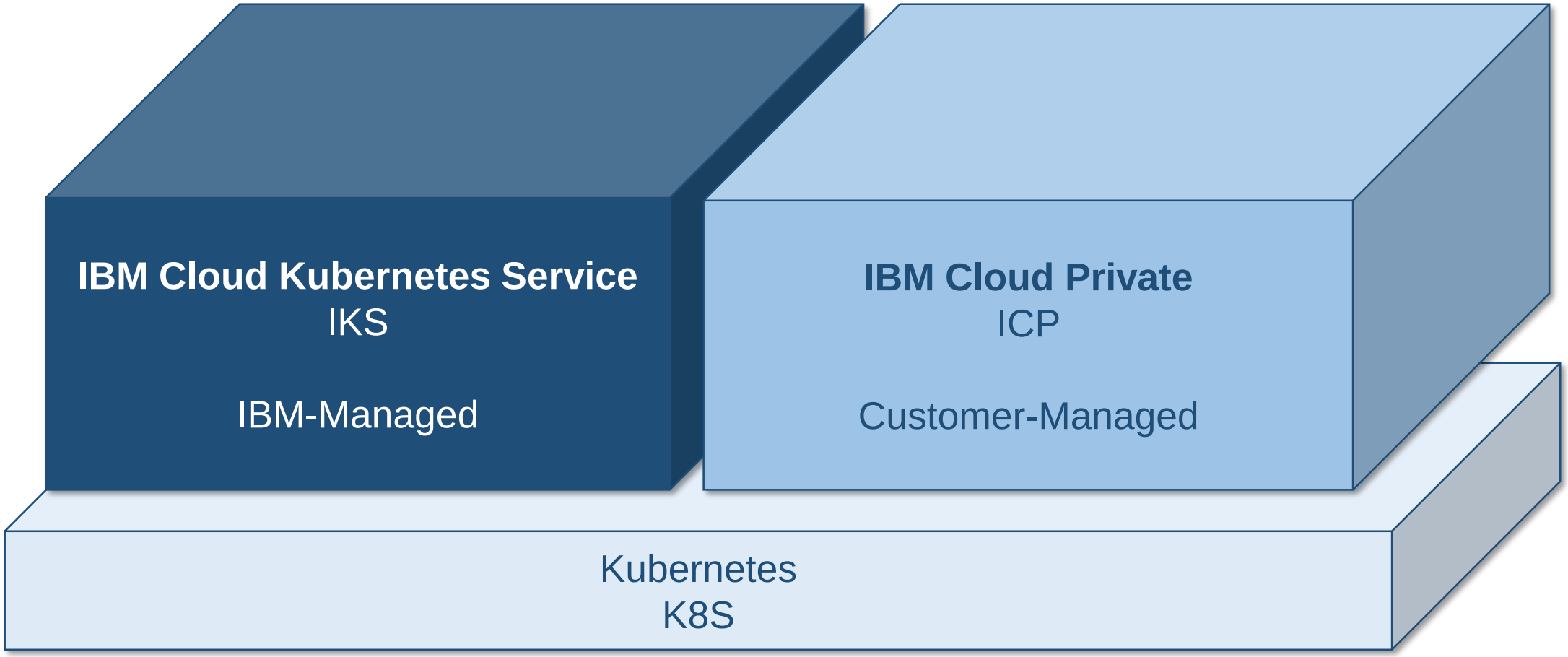
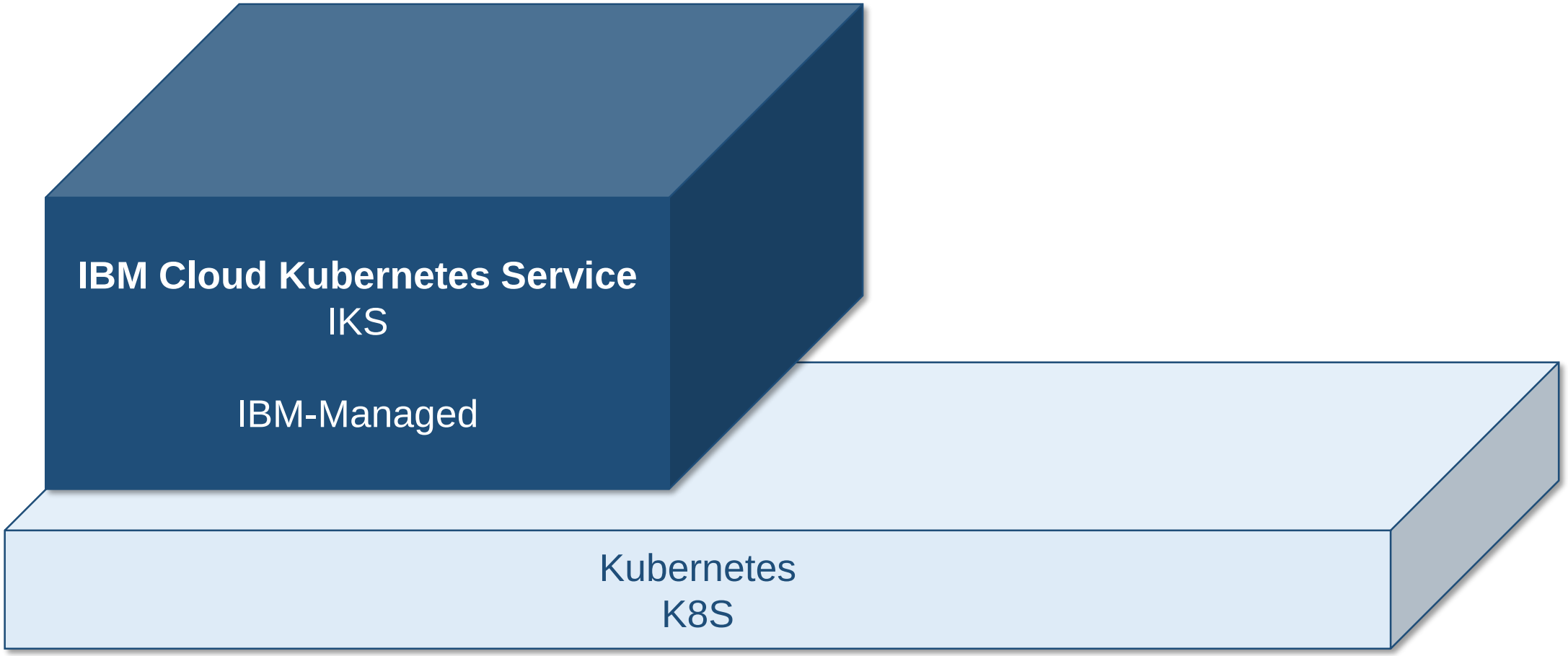
IBM Cloud Containers Workshop

IBM Cloud Kubernetes Services

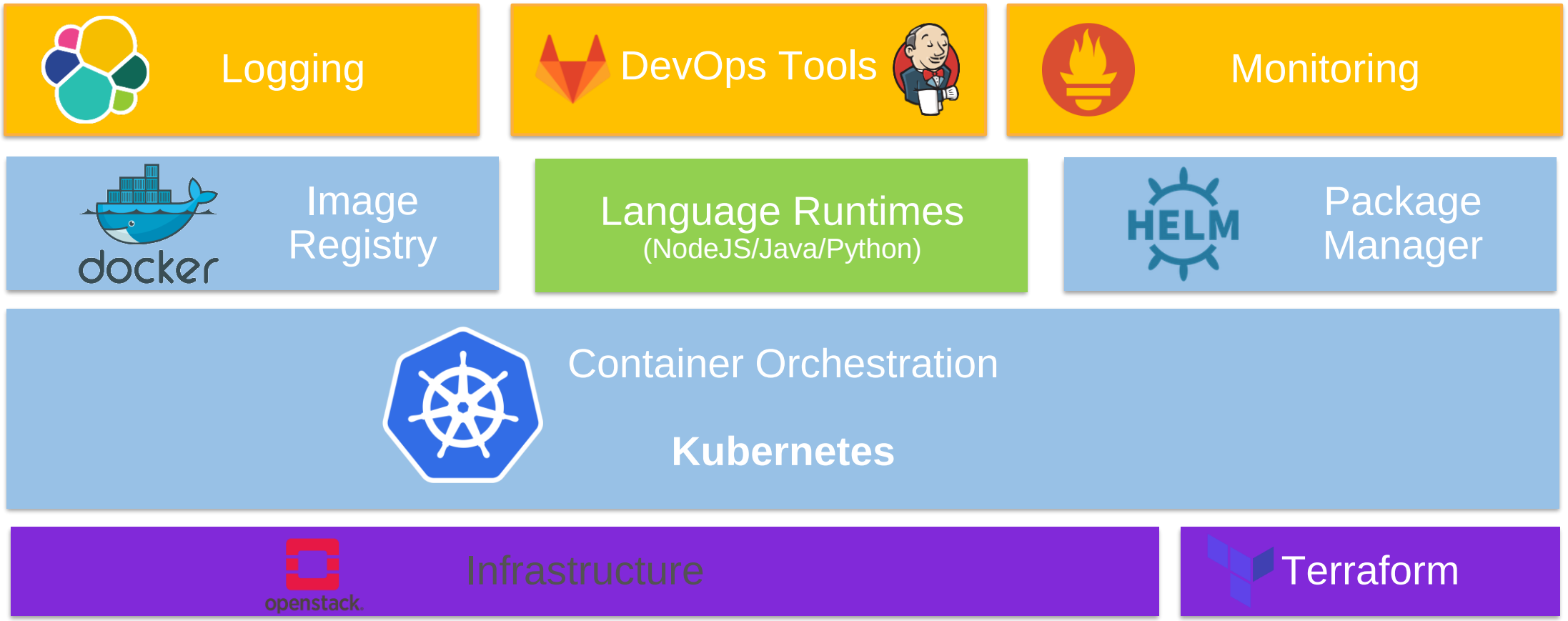
IBM Solutions based on Kubernetes



IBM Cloud Kubernetes Services



A few Kubernetes companions (based on Open Technologies)



IBM Cloud Container Registry

IBM **managed stand-alone Docker image registry**

Pre-integrated with our Kubernetes Service.

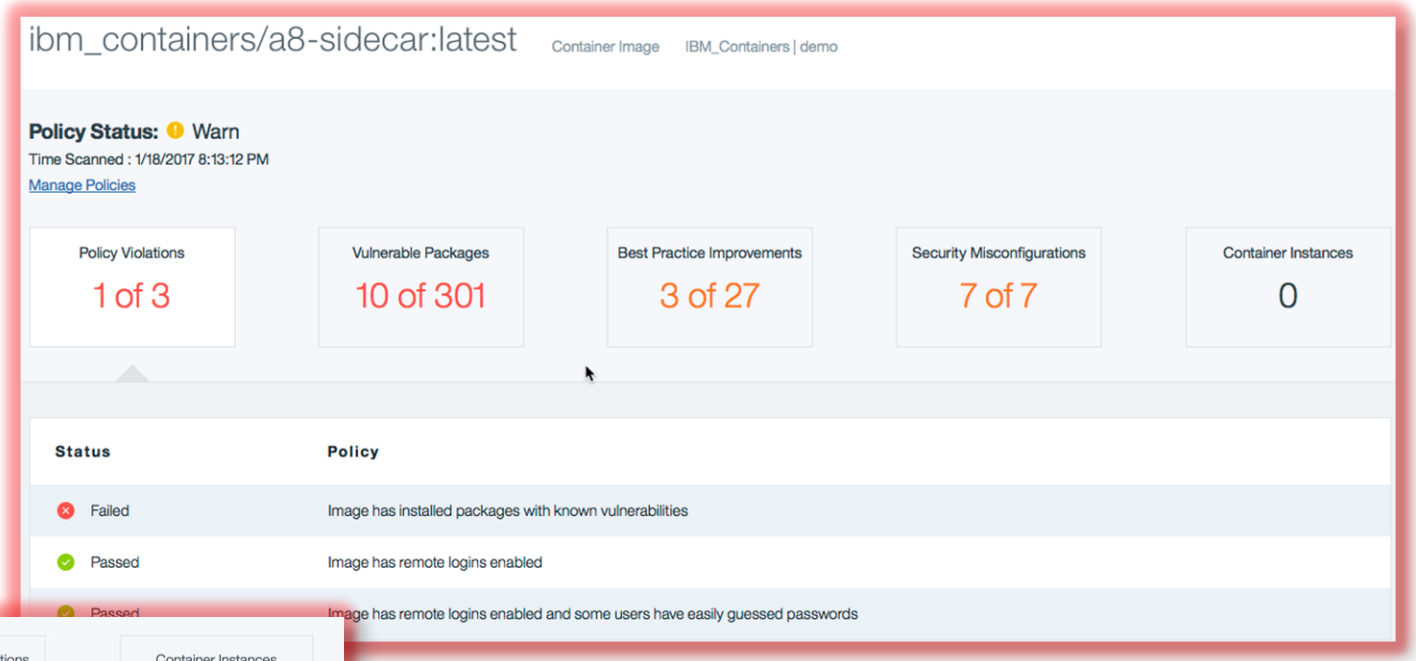
Secure with integration to IBM Identity and Access Manager.

No need to configure and maintain store for your images.

Advanced capabilities such as vulnerability scanning, malware detection, policy scanning, image signing, and deployment policy enforcement.

