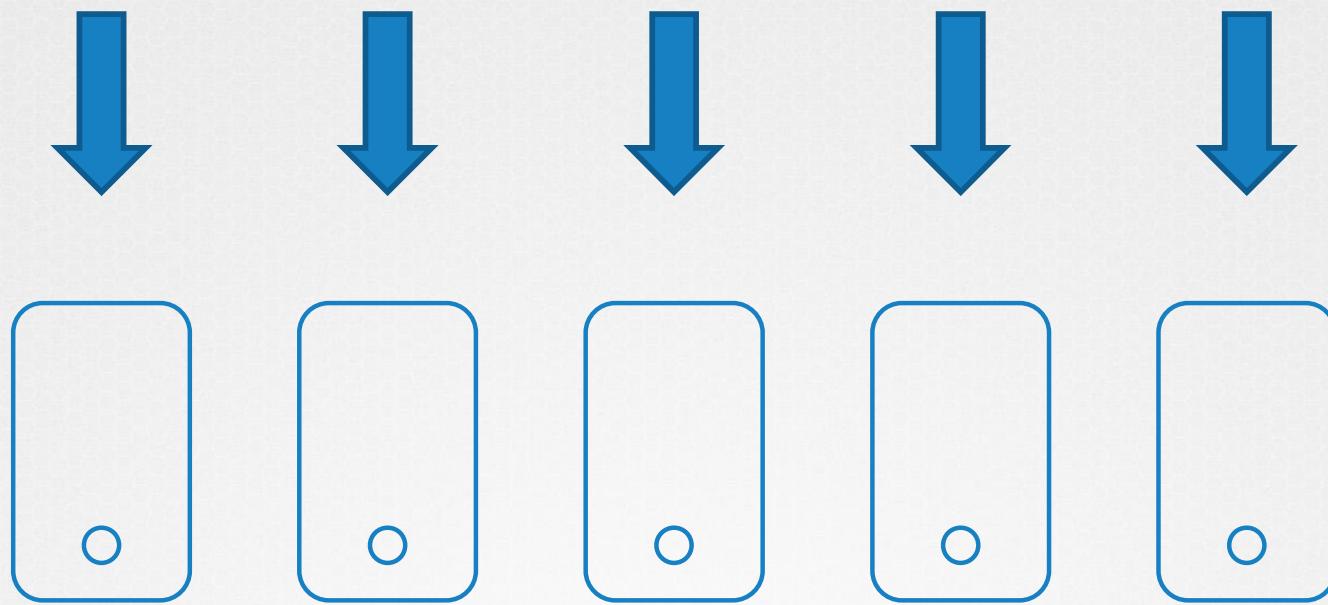




Auto Scaling

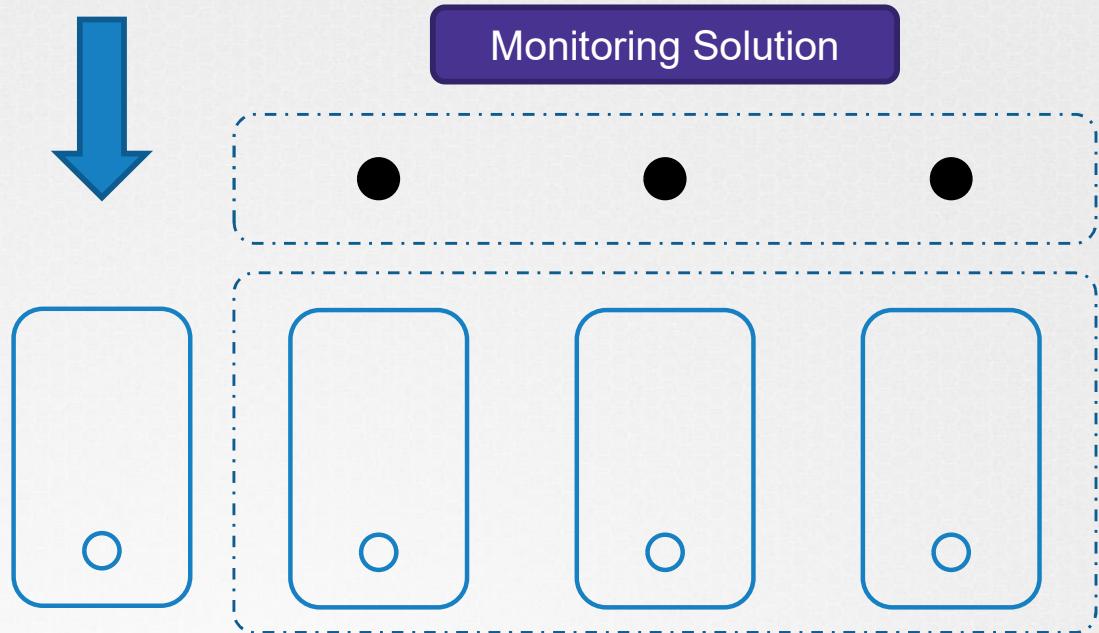


Autoscaling

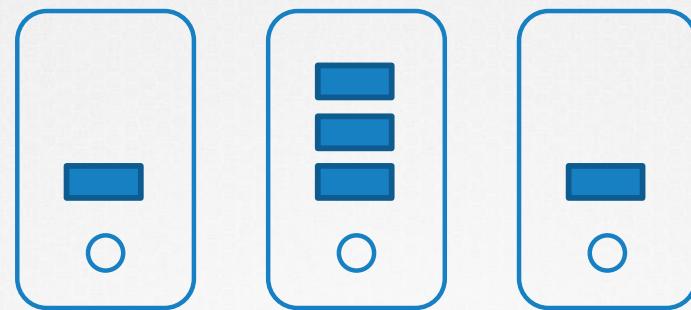


13 Key Factors of Autoscaling

- Designed for Scale
- Automatic
- Bidirectional - Scale up and down



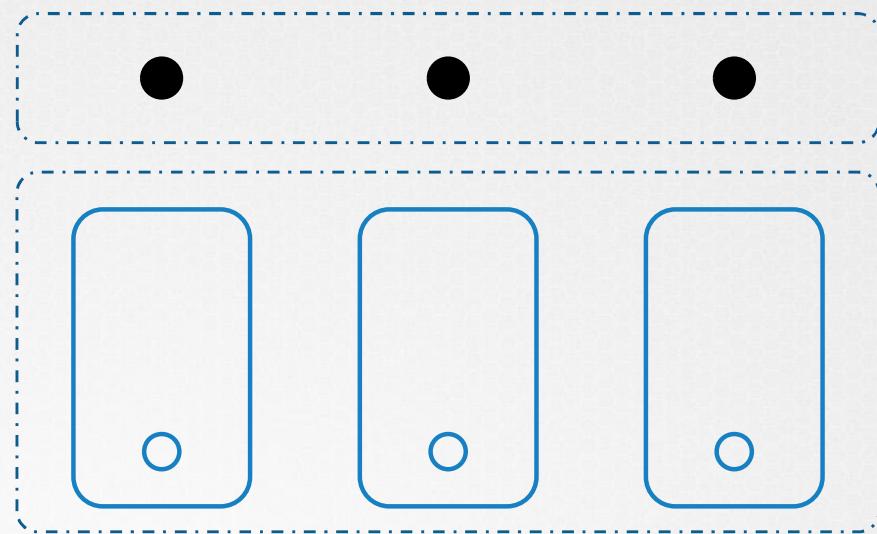
| Horizontal vs Vertical



Vertical
Horizontal

I Kubernetes Auto Scaling

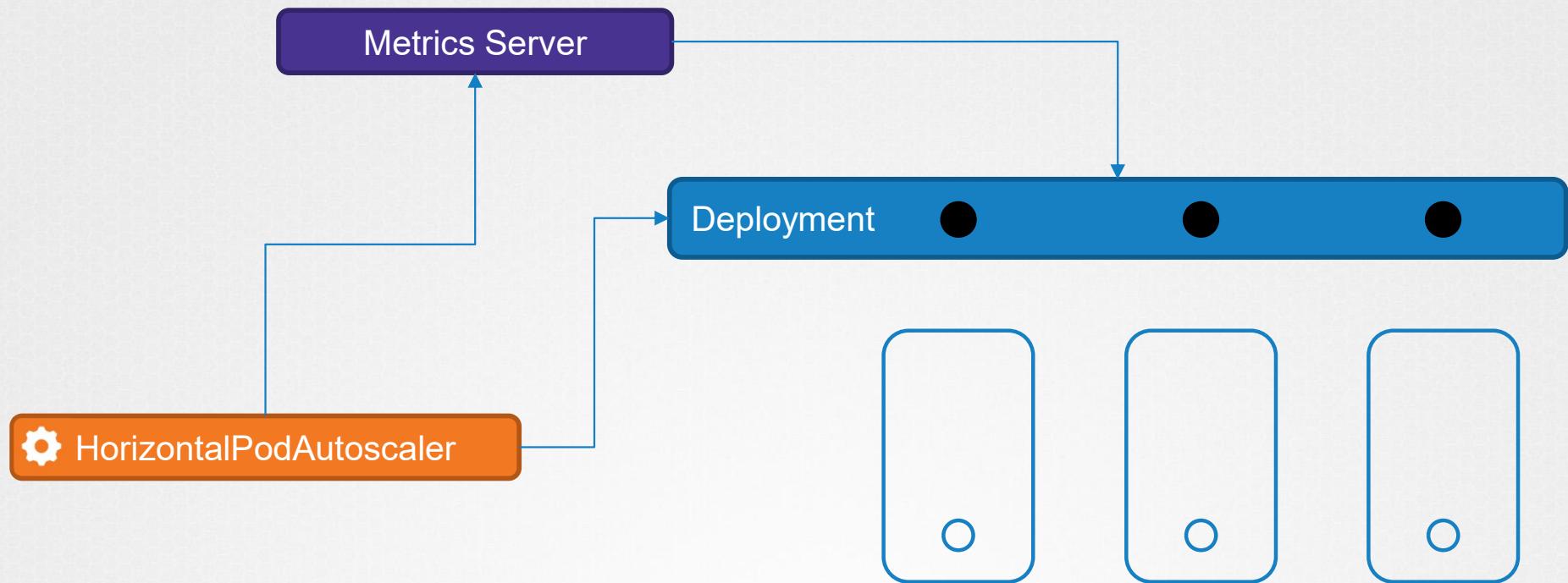
- Horizontal Pod Autoscaler
- Vertical Pod Autoscaler
- Cluster Autoscaler



Horizontal Pod Autoscaler



Horizontal Pod Autoscaler



Horizontal Pod Autoscaler

deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: myapp
spec:
  replicas: 3
  selector:
    matchLabels:
      app: myapp
  template:
    metadata:
      labels:
        app: myapp
    spec:
      containers:
        - name: myapp
          image: myapp:latest
          resources:
            limits:
              cpu: 500m
            requests:
              cpu: 200m
```

hpa.yaml

```
apiVersion: autoscaling/v2beta2
kind: HorizontalPodAutoscaler
metadata:
