instances



# Kubernetes HA - Deployment – With additional rules to ensure that no node has multiple instances



# Kubernetes HA - Deployment – With additional rules to ensure that no node has multiple instances



# WARNING

- Kubernetes has no system to try and spread pods across all nodes prior to having multiple nodes on the same pod.
- When looking at High Availability understand what the recovery time of a pod is including mapping its storage.

# Lessons from the field - 1

## Kubernetes provides fixes for the four latest versions 1.x versions

18th December v1.10.12

14th December v1.11.6

13th December v1.12.4

13th December v1.13.1

3rd December v1.13.0

26th November v1.12.3

26th November v1.11.5

26th November v1.10.11

13rd November v1.10.10

25th October v1.11.4

24th October v1.12.2

16th October v1.10.9

5th October v1.12.1

28th September v1.9.11

27th September v1.12.0

14th September v1.10.8

9th September v1.11.3

- New Major Version every Three Months,
- New Minor Version every Month
- No concept of fixes unless you want to go in and change yourself.
- No Liability
- This list does not include Beta Releases

# Lessons from the field - 2

## **No single supplier of crucial components.**

i.e. Helm, Calico and Kubernetes are built by different communities.

New version of components **can** cause breakages.

# Lessons from the field - 3

- Persistent Storage only left beta in the last year.
- Many critical area of function that are being actively used are still in Alpha and Beta

# Lessons from the field - 4

- Pods must be designed to cleanly die and come back.

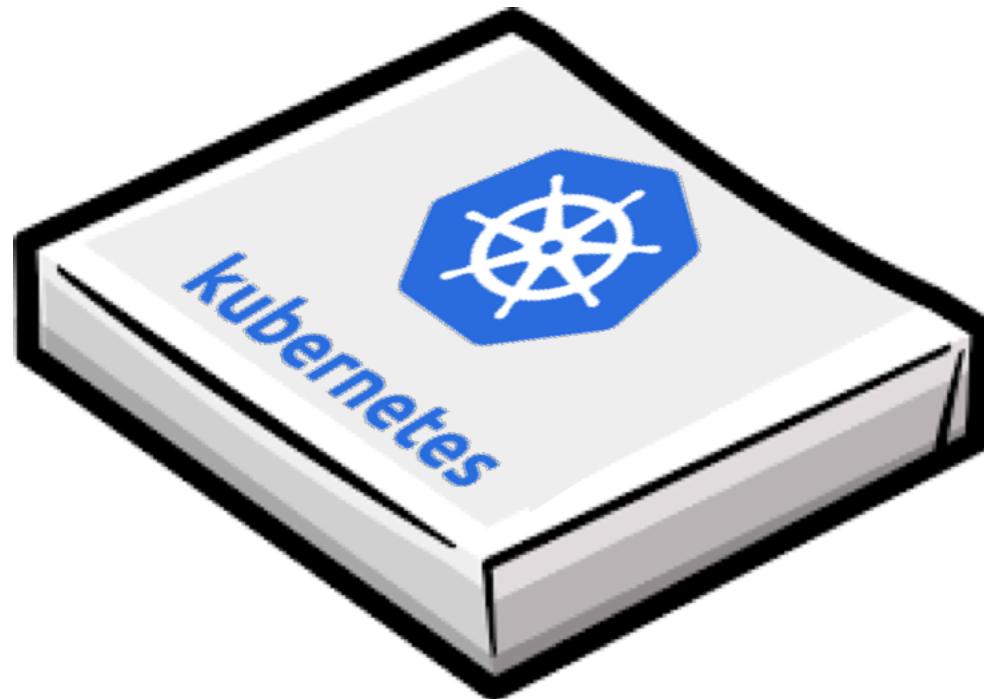
# Lessons from the field - 5

- Plan HA, Plan HA, Plan HA, Plan HA, Plan HA
- Document HA
- Test HA, Test HA, Test HA

# Lessons from the field - 6

- Avoid kubernetes clusters going across data centers unless there is a low latency network between them

# Key takeaways from projects with Kubernetes



# Kubernetes is not a magic bullet

# Kubernetes is powerful if designed correctly

Kubernetes will challenge the beliefs of anyone experienced in an operation or infrastructure role.

**Do NOT just lift and shift**

# Thank you

Chris Phillips  
Master Inventor  
SWAT Integration Architect

---

[Chris.Phillips@uk.ibm.com](mailto:Chris.Phillips@uk.ibm.com)  
ibm.com

Ivan Pryanichnikov  
Pan-IMT Technical Sales Specialist

---

[ivanpran@ru.ibm.com](mailto:ivanpran@ru.ibm.com)  
ibm.com