Vulnerability Advisor

Policy Violations



Vulnerable Packages

Integration between Vulnerability Advisor and IBM X-Force

				Description	Corrective Action
				Several security issues were fixed in curl.	Upgrade libcurl3 to at least version 7.35.0-1ubuntu2.10
				Several security issues were fixed in Python.	Upgrade libpython3.4-stdlib to at least version 3.4.3-1ubuntu1~14.04.5, Upgrade python3.4 to at least version 3.4.3-1ubuntu1~14.04.5, Upgrade libpython3.4-minimal to at least version 3.4.3-1ubuntu1~14.04.5, Upgrade python3.4-minimal to at least version 3.4.3-1ubuntu1~14.04.5
				The GD library could be made to crash or run programs if it processed aspecially crafted image file.	Upgrade libgd3 to at least version 2.1.0-3ubuntu0.5
Medium	apt	3156-1	CVE-2016-1252	An attacker could trick APT into installing altered packages.	Upgrade apt to at least version 1.0.1ubuntu2.17
Medium	vim-common	3139-1	CVE-2016-1248	Vim could be made run programs as your login if it opened a speciallycrafted file.	Upgrade vim-common to at least version 2:7.4.052-1ubuntu3.1
Medium	tar	3132-1	CVE-2016-6321	tar could be made to overwrite files.	Upgrade tar to at least version 1.27.1-1ubuntu0.1
Low	libdbus-1-3	3116-1	CVE-2015-0245	Several security issues were fixed in DBus.	Upgrade libdbus-1-3 to at least version 1.6.18-0ubuntu4.4

http://www-03.ibm.com/security/xforce/

IBM Cloud Kubernetes Service

A **certified, managed Kubernetes service**

Built-in **security and isolation** to enable rapid delivery of apps.

Available in six IBM regions WW, including **20+ datacenters**.

Fully **dedicated, single tenant clusters** deployed within customer account and network

Seamless integration with IBM Cloud services

Portability with native Kubernetes experience and **full API support**

IBM **Cloud Kubernetes Service**

Kubernetes Capabilities

Intelligent Scheduling

Self-healing

Horizontal scaling

Service discovery & load balancing

Automated rollouts and rollbacks

Secret and configuration management

IBM Cloud Kubernetes Service capabilities

Simplified cluster management

Flexible cluster topologies

Container security & isolation

Extend with IBM Cloud & Watson

Native open-source experience

Integrated operational tools

Pace of evolution of Kubernetes

Supported?	Version	IBM Cloud Kubernetes Service release date	IBM Cloud Kubernetes Service unsupported date
✓	1.14	07 May 2019	Mar 2020 †
✓	1.13	05 Feb 2019	Dec 2019 †
✓	1.12	07 Nov 2018	Sep 2019 †
!	1.11	14 Aug 2018	27 Jun 2019 †
!	1.10	01 May 2018	15 May 2019
✗	1.9	08 Feb 2018	27 Dec 2018
✗	1.8	08 Nov 2017	22 Sep 2018
✗	1.7	19 Sep 2017	21 Jun 2018
✗	1.6	N/A	N/A
✗	1.5	23 May 2017	04 Apr 2018

Release history for IBM Cloud Kubernetes Service.

Select the Kubernetes Version

Native open-source experience

Supported Kubernetes versions:

- Latest 1.14.3
- Stable, Default 1.13.7
- Stable 1.12.9
- Deprecated 1.11.10

1.14.3_1506

Kubernetes version

IKS version

Newest version supported is tagged **Latest**
The n-2 versions are marked **Stable**
The n-1 version also marked **Default**

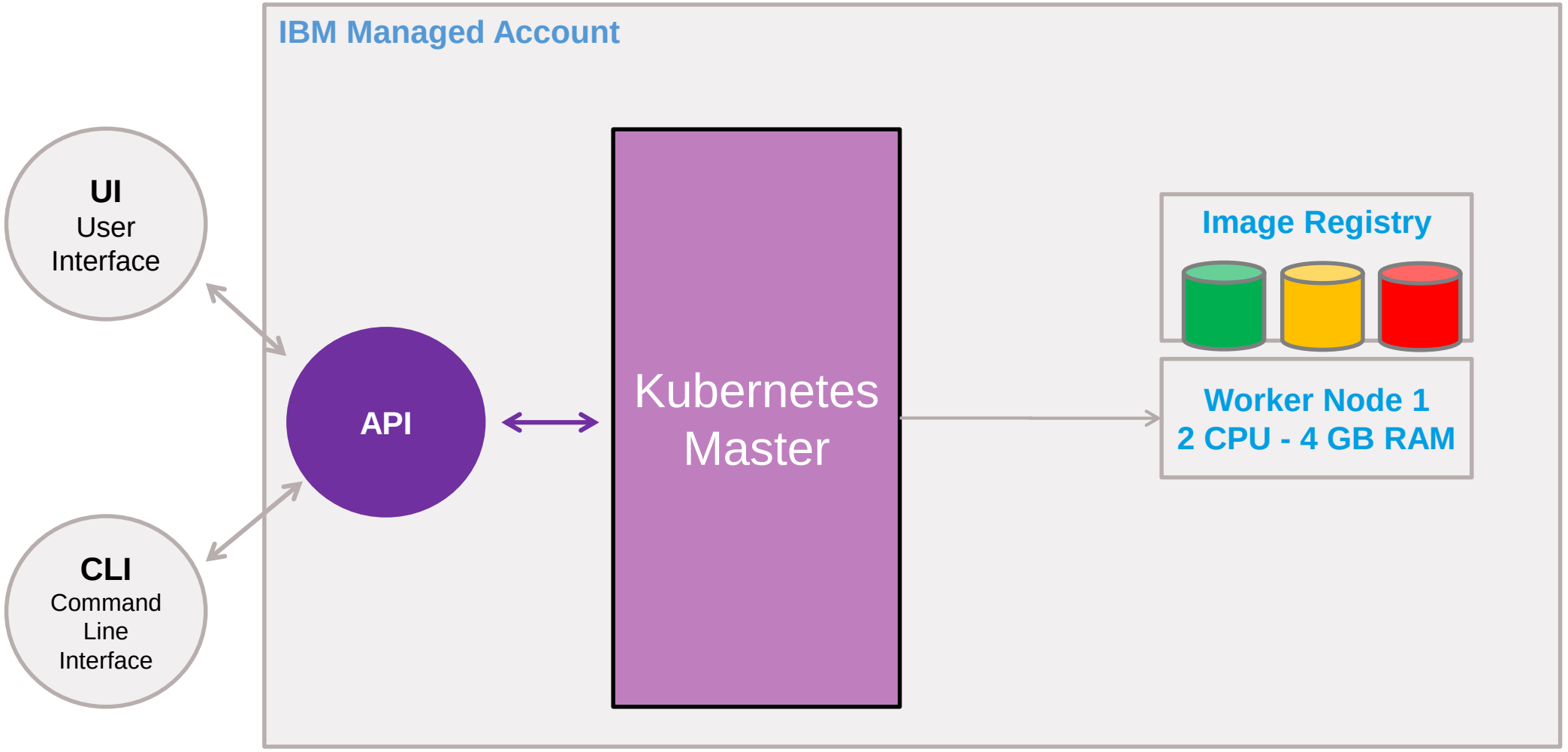
DEVELOPER

> ibmcloud ks kube-versions

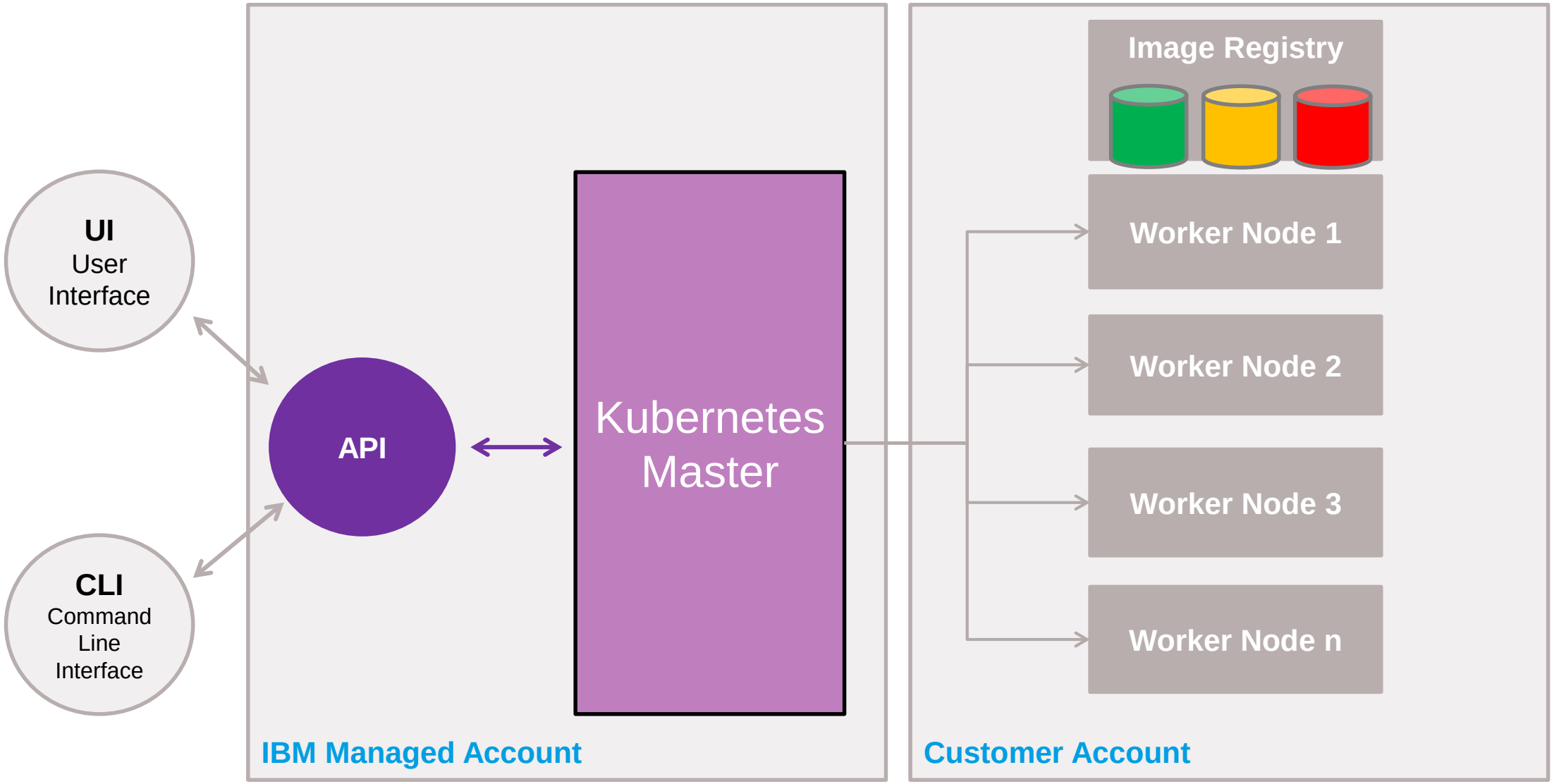
Lite Cluster – Single Worker Node


Simplified cluster management


Flexible cluster topologies



Standard Cluster - fully customizable, production-ready

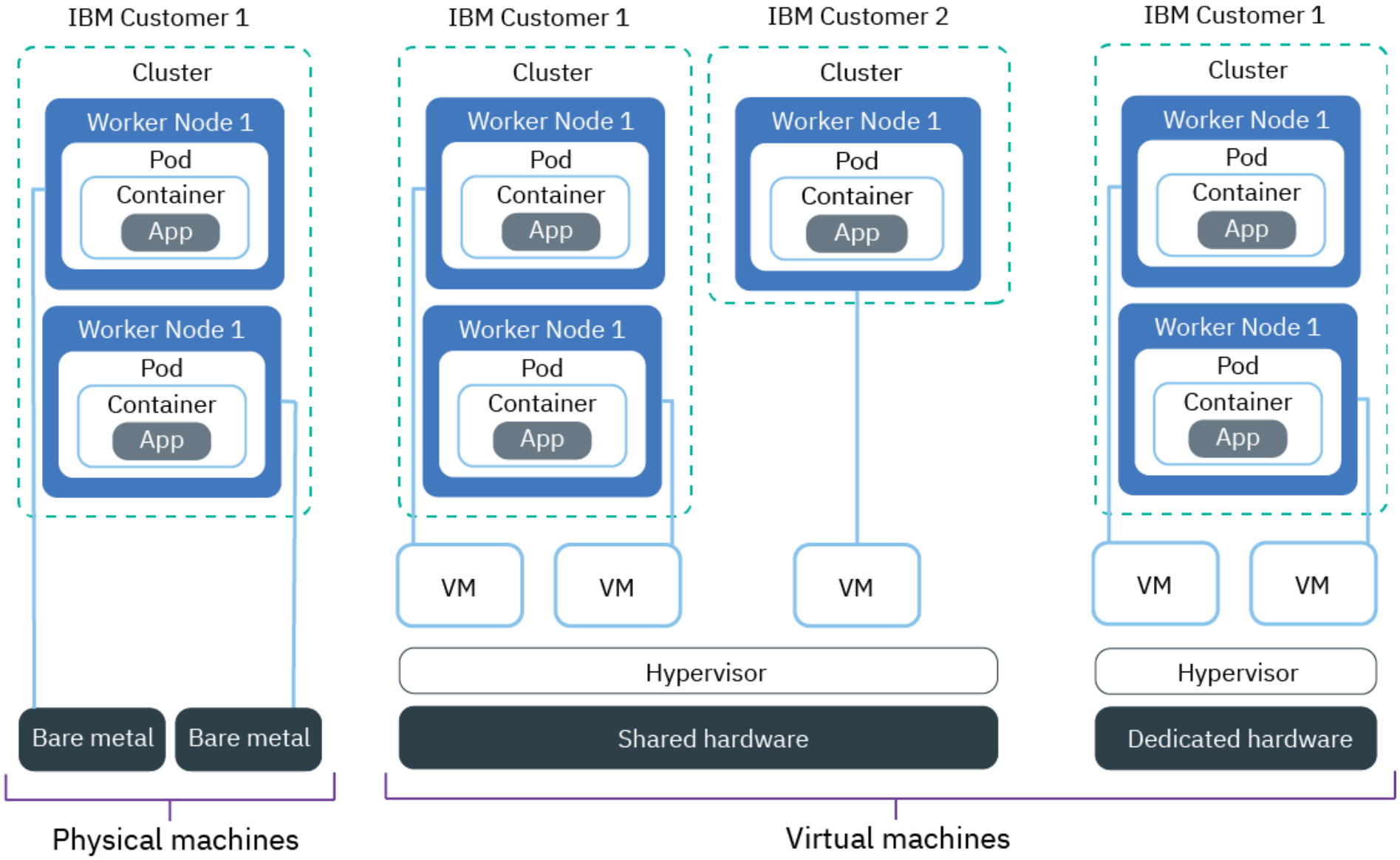


Select the type of machines


Container security & isolation


Simplified cluster management


Flexible cluster topologies

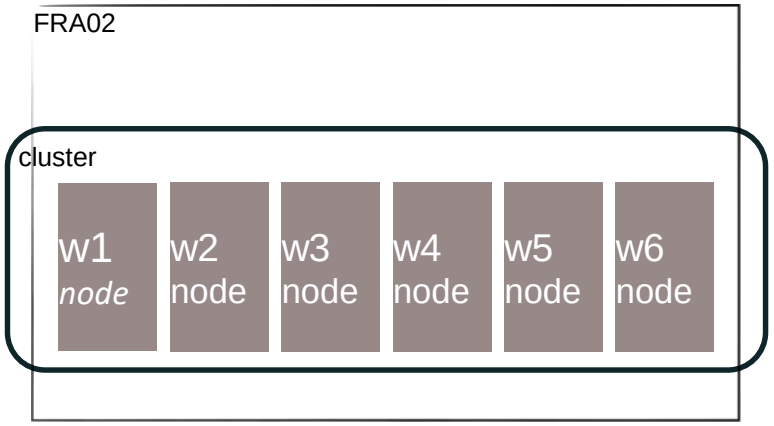


Single Zone Cluster


Simplified cluster management


Flexible cluster topologies

EU-DE



- Cluster created in a single location (datacenter)
- Worker nodes provisioned in a single location
- Master managed by IKS runs in the same datacenter within the IKS account

Multizone Cluster


Simplified cluster management


Flexible cluster topologies

What is it?