  name: myapp-hpa
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: myapp
  minReplicas: 1
  maxReplicas: 10
  metrics:
    - type: Resource
      resource:
        name: cpu
      target:
        type: Utilization
        averageUtilization: 50
```

▶ kubectl create -f hpa.yaml

HorizontalPodAutoscaler Created

View Horizontal Pod Autoscaler

```
▶ kubectl get hpa
```

NAME	REFERENCE	TARGETS	MINPODS	MAXPODS	REPLICAS	AGE
myapp-hpa	Deployment/myapp	50%/50%	3	10	3	10m

```
▶ kubectl delete hpa myapp-hpa
```

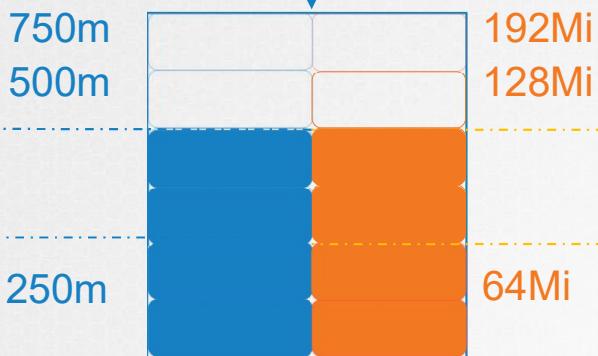


Vertical Pod Autoscaler



Resource Requests & Limits

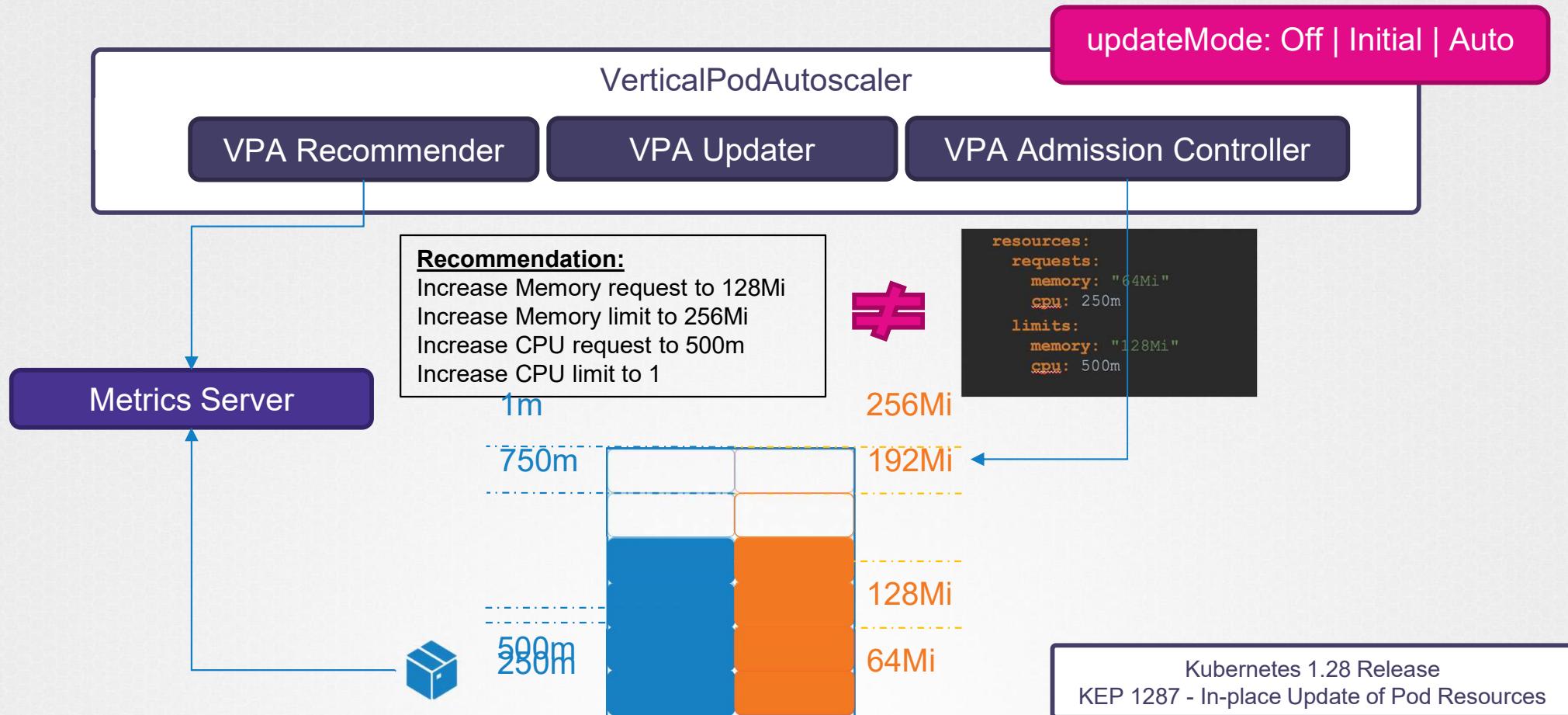
VerticalPodAutoscaler



pod-definition.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: simple-webapp-color
  labels:
    name: simple-webapp-color
spec:
  containers:
  - name: simple-webapp-color
    image: simple-webapp-color
    ports:
    - containerPort: 8080
  resources:
    requests:
      memory: "64Mi"
      cpu: 250m
    limits:
      memory: "128Mi"
      cpu: 500m
```

Vertical Pod Autoscaler



| Deploy Vertical Pod Autoscaler

```
▶ git clone https://github.com/kubernetes/autoscaler.git
```

```
▶ cd autoscaler/vertical-pod-autoscaler/
```

```
▶ ./hack/vpa-up.sh
```

Vertical Pod Autoscaler

pod-definition.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: simple-webapp-color
  labels:
    name: simple-webapp-color
spec:
  containers:
    - name: simple-webapp-color
      image: simple-webapp-color
      ports:
        - containerPort: 8080
      resources:
        requests:
          memory: "64Mi"
          cpu: 250m
        limits:
          memory: "128Mi"
          cpu: 500m
```

vpa-definition.yaml

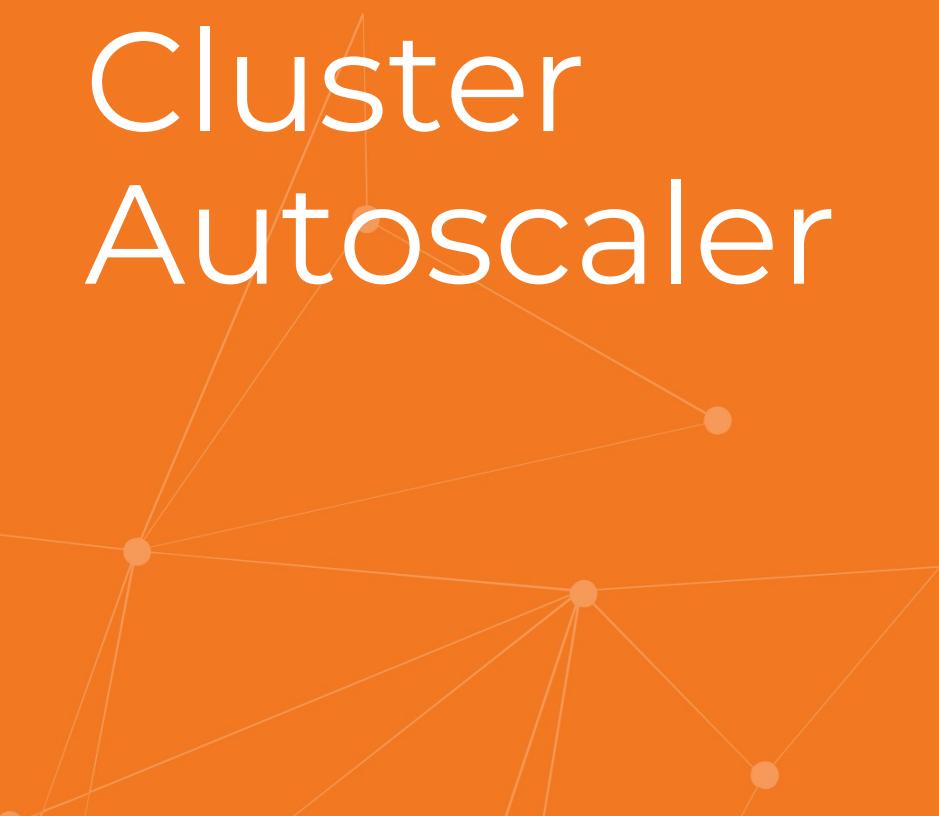
```
apiVersion: "autoscaling.k8s.io/v1"
kind: VerticalPodAutoscaler
metadata:
  name: webapp-vpa
spec:
  targetRef:
    apiVersion: "v1"
    kind: Pod
    name: simple-webapp-color
  updatePolicy:
    updateMode: "Off"
```

Vertical Pod Autoscaler