A single Kubernetes cluster that has worker nodes spread across multiple failure domains (i.e., zones).

Why is it valuable?

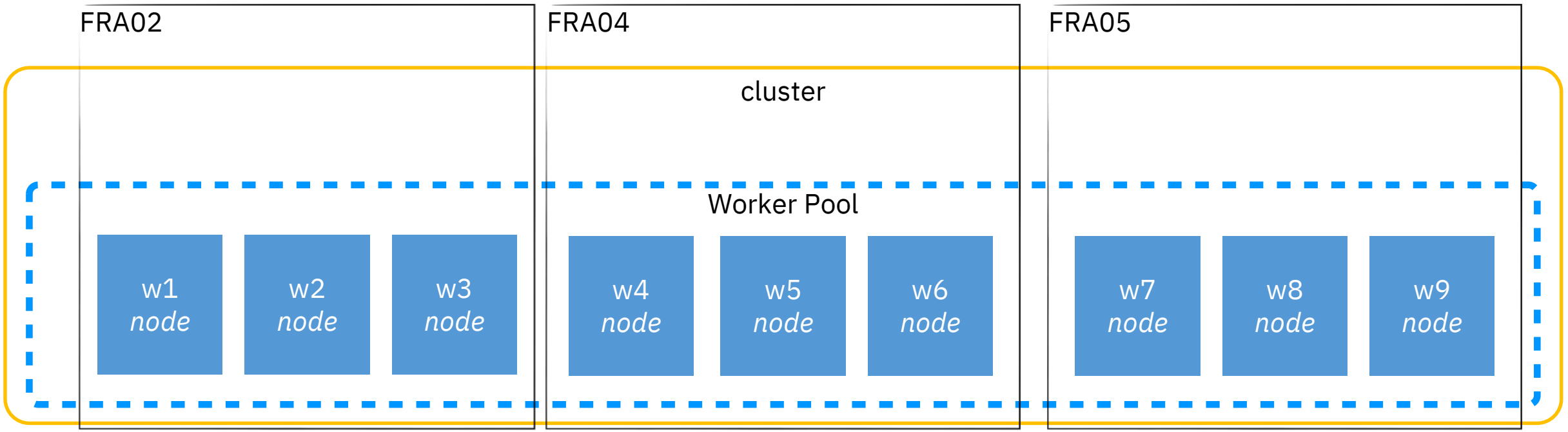
Enables customers to achieve higher availability of their services without an increased operational burden of having to run multiple clusters in separate zones.

https://console.bluemix.net/docs/containers/cs_internal_multi_az.html#cs_multi_az

Multizone Cluster

- Worker nodes are automatically provisioned in the other zones
- Three zones at 150% provides 100% capacity in event of a zone failure.
- Note, 200% capacity required if using two zones

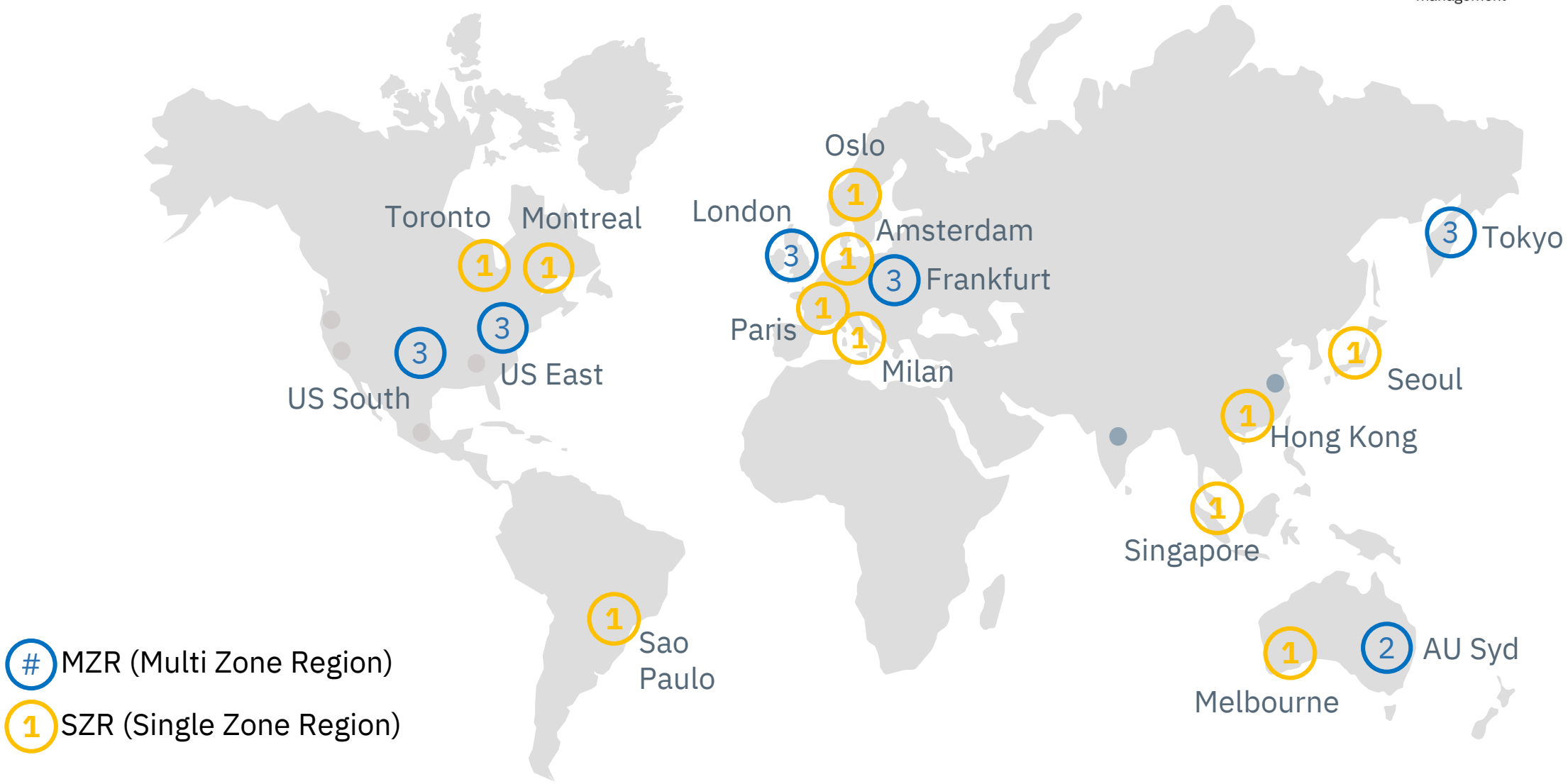
EU-DE



Where are Multizone Clusters available?

Simplified cluster management

Flexible cluster topologies



(#) MZR (Multi Zone Region)

(1) SZR (Single Zone Region)

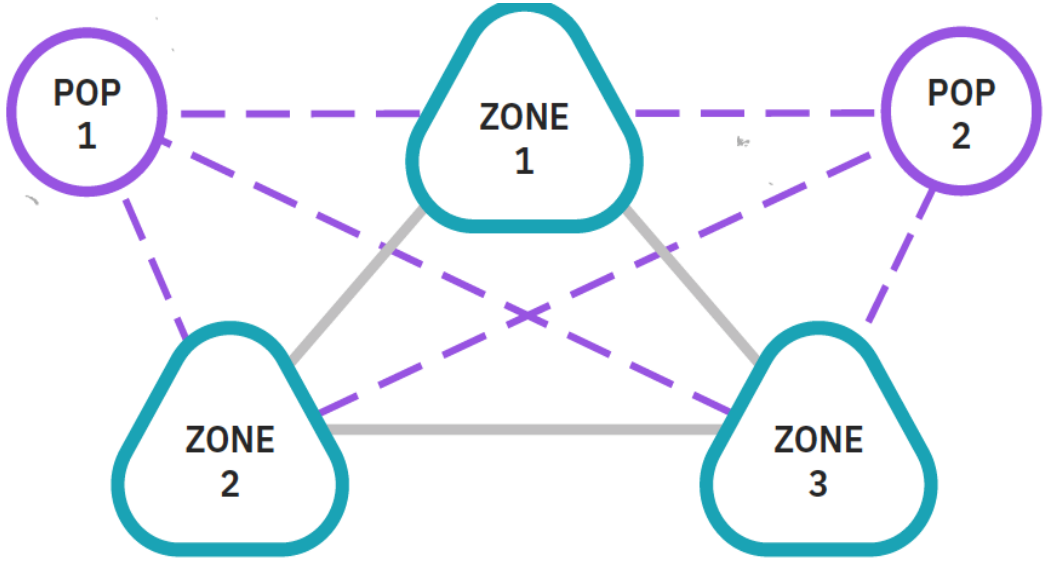
IBM Cloud / © 2018 IBM Confidential

Select the Location where to provision your cluster

Region	Location	City
AP North	hkg02, seo01, sng01, tok02	Hong Kong S.A.R. of the PRC, Seoul, Singapore, Tokyo
AP South	syd01, syd04 mel01	Sydney Melbourne
EU Central	fra02, fra04, fra05 ams03, par01	Frankfurt Amsterdam, Paris
UK South	lon02, lon04, lon06	London
US East	wdc04, wdc06, wdc07 mon01, tor01	Washington DC Montreal, Toronto,
US South	dal10, dal12, dal13 sao01	Dallas São Paulo

High Availability with Multi Zone Region (MZR)

- 3 AZ (Availability Zones) to separate failure domains per MZR (multi-zone region)
- Each AZ is a separate physical data center building.
- Data centers have high bandwidth, low latency redundant links with dual POPs.
- Zones built less than 2msec fiber distance from one another in a region.