```
▶ kubectl describe vpa webapp-vpa
```

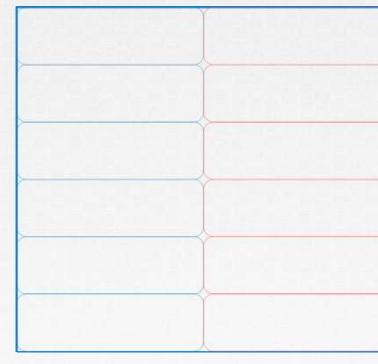
```
Name:          webapp-vpa
Namespace:    default
Labels:        <none>
Annotations:   <none>
API Version:  autoscaling.k8s.io/v1
Kind:         VerticalPodAutoscaler
...
Status:
  Recommendation:
    Container Recommendations:
      Container Name:  webapp
      Lower Bound:
        Cpu:        500m
        Memory:     128Mi
      Target:
        Cpu:        1
        Memory:     256Mi
      Uncapped Target:
        Cpu:        1
        Memory:     256Mi
      Upper Bound:
        Cpu:        2
        Memory:     512Mi
...
...
```

Cluster Autoscaler

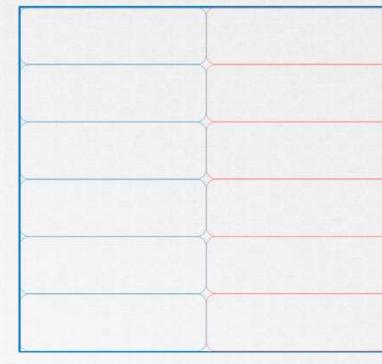




Node 01



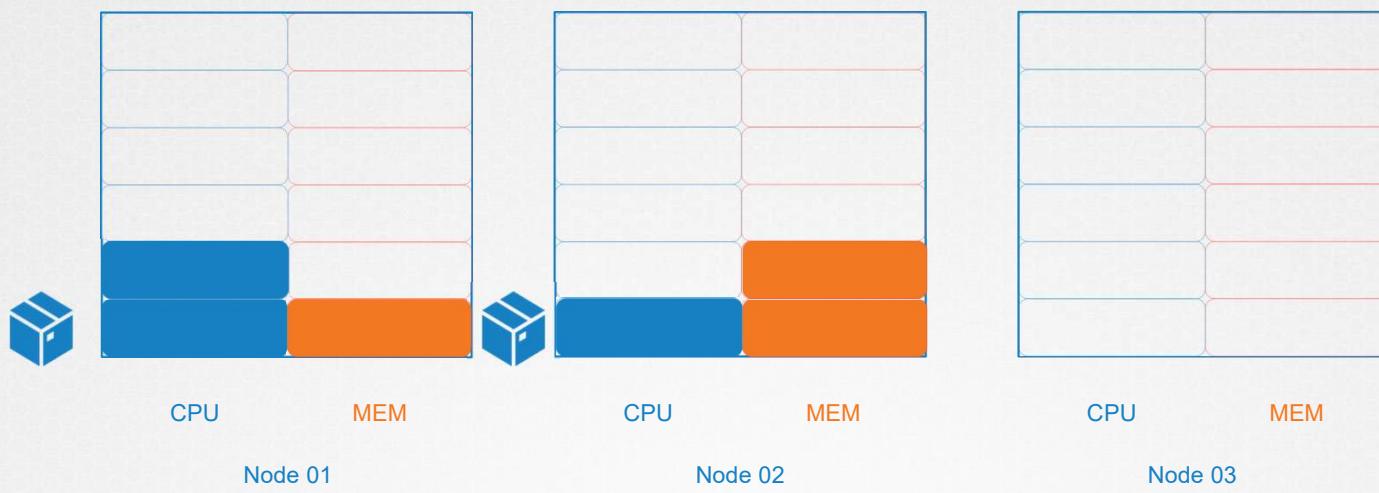
Node 02



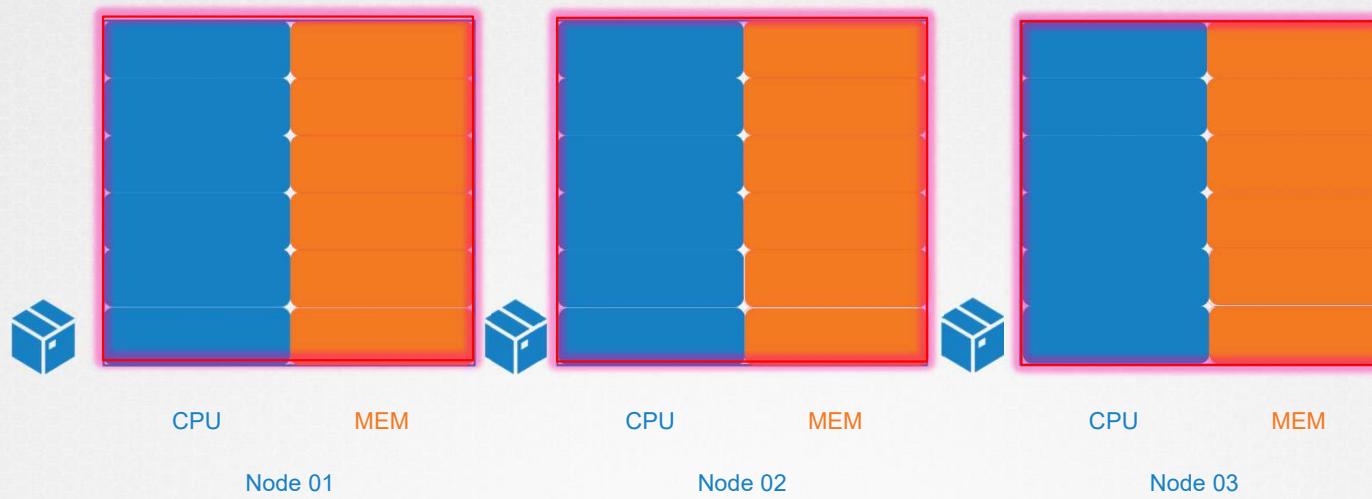
Node 03



kube-scheduler

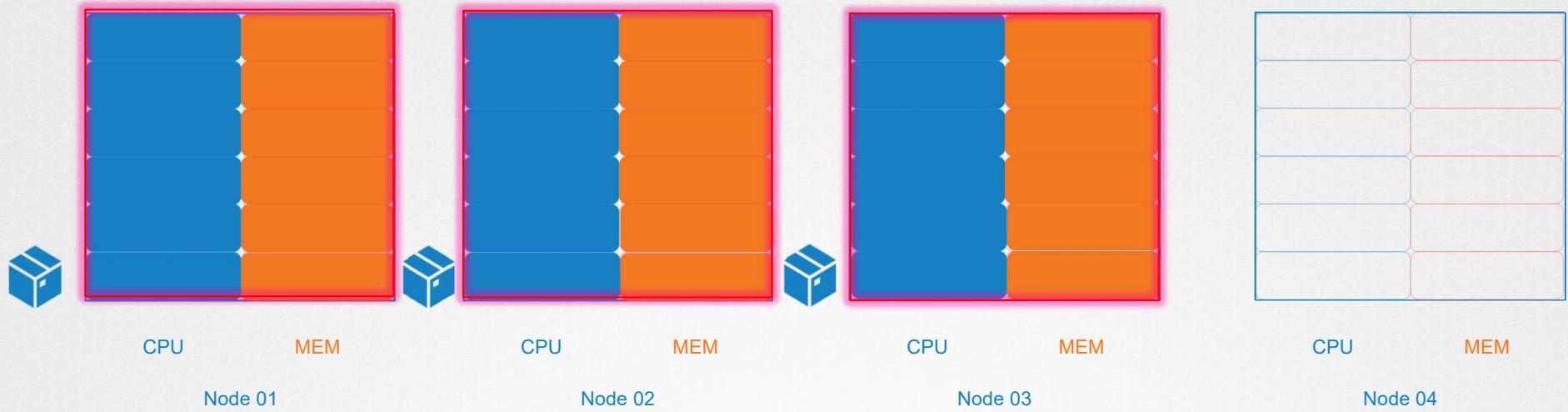


NAME	READY	STATUS	RESTARTS	AGE
Nginx	0/1	Pending	0	7m
Events:				
Reason -----				
FailedScheduling		Message -----		
		No nodes are available that match all of the following predicates:: Insufficient cpu (3).		





kube-scheduler



Cluster Autoscaler

FAQ/Documentation

An FAQ is available [HERE](#).

You should also take a look at the notes and "gotchas" for your specific cloud provider:

- [AliCloud](#)
- [AWS](#)
- [Azure](#)
- [BaiduCloud](#)
- [BizflyCloud](#)
- [Brightbox](#)
- [CherryServers](#)
- [Civo](#)
- [CloudStack](#)
- [ClusterAPI](#)
- [DigitalOcean](#)
- [Exoscale](#)
- [Equinix Metal](#)
- [External gRPC](#)
- [Hetzner](#)
- [HuaweiCloud](#)
- [IonosCloud](#)
- [Kamatera](#)
- [Linode](#)
- [Magnum](#)
- [OracleCloud](#)
- [OVHcloud](#)
- [Rancher](#)
- [Scaleway](#)
- [TencentCloud](#)
- [Vultr](#)

<https://github.com/kubernetes/autoscaler/tree/master/cluster-autoscaler>

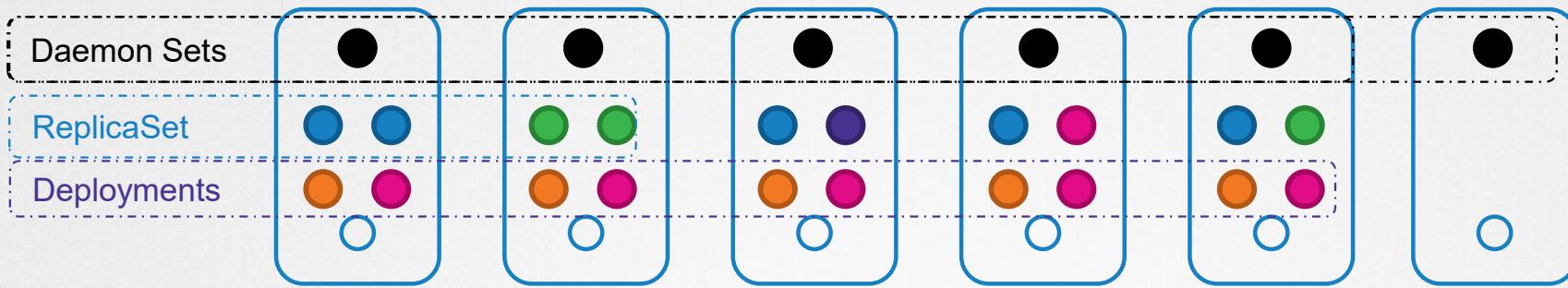
Cluster Autoscaler