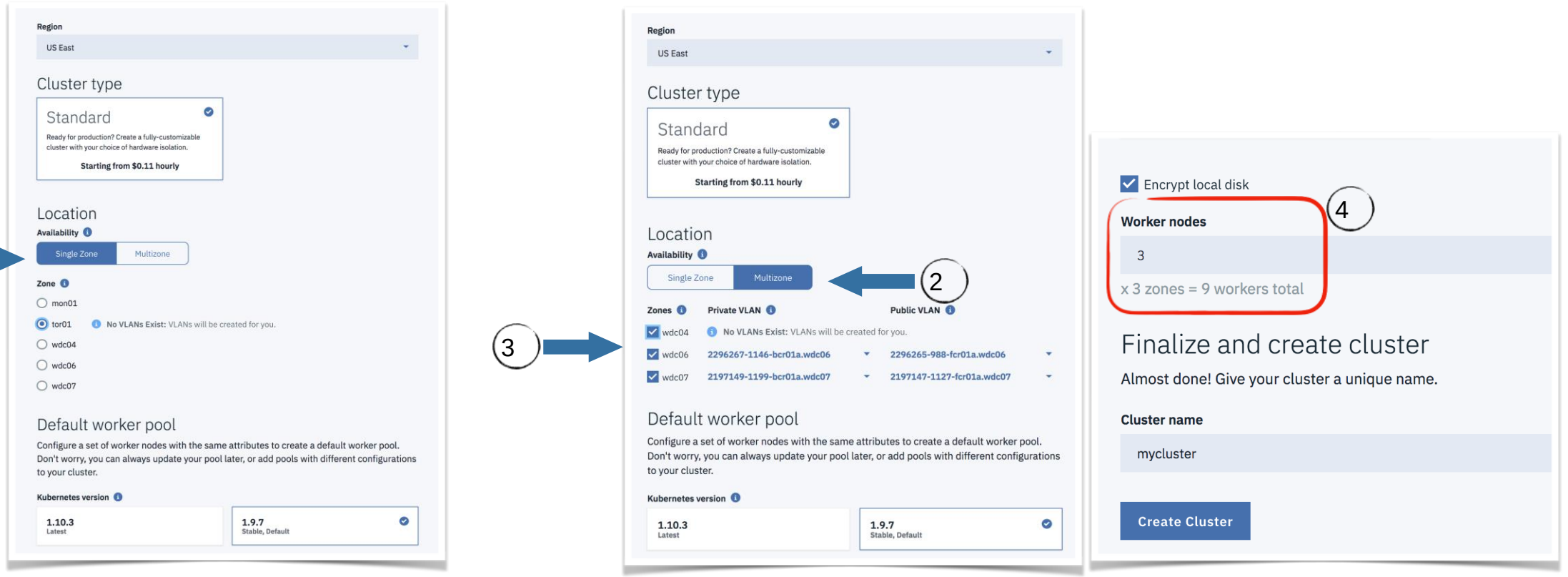
Create Cluster

Simplified cluster management

Flexible cluster topologies

Single Zone Cluster

Multizone Cluster



Persistent Data Storage

Option 1: EmptyDir

Pod

Container

Volume: /emptyDir

Disk space on worker node

Option 2: NFS based file storage

Pod

Container

Volume:
PersistentVolumeClaim

Persistent Volume

Option 3: IBM Cloud database service

Pod

Container

Volume:
IBM Cloud service secret

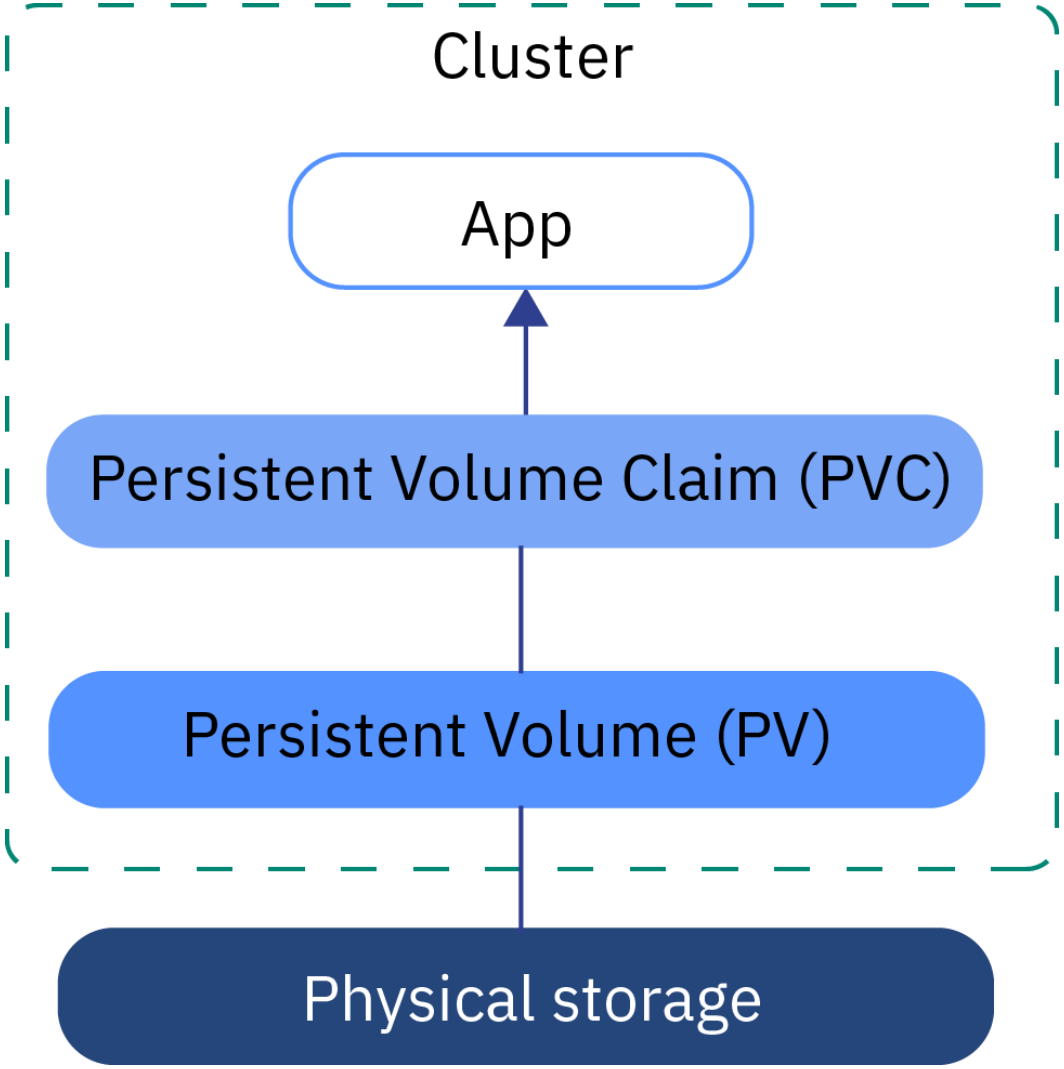
IBM Cloud service connected to
external database

Persistent volumes and persistent volume claims

By default, every cluster is set up with a plug-in to provision file storage.

You can choose to install other add-ons, such as the one for block storage.

To use storage in a cluster, you must create a persistent volume claim, a persistent volume and a physical storage instance.



Option 2 – Storage Classes – NFS 2, 4, 10 IOPS

Developer

kubectl get storageclasses

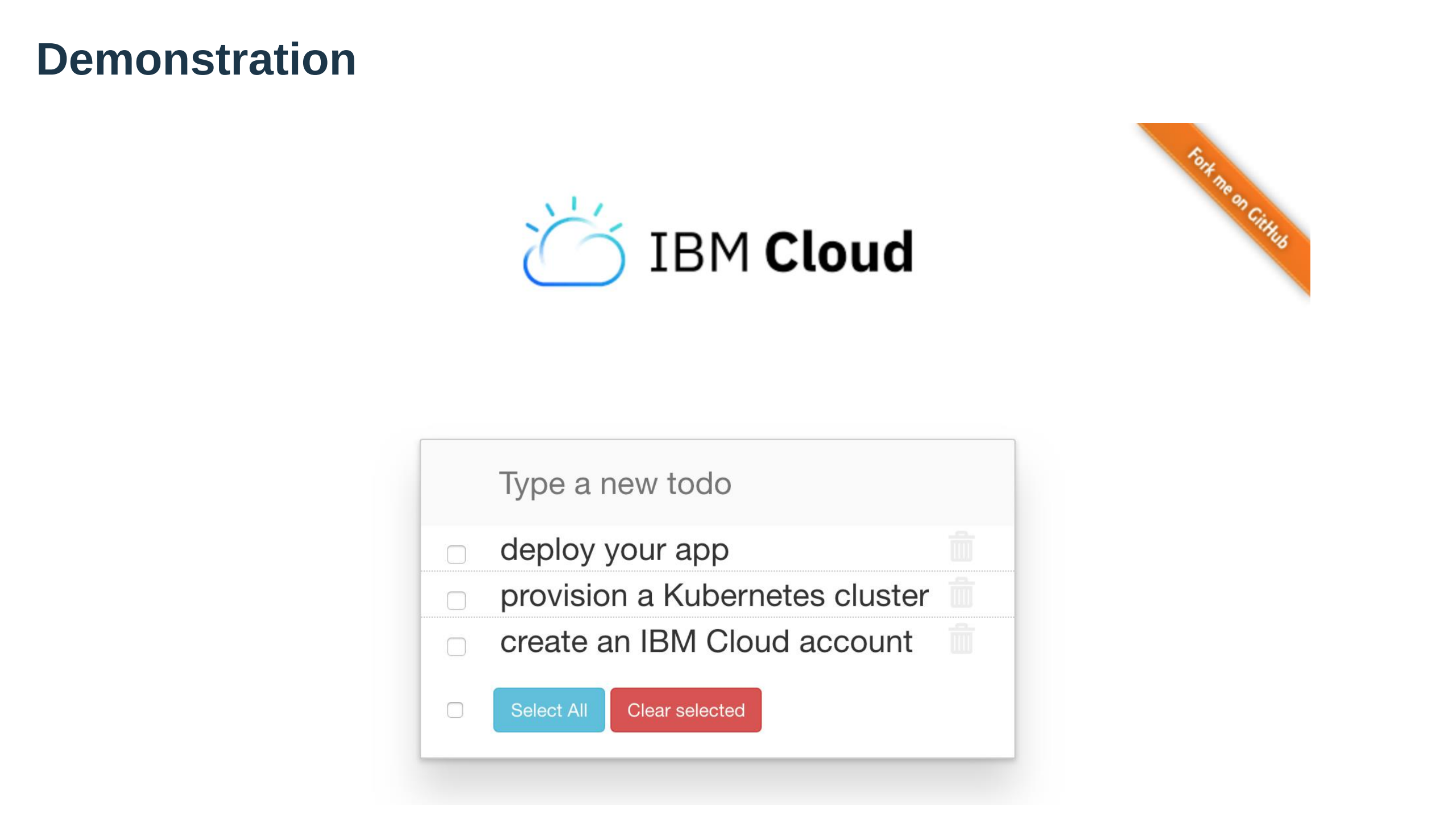
```
NAME                          PROVISIONER
default                       ibm.io/ibmc-file
ibmc-file-bronze              ibm.io/ibmc-file
ibmc-file-custom              ibm.io/ibmc-file
ibmc-file-gold                ibm.io/ibmc-file
ibmc-file-retain-bronze       ibm.io/ibmc-file
ibmc-file-retain-custom       ibm.io/ibmc-file
ibmc-file-retain-gold         ibm.io/ibmc-file
ibmc-file-retain-silver       ibm.io/ibmc-file
ibmc-file-silver              ibm.io/ibmc-file
```

Select Local Disk Encryption

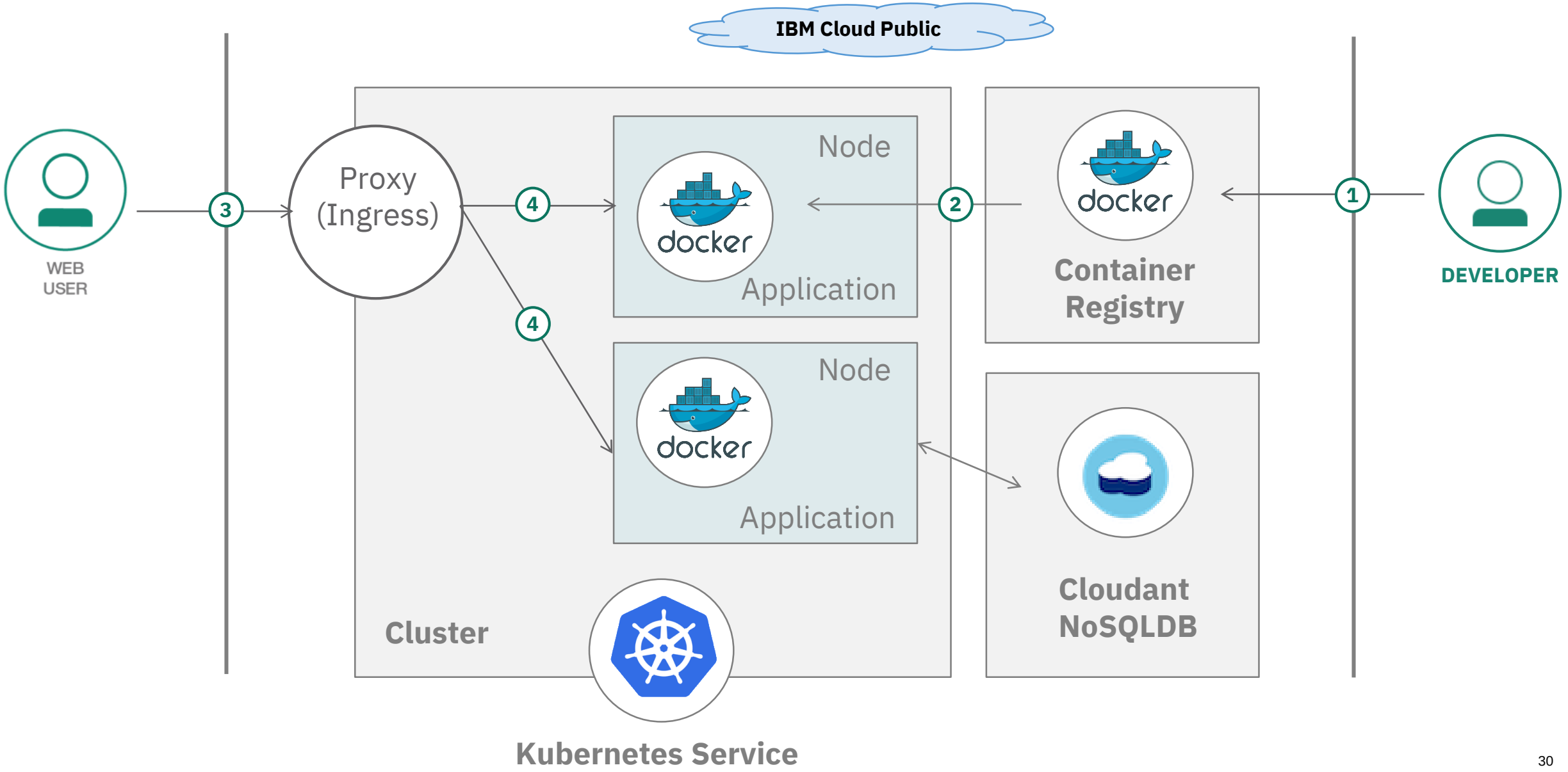
Container security
& isolation

- Encryption by default
- /var/lib/docker is unlocked using LUKS encryption keys
- Each worker node in each cluster has its own unique LUKS encryption key