```
▶ gcloud container clusters create my-cluster --cluster-autoscaler=min-nodes=3,max-nodes=10
```

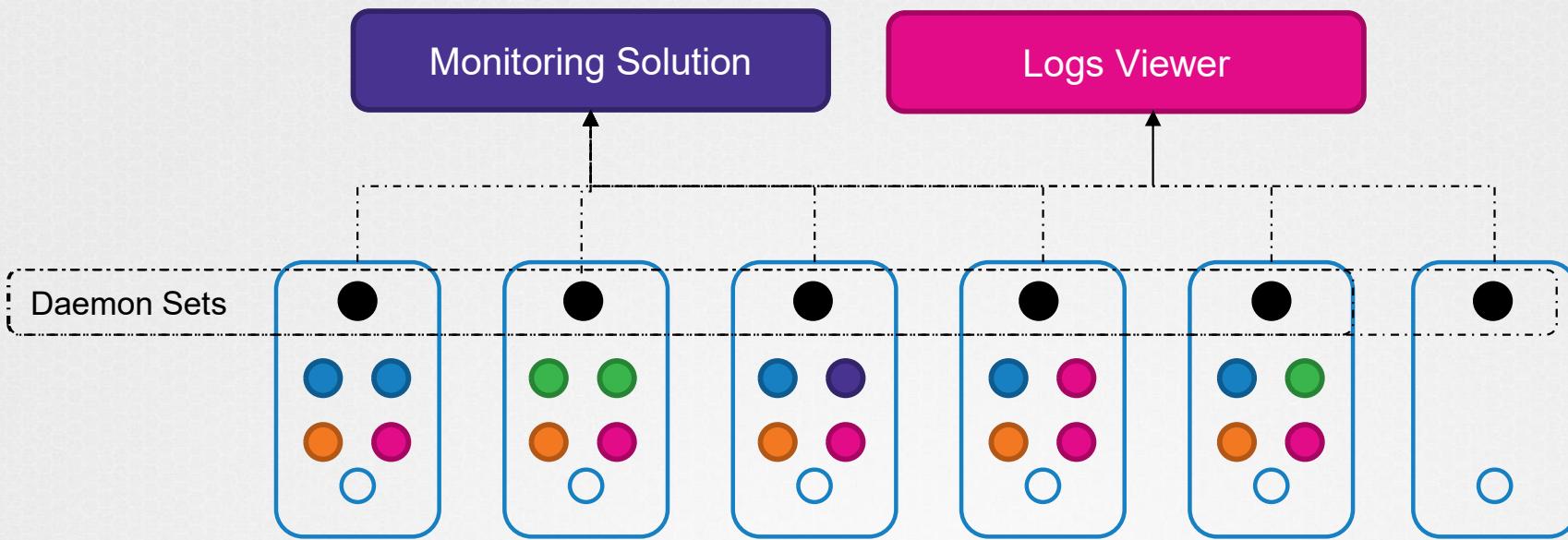
RESOURCE LIMITS



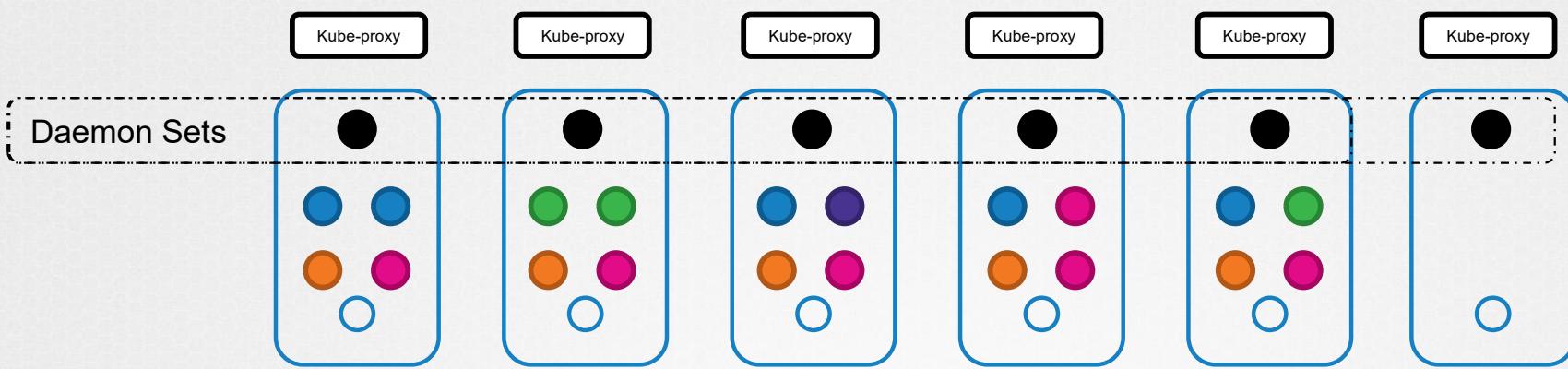
■ Daemon Sets



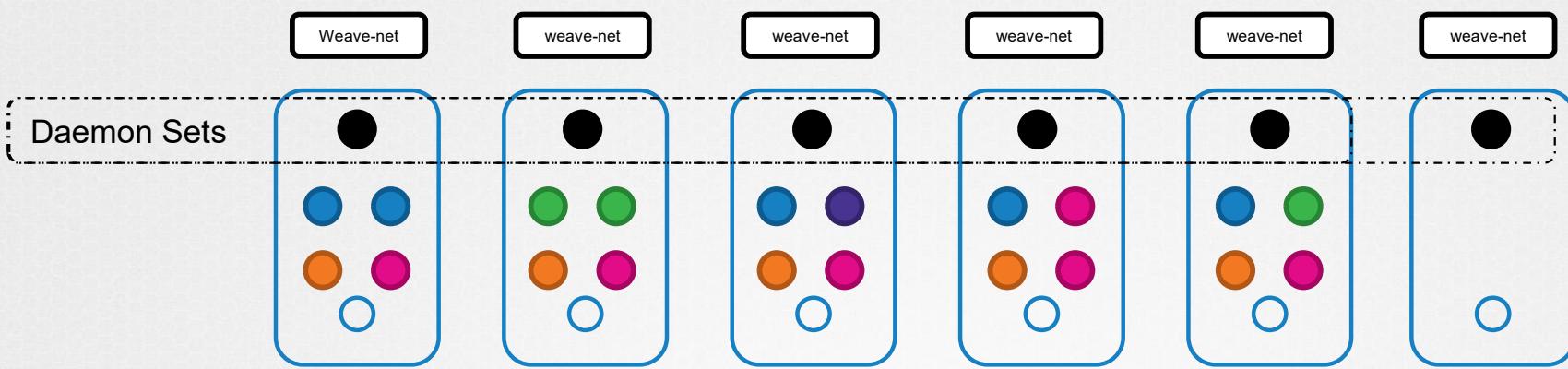
I Daemon Sets – UseCase



■ Daemon Sets – UseCase – kube-proxy



■ Daemon Sets – UseCase – Networking



■ DaemonSet Definition

daemon-set-definition.yaml

```
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: monitoring-daemon
spec:
  selector:
    matchLabels:
      app: monitoring-agent
  template:
    metadata:
      labels:
        app: monitoring-agent
    spec:
      containers:
        - name: monitoring-agent
          image: monitoring-agent
```

replicaset-definition.yaml

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: monitoring-daemon
spec:
  selector:
    matchLabels:
      app: monitoring-agent
  template:
    metadata:
      labels:
        app: monitoring-agent
    spec:
      containers:
        - name: monitoring-agent
          image: monitoring-agent
```

▶ kubectl create -f daemon-set-definition.yaml

daemon-set Created

View DaemonSets

```
kubectl get daemonsets
```

NAME	DESIRED	CURRENT	READY	UP-TO-DATE	AVAILABLE	AGE
monitoring-daemon	1	1	1	1	1	41

```
kubectl describe daemonsets monitoring-daemon
```

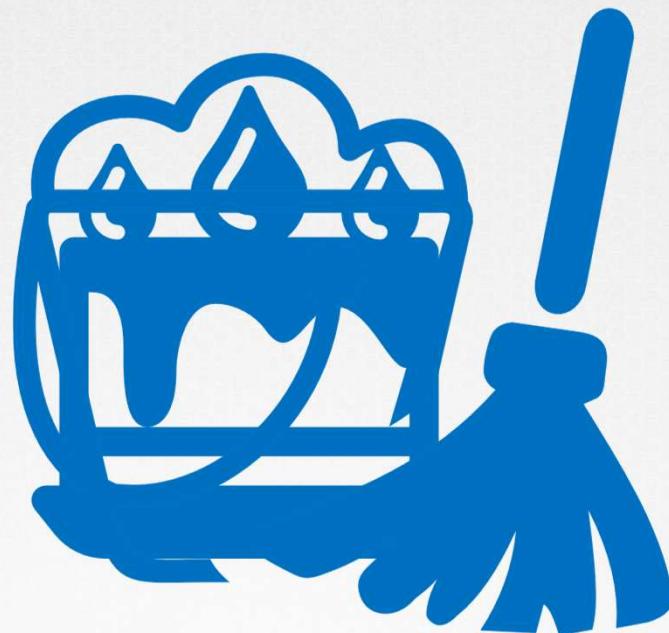
```
Name:           monitoring-daemon
Selector:       name=monitoring-daemon
Node-Selector:  <none>
Labels:         name=monitoring-daemon
Desired Number of Nodes Scheduled: 2
Current Number of Nodes Scheduled: 2
Number of Nodes Scheduled with Up-to-date Pods: 2
Number of Nodes Scheduled with Available Pods: 1
Number of Nodes Misscheduled: 0
Pods Status:   2 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:        app=monitoring-agent
  Containers:
```



Serverless



|Analogy

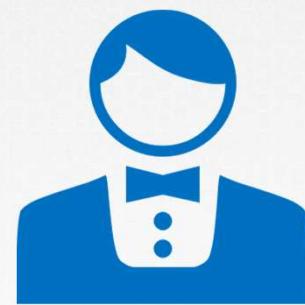


nj0

Analogy



Before the Party



During the Party



After the Party

Slide 32

nj0 does it make sense to have this image for during the party audio where mumshad says keep things tidy...
nimesha jinarajadasa, 2023-05-31T03:15:56.458

JD0 0 Used the waiter/server illustration instead
John Rhey Dayang, 2023-05-31T06:49:36.317

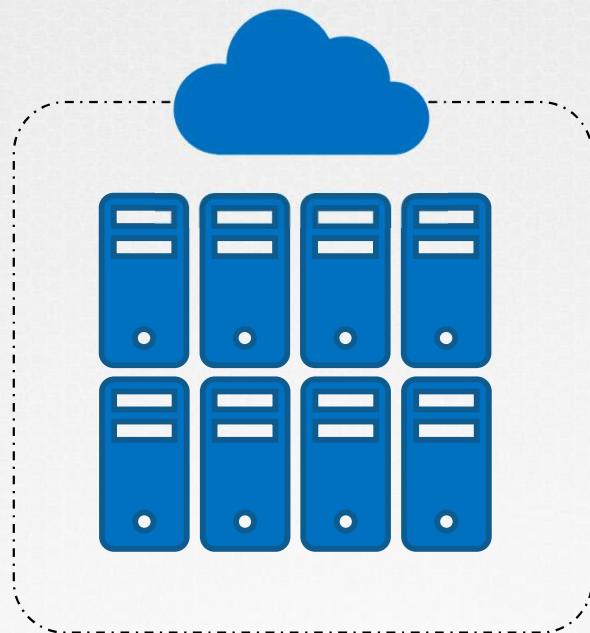
AC0 1 Can we resolve??
Aditya Chauhan, 2023-06-05T14:40:06.678

nj0 2 yes
nimesha jinarajadasa, 2023-06-05T14:55:07.570

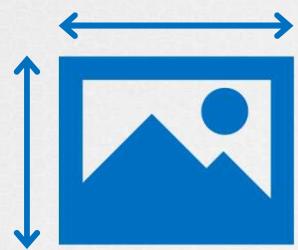
| Traditional Model



Serverless Computing



Tasks in the Application



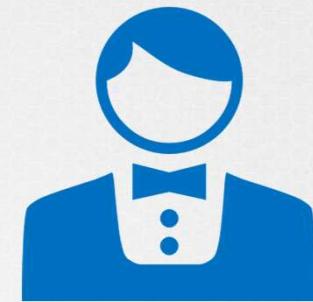
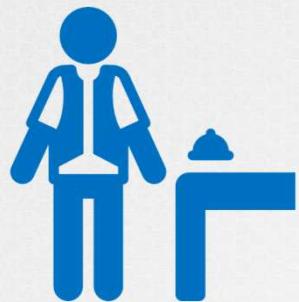
Slide 35

- nj0** Why we show Function as a service in this slide? since we talk about it in the next slide.
nimesha jinarajadasa, 2023-05-31T03:21:22.782
- JD0 0** That was suppose to introduce the topic Nimesha. Already removed it. Maybe we'll just do it in the next slide . Thanks
John Rhey Dayang, 2023-05-31T05:59:24.460
- JD0 1** I removed it. Perhaps we can use the next slide for that.
John Rhey Dayang, 2023-05-31T06:50:11.244