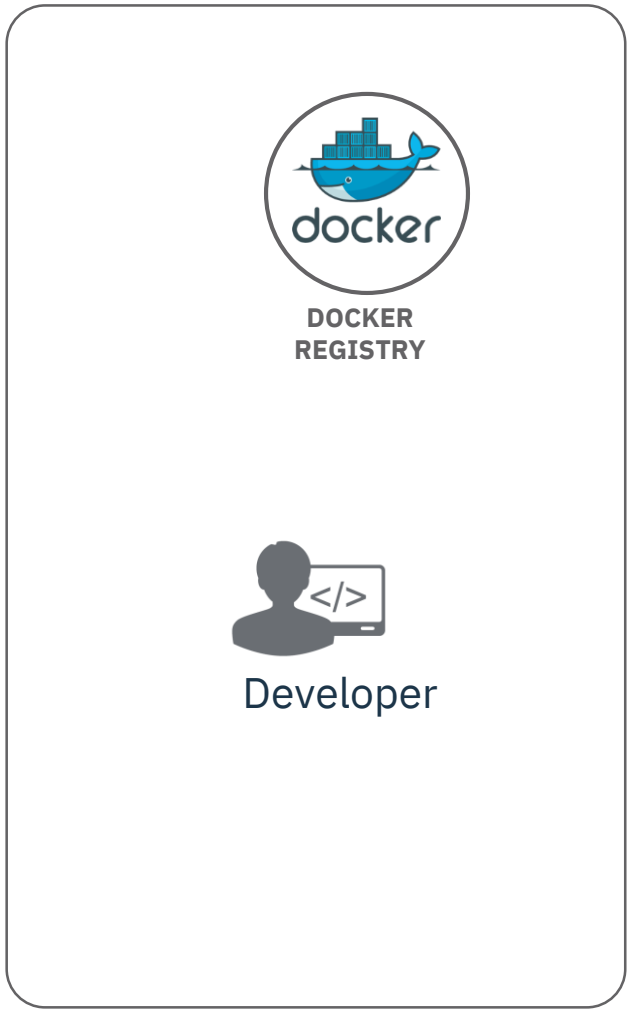
Demonstration



Demonstration



Command Lines



ibmcloud login

ibmcloud target –r eu-de

ibmcloud cr images

docker push <my-image>

ibmcloud target –g <rg-name>

ibmcloud ks

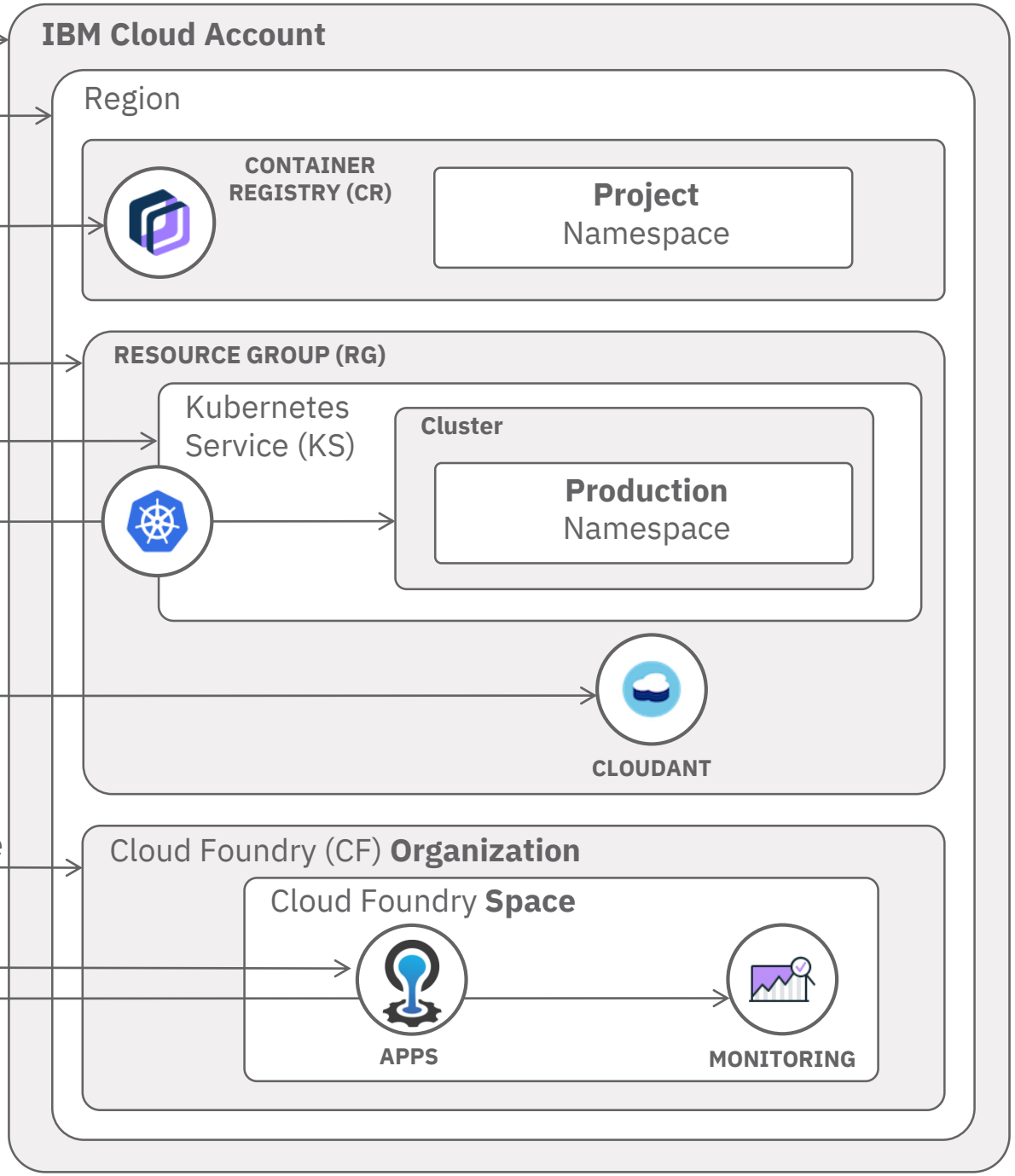
kubectl

ibmcloud resource

ibmcloud target –o org –s space

ibmcloud cf

ibmcloud service



Github repository for Workshop content and Labs instructions: https://ibm.biz/container-ws

Kubernetes Service

Expose your application

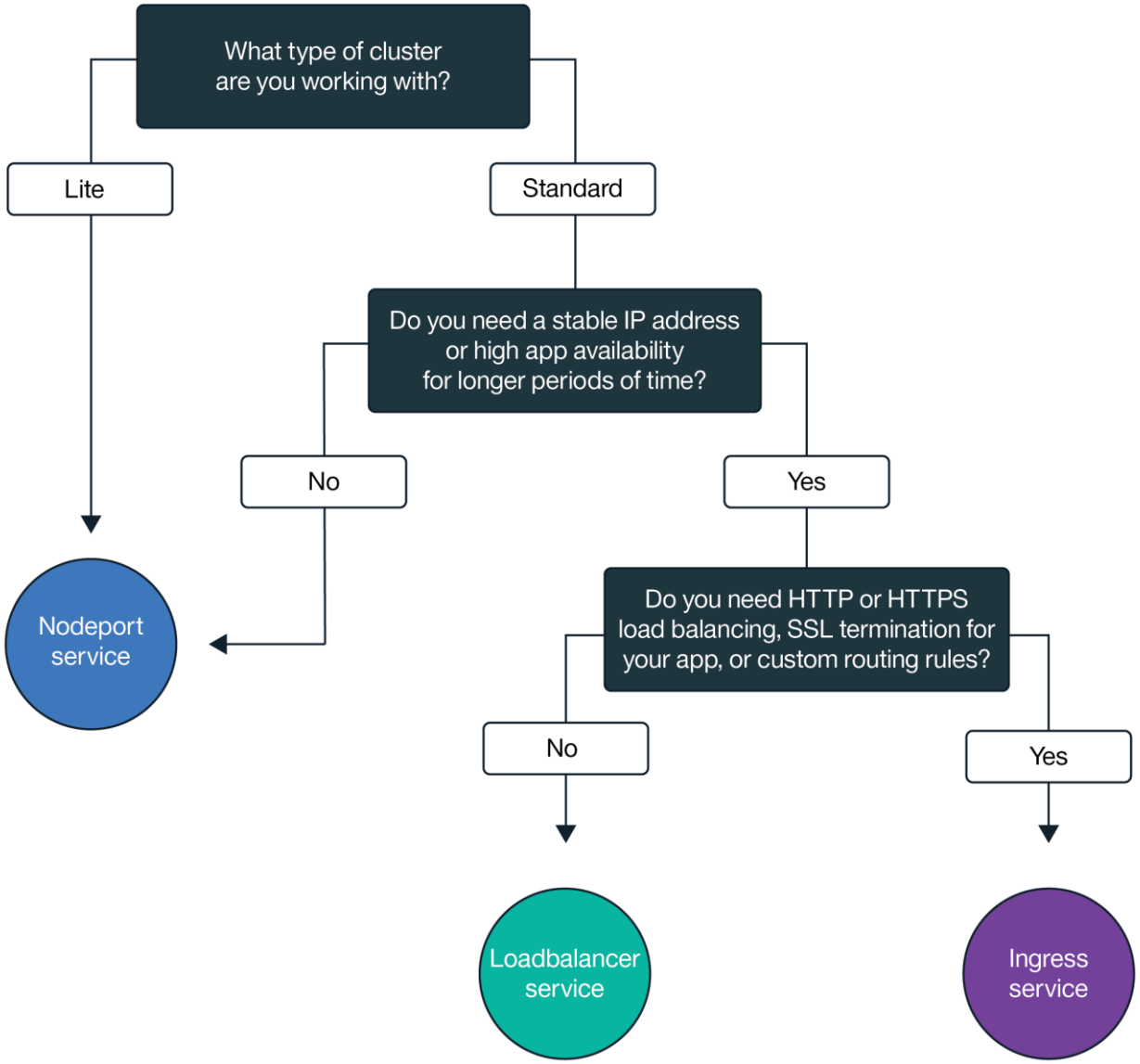
IBM

Kubernetes Service

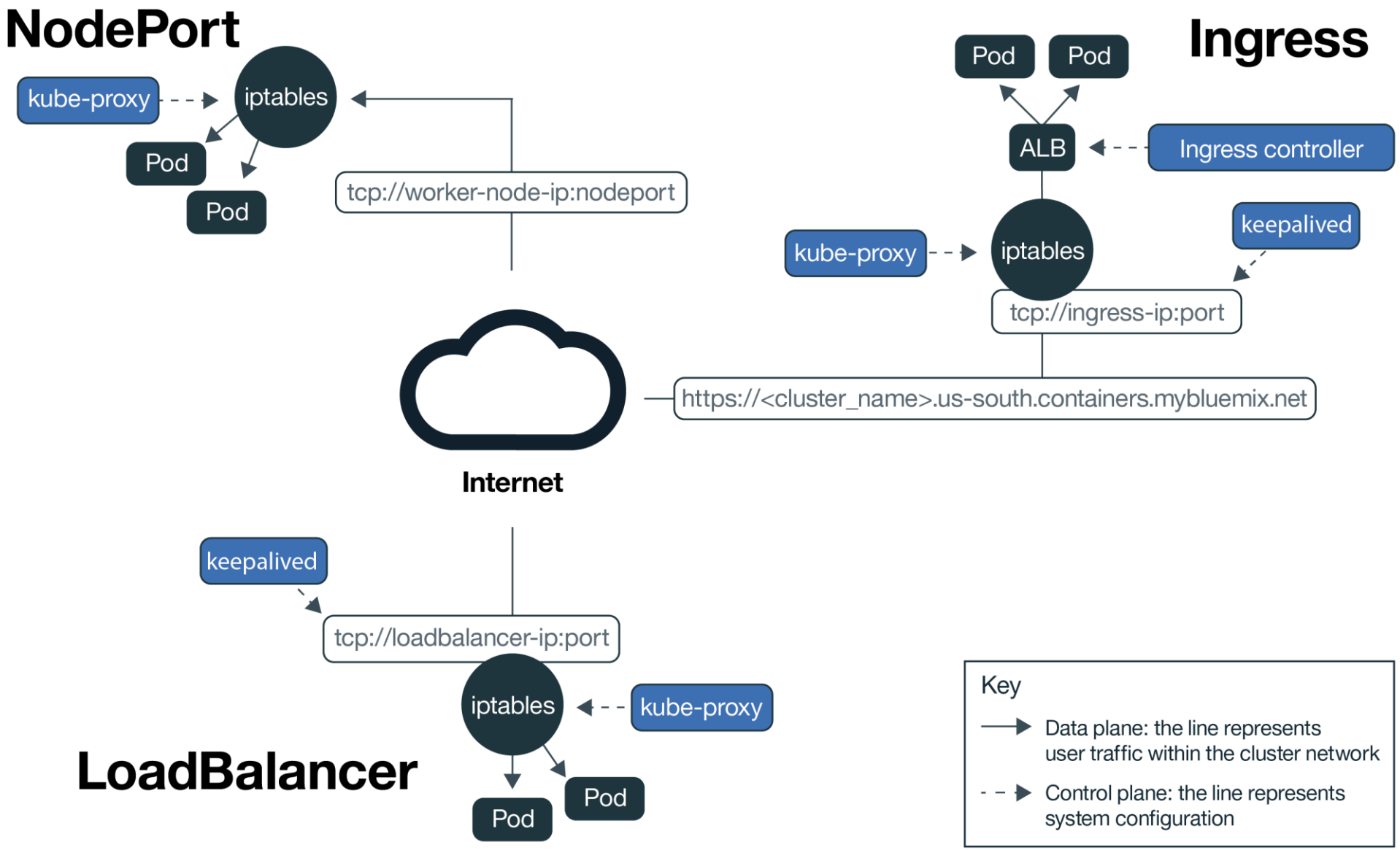
- An abstraction layer which defines a logical set of Pods and enables external traffic exposure, load balancing and service discovery for those Pods.
- Services enable a loose coupling between dependent Pods.
- Although Pods each have a unique IP address, those IPs are not exposed outside the cluster without a Service.
- Services match a set of Pods using labels and selectors, a grouping primitive that allows logical operation on objects in Kubernetes.

```
apiVersion: v1
kind: Service
metadata:
  name: mytodos
  labels:
    app: mytodos
    tier: frontend
spec:
  ports:
  - protocol: TCP
    port: 8080
  selector:
    app: mytodos
    tier: frontend
```

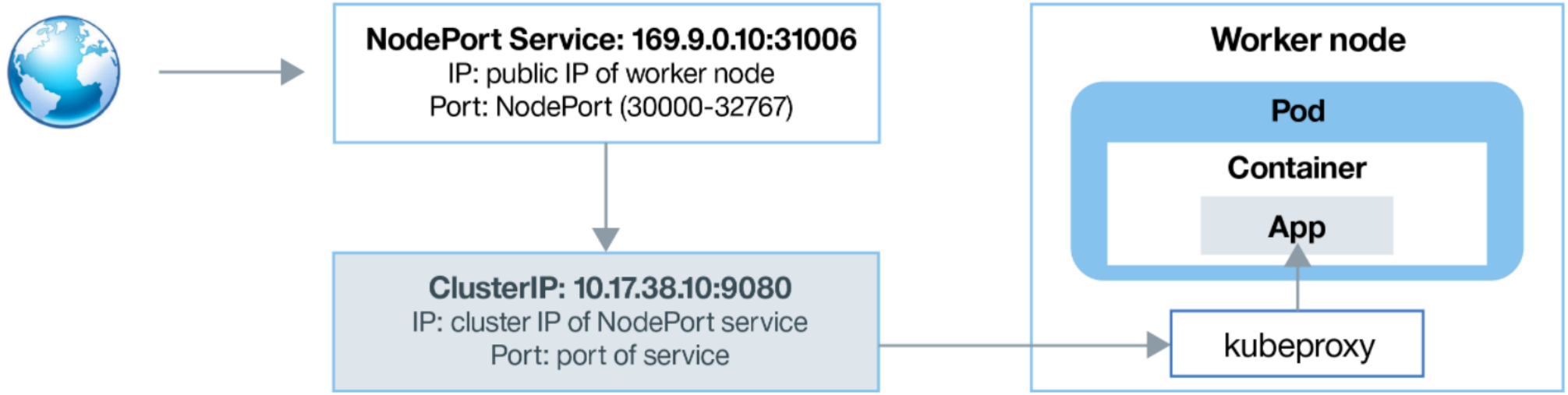
Choose the best networking option to expose service



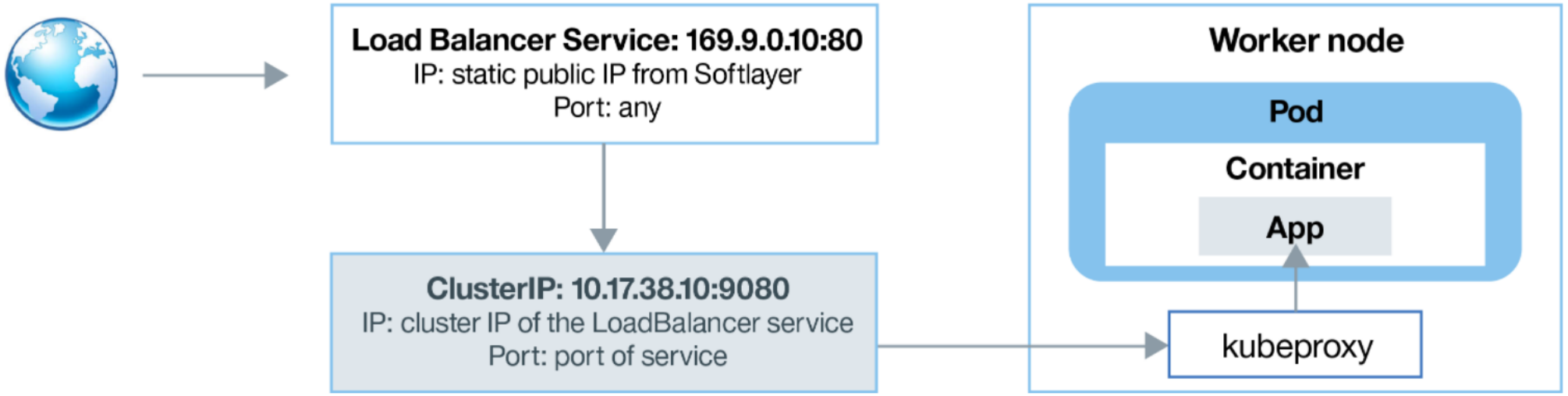
Allowing public access to apps



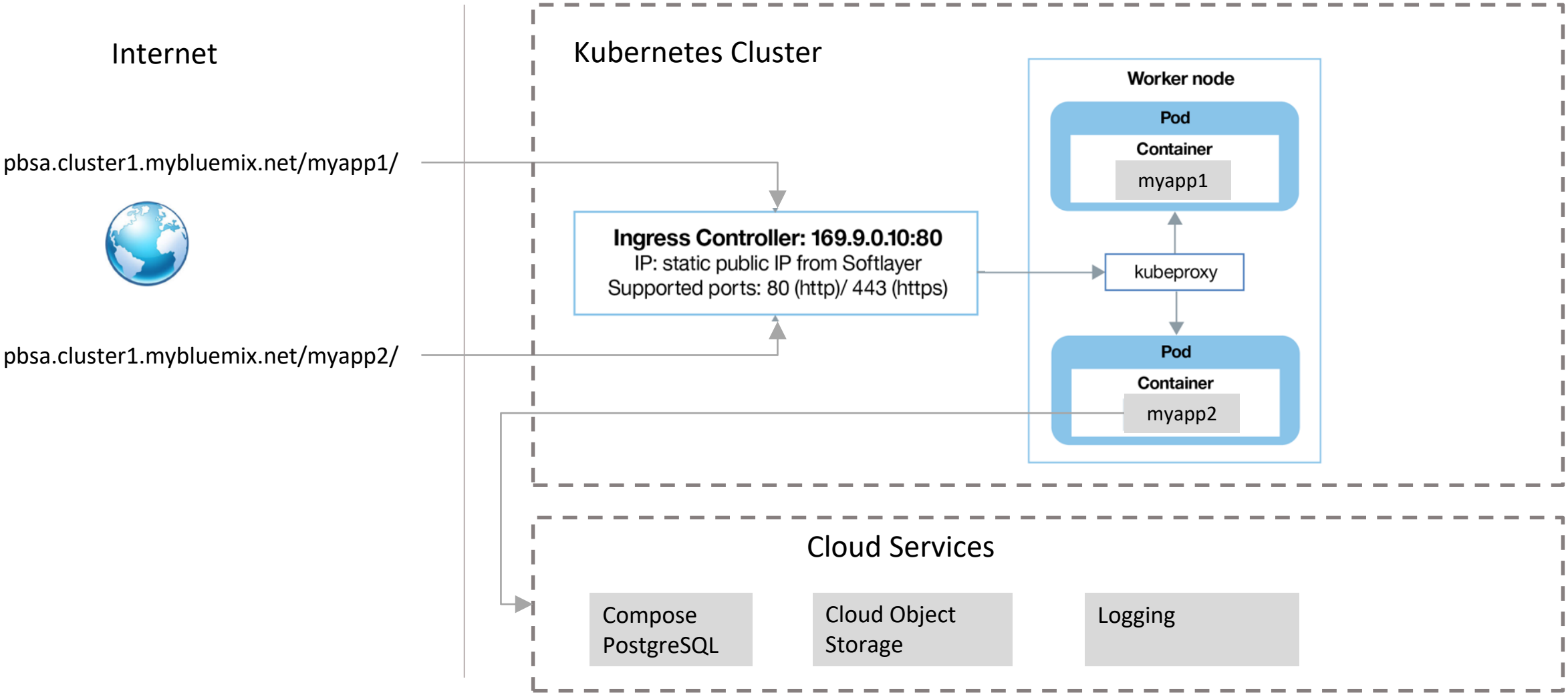
Access the app using NodePort



Access the app using LoadBalancer



Ingress Controller with SSL Termination



Benefits of Ingress Controller

- Special **LoadBalancer-type** service automatically deployed with the cluster.
- Use the reserved public IP, making it highly available.
- Automatically register a **unique public DNS entry with CA signed certificate** that resolves to the public IP address for my Ingress controller, similar to <my-cluster-name>.<region>.containers.mybluemix.net.
- **NGINX-based container deployment** that can be used to expose one or more services to the Internet.
- Certificate stored as a Kubernetes secret in the “default” namespace and can be used to terminate TLS connections for L7 routing.

IBM

Use Ingress Annotations

- Levera Ingress Annotation
- Exemple: Force the use of https if the request is http

```
apiVersion: v1
kind: policy
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
 name: mytodos-ingress
 annotations:
  ingress.bluemix.net/redirect-to-https: "True"
spec:
 tls:
 - hosts:
  - <cluster-name>.eu-de.containers.appdomain.cloud
  secretName: <cluster-name>.
 rules:
 - host: <cluster-name>.eu-de.containers.appdomain.cloud
  http:
   paths:
   - path: /
    backend:
     serviceName: mytodos
     servicePort: 8080
```

Kubernetes Service

Hands-on Labs

IBM

Securing Containers

Secure compute hosts

Built-in security and isolation

Hosted secured **Private image Registry**

Private network overlays

Automatic **Vulnerability scanning**

https://console.bluemix.net/docs/services/Registry/registry_trusted_content.html#registry_trustedcontent

Container Image Security Enforcement

ibmcloud-image-enforcement 0.2.2 IBM

Chart Details

Container Image Security Enforcement

This chart installs Container Image Security Enforcement for IBM Cloud Kubernetes Service in your cluster.

Prerequisites

- Kubernetes v1.9+
- Tiller v2.8+

https://console.bluemix.net/containers-kubernetes/solutions/helm-charts/ibm/ibmcloud-image-enforcement

Kubernetes Service

High Availability

Scaling Services

Scaling is accomplished by changing the number of replicas in a Deployment.

Scaling up a Deployment will ensure new Pods are created and scheduled to Nodes with available resources.

Scaling down will reduce the number of Pods to the new desired state. Kubernetes also supports autoscaling of Pods.

Services have an integrated load-balancer that will distribute network traffic to all Pods of an exposed Deployment.

kubectl scale
--replicas=2 deployment/mytodos

```
apiVersion: v1
kind: Deployment
metadata:
  name: mytodos
spec:
  replicas: 2
  template
    metadata:
      labels:
        app: mytodos
        tier: frontend
    spec:
      containers:
      - name: mytodos
        image: registry.eu-
de.bluemix.net/namespace/mytodos:1
        imagePullPolicy: Always
        resources:
          requests:
            cpu: 100m
            memory: 100Mi
```

Horizontal Pod Autoscaler HPA

kubectl apply –f hpa.yaml

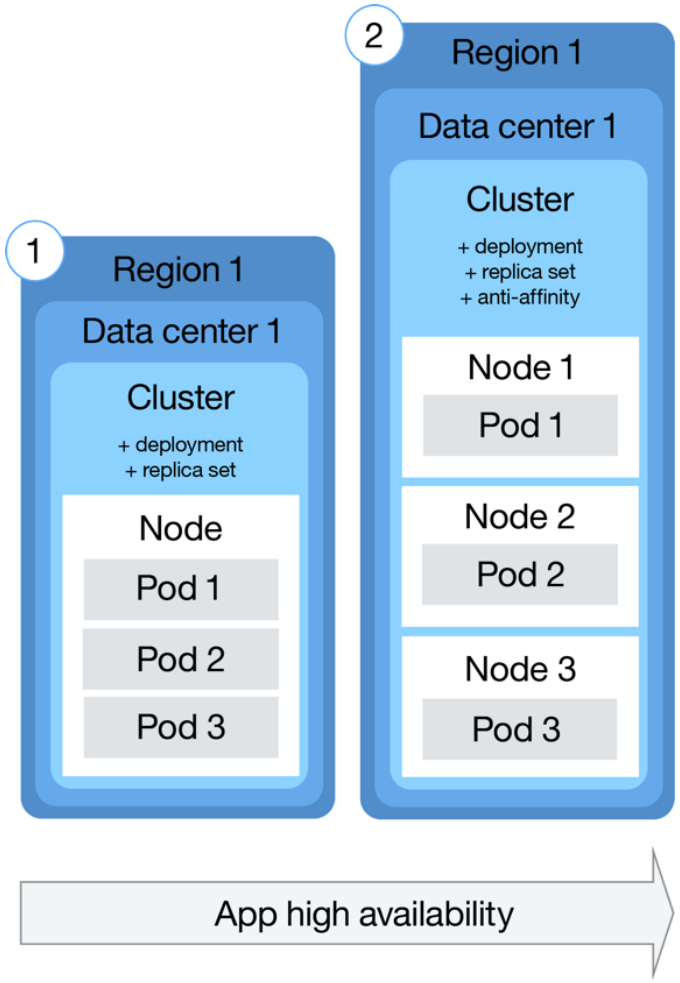
```
apiVersion: autoscaling/v2beta1
kind: HorizontalPodAutoscaler
metadata:
  name: mytodos-scaler
spec:
  scaleTargetRef:
    apiVersion: extensions/v1beta1
    kind: Deployment
    name: mytodos
  minReplicas: 2
  maxReplicas: 10
  metrics:
  - type: Resource
    resource:
      name: cpu
      targetAverageUtilization: 60
  - type: Resource
    resource:
      name: memory
      targetAverageUtilization: 55
```

High Availability

Scale Pods via Replica Sets
Use deployments and replica sets to deploy your app
Include enough replicas for your app's workload