I Function-As-A-Service



| Function-As-A-Service

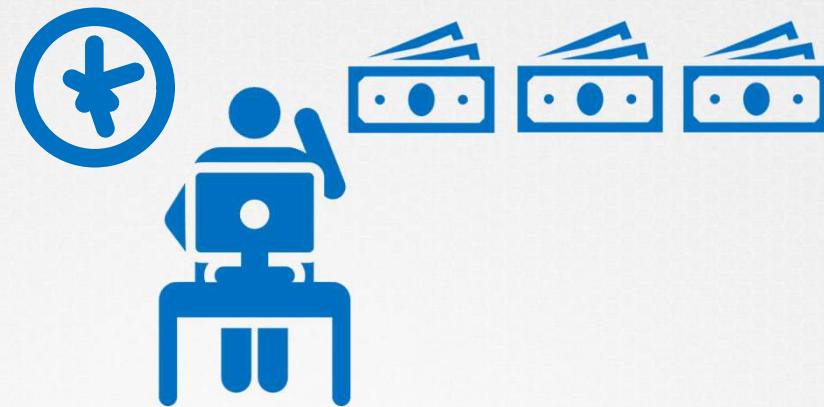


| Function-As-A-Service



nj0

| Pay As You Go Model

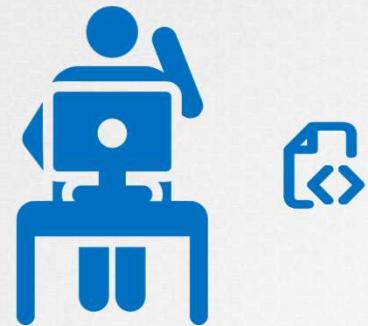


Slide 39

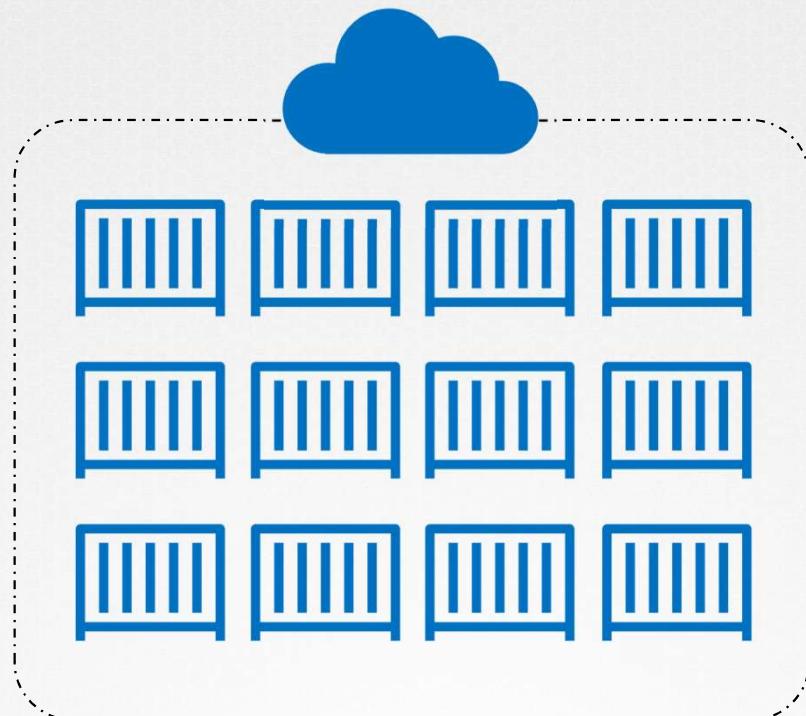
nj0 can we do an animation like,
when the clock ticks, more cost by showing more bucks in the screen with the time increases.
nimesha jinarajadasa, 2023-05-31T03:27:46.576

JDO 0 Done
John Rhey Dayang, 2023-05-31T06:52:26.025

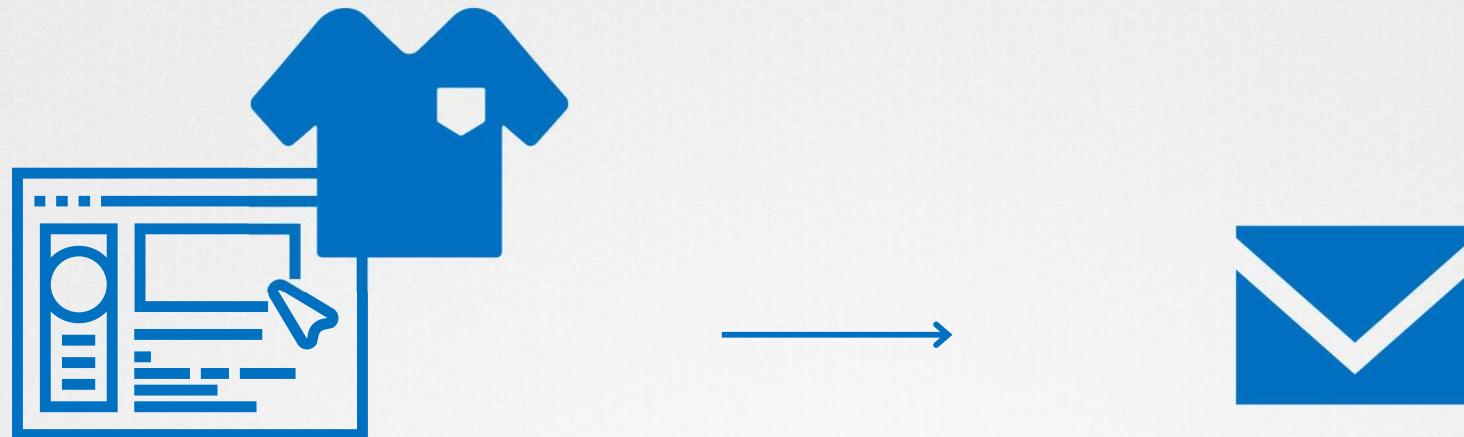
| Serverless Computing



Serverless Computing



| Serverless Computing



Serverless Computing

Function-As-A-Service (FaaS)



AWS Lambda

Serverless Computing



Send Grid



Amazon Simple Email
Service (SES)