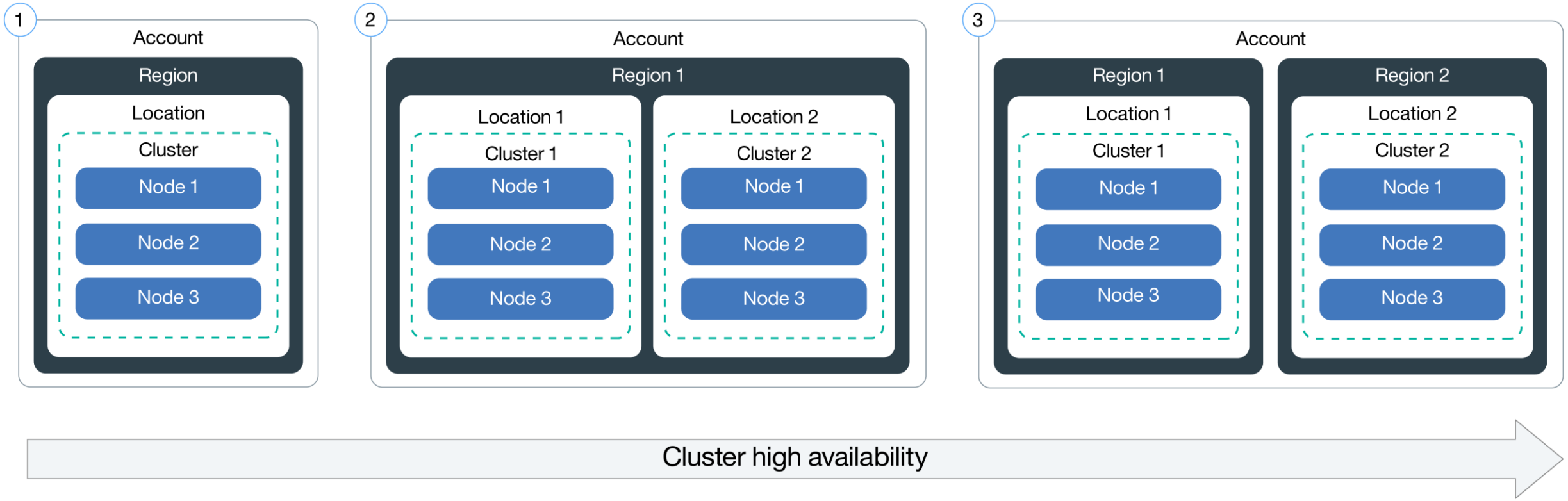
Scale Nodes
Spread pods across multiple nodes (anti-affinity)
Access nodes via external load balancers

Use multiple zone cluster
Implement HA and DR use cases

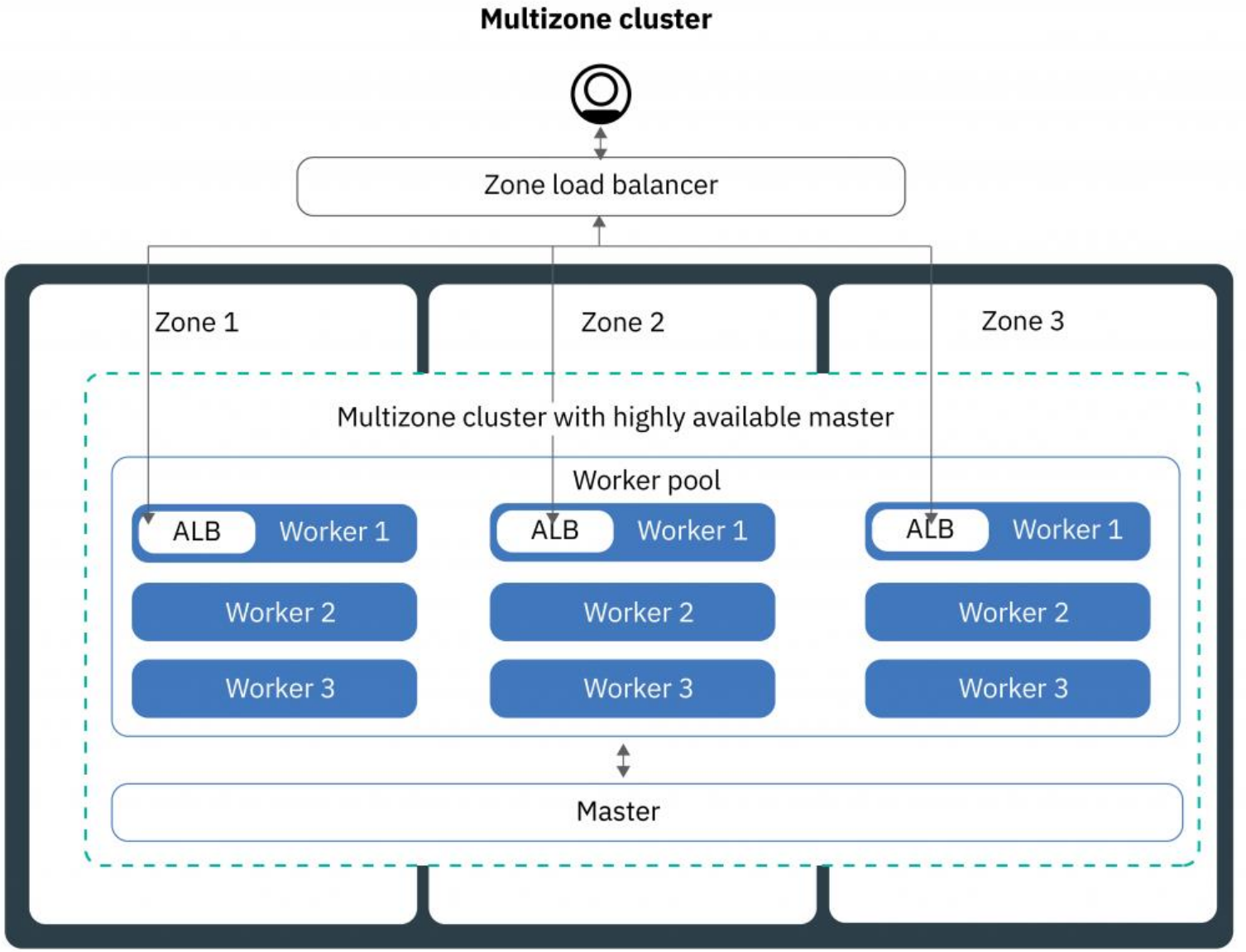


Container Application Resiliency – High Availability Patterns

Design your cluster setup for maximum availability and capacity.



Multi Zone Cluster support in Public Regions

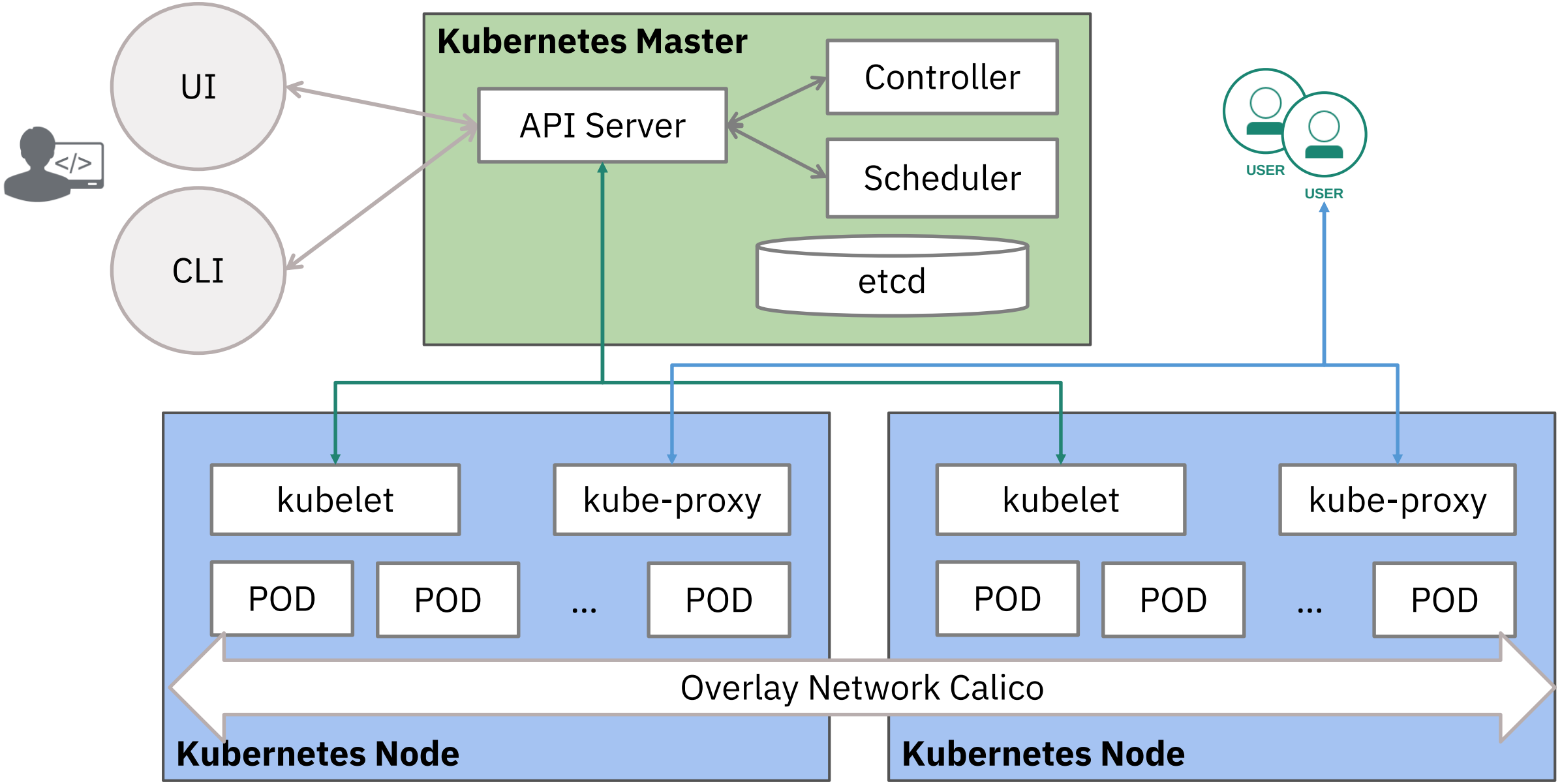


Kubernetes Service

Networking

IBM

Kubernetes Architecture



How Kubernetes Service control networks in Kubernetes?

Uses the open-source Project Calico under the covers to control networks in Kubernetes.

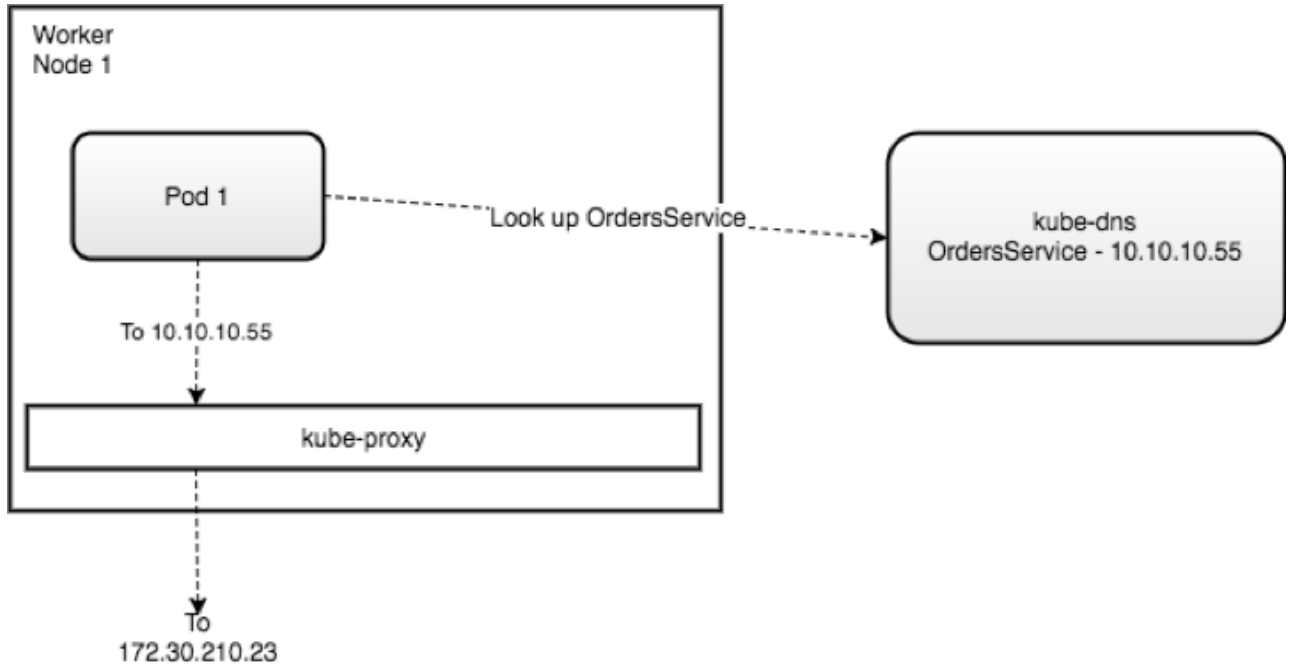
Calico is a Software Defined Networking (SDN) controller that can manage virtual networks across a cluster.

Subnets for container traffic are defined by Kubernetes, and routes for each of the container subnets in the Kubernetes cluster are distributed using Border Gateway Protocol (BGP)

All traffic to and from pods (groups of containers in Kubernetes) is encapsulated using IP-in-IP tunnels, and is routed through the kube-proxy process running on each worker node.

As a developer, it's enough to know that "it just works" and your containers can talk to any other container running in the same cluster over a flat subnet.

Using the Cluster DNS to perform name lookups

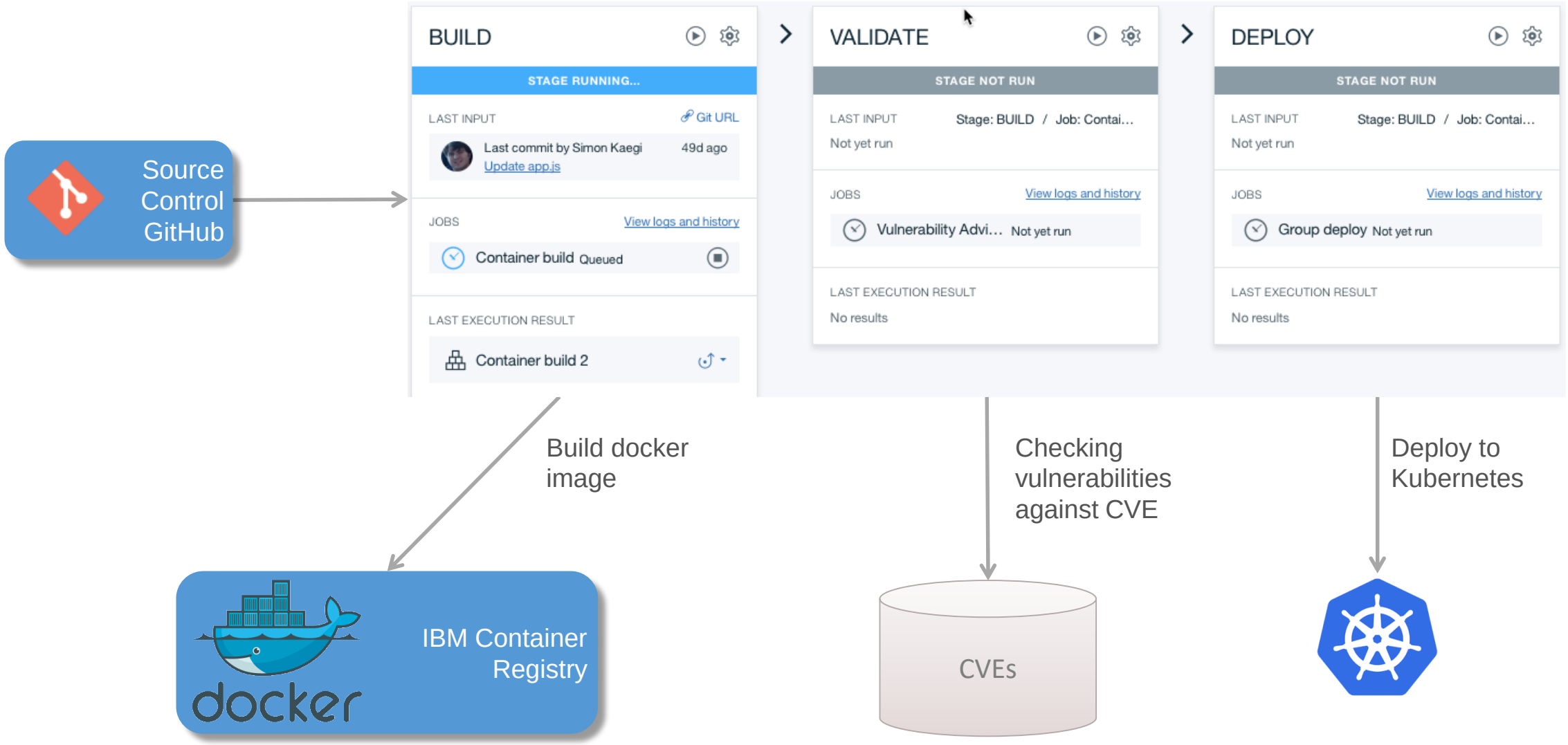


The cluster IP chosen for a service is randomly selected, but the service resource's name-cluster IP pair is registered to the internal kube-dns service, which is an internal DNS system that each pod's DNS resolver configuration points at by default.

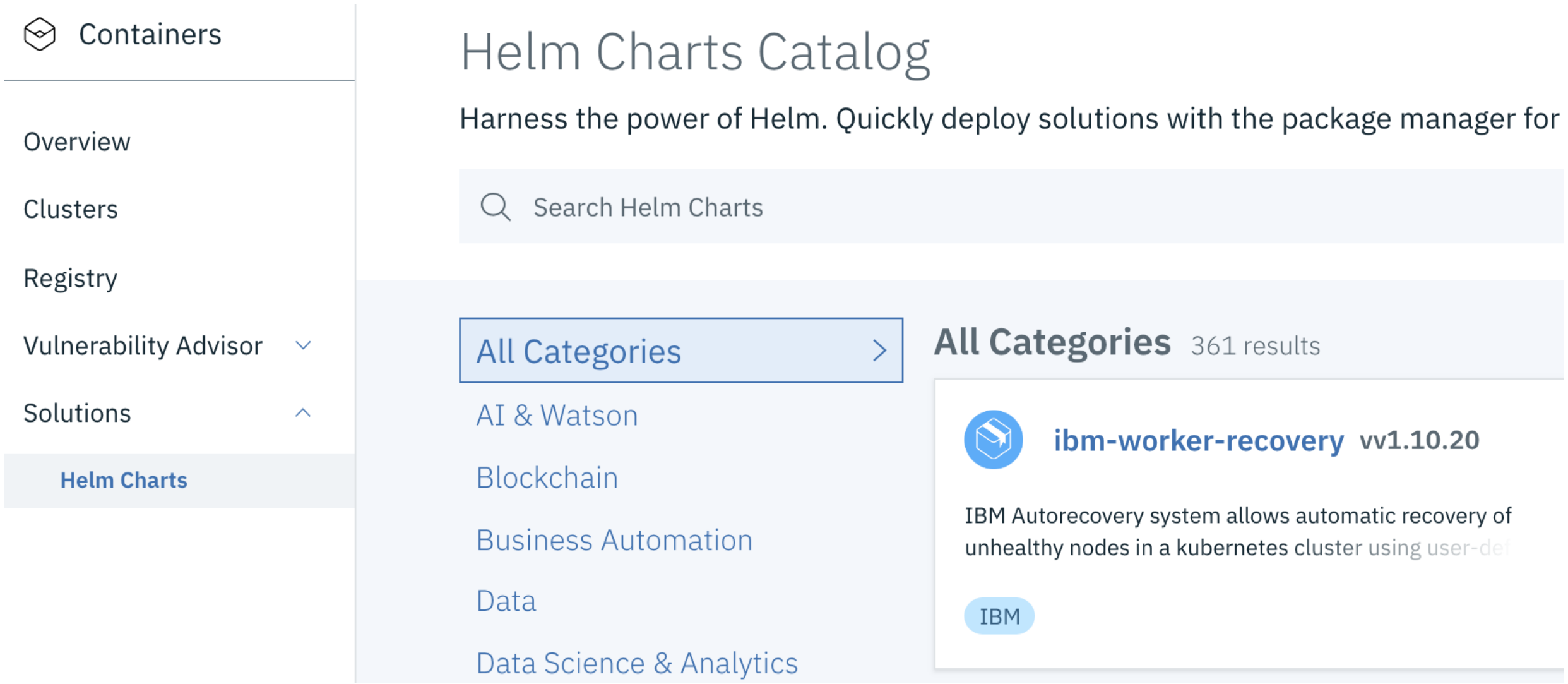
To call a service from any pod in a Kubernetes cluster, it's as simple as resolving the named service to a cluster IP.

Another developer writing a microservice that calls my OrdersService REST API just uses an HTTP client that points at http://orders-service/micro/ordersto get my API.

Open Toolchain with Delivery Pipeline for Kubernetes



Helm Charts Catalog



https://console.bluemix.net/containers-kubernetes/solutions/helm-charts

IBM Cloud / © 2018 IBM Corporation

Great articles

Online Documentation
https://console.bluemix.net/docs/containers/container_index.html

5 Great Articles Kubernetes and Networking
https://www.ibm.com/blogs/bluemix/2017/05/kubernetes-and-bluemix-container-based-workloads-part1

Securing Containers in IKS
https://developer.ibm.com/dwblog/2018/securing-containers-iks-kubernetes/
https://www.ibm.com/blogs/bluemix/2018/06/pod-security-policies-ibm-cloud-kubernetes-service/

Deployment Patterns for Maximizing Throughput and Availability
https://www.ibm.com/blogs/bluemix/2018/10/ibm-cloud-kubernetes-service-deployment-patterns-for-maximizing-throughput-and-availability

IBM

Most common ibmcloud ks commands

Command	Description
ibmcloud ks **clusters**	Get the list of clusters
ibmcloud ks **cluster-config** <cluster-name>	
ibmcloud ks **workers** <cluster-name>	Find out the IP of the worker nodes
ibmcloud ks **cluster-get**	check which space your cluster is associated with (for logs and services)
ibmcloud ks **albs** –cluster	Find the private application load balancer ID.
ibmcloud **iam user-policy-create** <email> --roles Administrator --service-name containers-kubernetes	Assign the policy to allow a developer to administrate a cluster

ks = kubernetes service

Most common ibmcloud cr commands

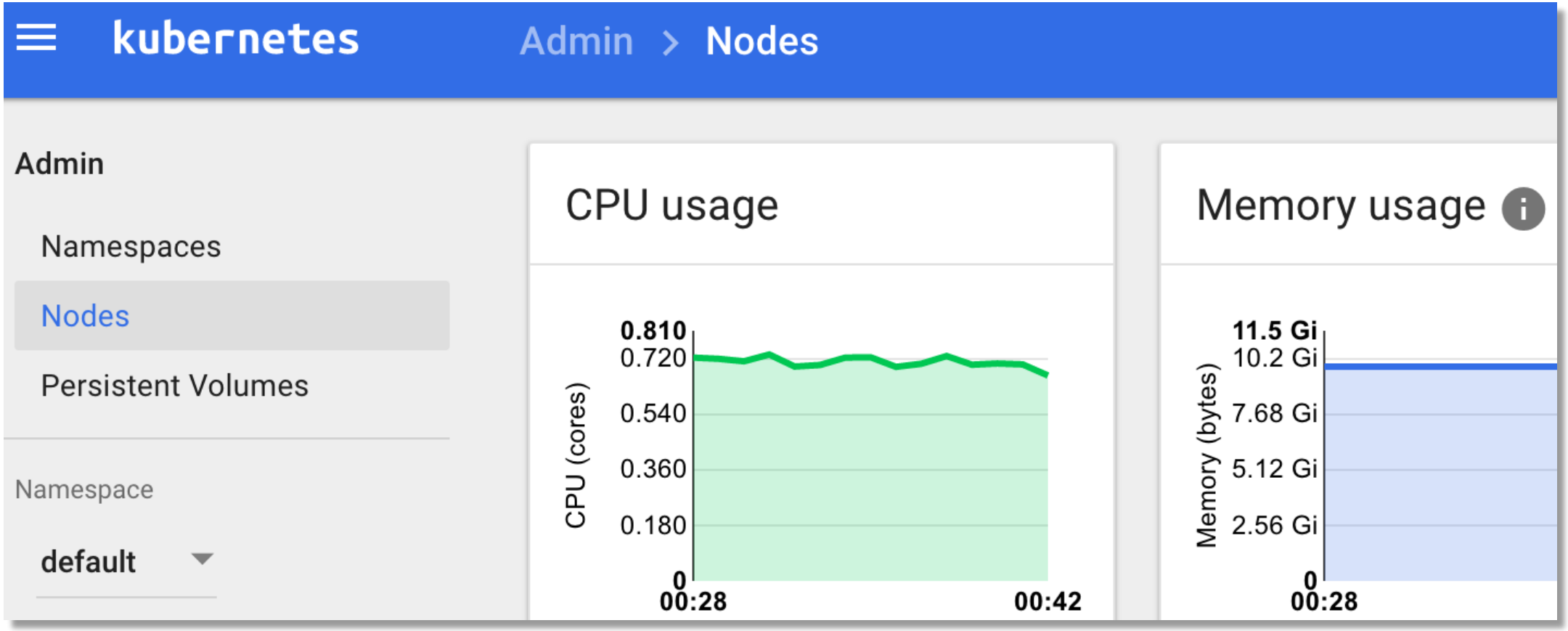
Command	Description
ibmcloud cr **images**	List all images
ibmcloud cr **images** –restrict <namespace>	List of images within a namespace

cr = container registry

Most common kubectl commands

Command	Description
kubectl **cluster-info** <cluster-name>	
kubectl **config** view <cluster-name>	
kubectl **get pods** -n <namespace>	List of pods, number of restarts, age ex: kubectl get pods-n kube-system
kubectl **describe** pod <pod-name>	Get detailed pod info including logs
kubectl **top nodes**	CPU/Memory of all nodes
kubectl **top pods** -n <namespace>	CPU/Memory of pods in namespace
kubectl **describe** node	Show you the allocated memory/usage.
kubectl **logs** -f <pod-name>	Retrieve logs from a pod
kubectl **exec** -it <pod-name> -- /bin/sh	Enter the shell in the pod
kubectl **delete** pods -n kube-system <kube-dns-amd64-pod-name>	Restarting the kube-dns pods
kubetail --selector service=<service-name>	Trace logs for a service

Cluster Monitoring with the Kubernestes Dashboard



Manager Cluster using REST API

clusters

Show/Hide | List Operations | Expand Operations

GET	/v1/clusters	List the clusters that you have access to.
POST	/v1/clusters	Create a cluster.
DELETE	/v1/clusters/{idOrName}	Delete a cluster.
GET	/v1/clusters/{idOrName}	View details for a cluster.
PUT	/v1/clusters/{idOrName}	Update the version of the Kubernetes cluster master node.
GET	/v1/clusters/{idOrName}/config	Download the cluster-specific configuration and certificates.
POST	/v1/clusters/{idOrName}/kms	Create a Key Protect configuration for a cluster.
PUT	/v1/clusters/{idOrName}/masters	Refresh the Kubernetes master.

https://eu-de.containers.bluemix.net/swagger-api/#/clusters