|Cloud Providers



AWS Lambda



AWS Fargate



Azure Functions



Google Cloud Functions



Amazon Web
Services



Microsoft
Azure



Google Cloud



IBM Cloud

I Kubernetes Serverless

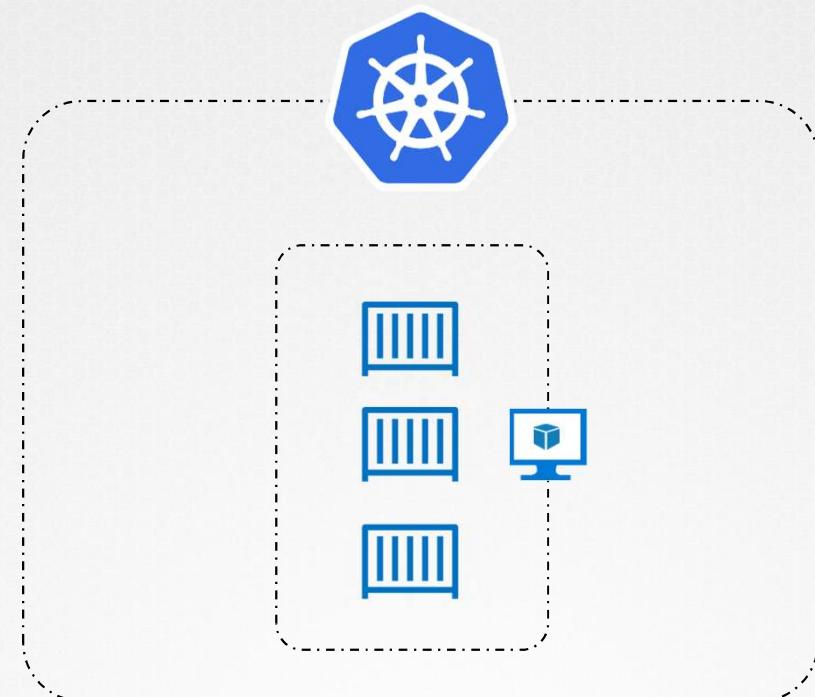


I Kubernetes Serverless

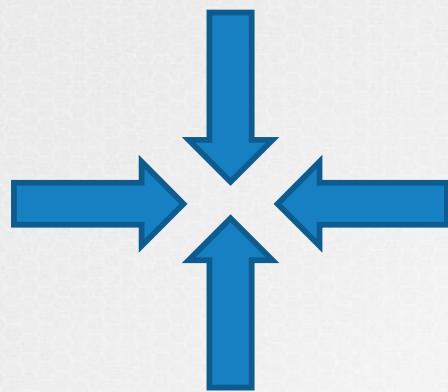


An open-source container
orchestration platform

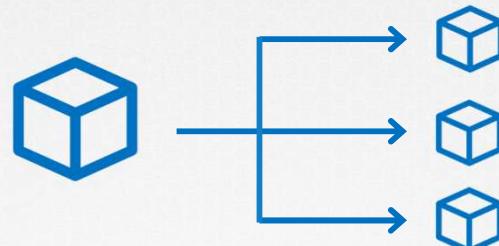
I Kubernetes Serverless



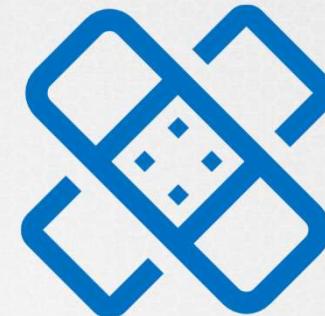
Kubernetes Serverless



Automatic Scaling

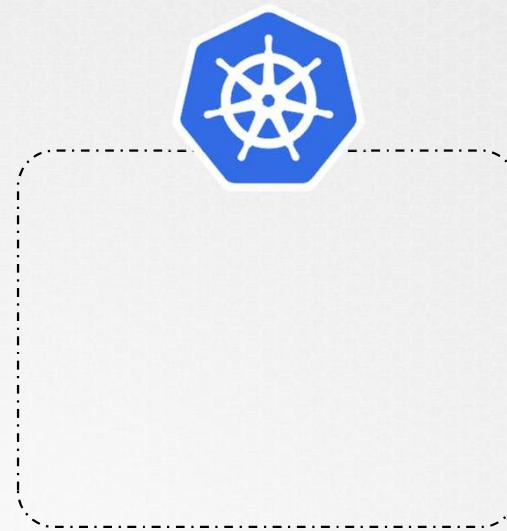
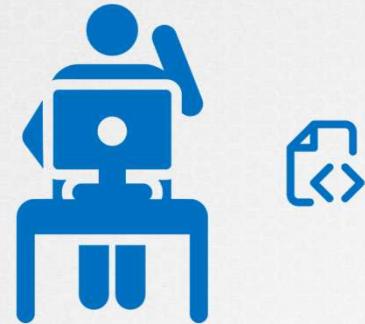


Load Balancing



Self-healing

Kubernetes Serverless



I Kubernetes Serverless



Knative



OpenFaaS

nj0

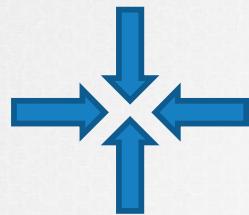
Kubernetes Serverless



Knative



OpenFaaS



Scaling



Event Triggering



Lifecycle Management

Slide 53

nj0

The words, knative, openfaas,..scaling

Those words should be in one line.

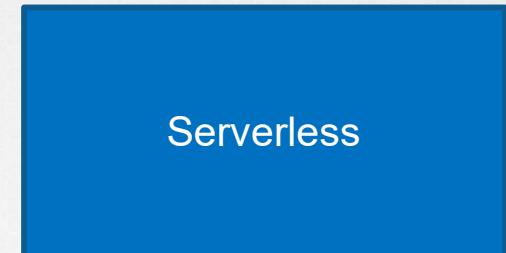
nimesha jinarajadasa, 2023-05-31T03:35:35.327

| Kubernetes Serverless



Scalability

Reliability



Serverless

Reduced Operational Cost

Improved Developer Productivity

Slide 54

nj0 can we list kubernetes benefits scalability and reliability in left side

then Serverless benefits in right side

and finally merge those together, also use the word serverless inside a box instead this image in the right side which doesn't make sense,
nimesha jinarajadasa, 2023-05-31T03:38:58.225

Kubernetes Serverless



Scalability

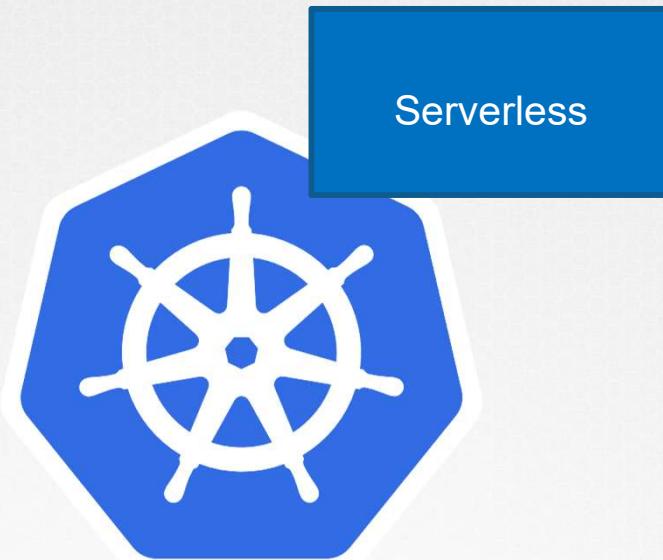
Reliability

Serverless

Reduced Operational Cost

Improved Developer Productivity

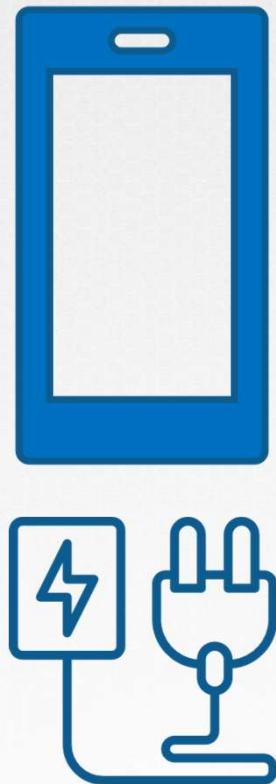
I Kubernetes Serverless



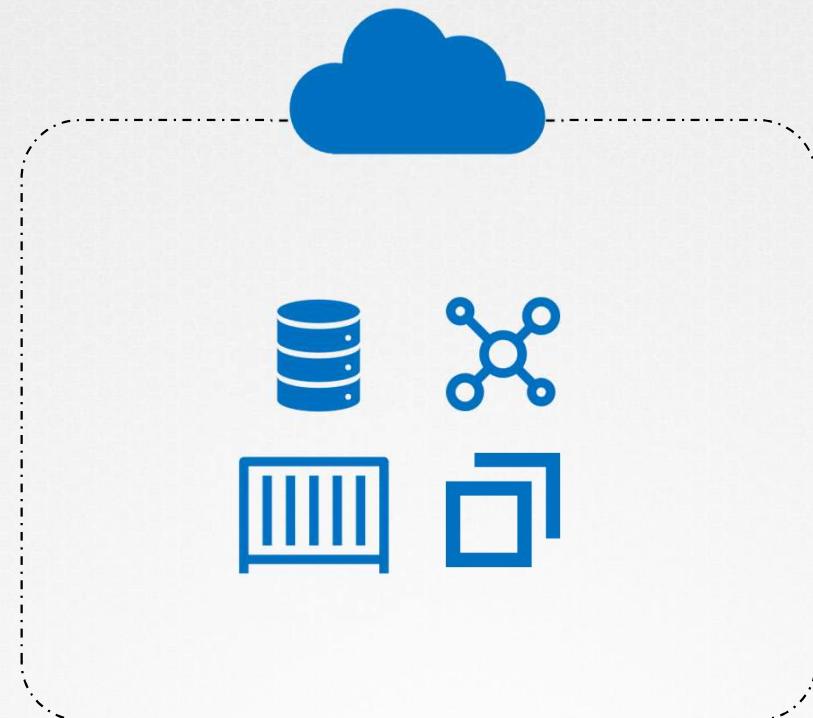
Open Standards



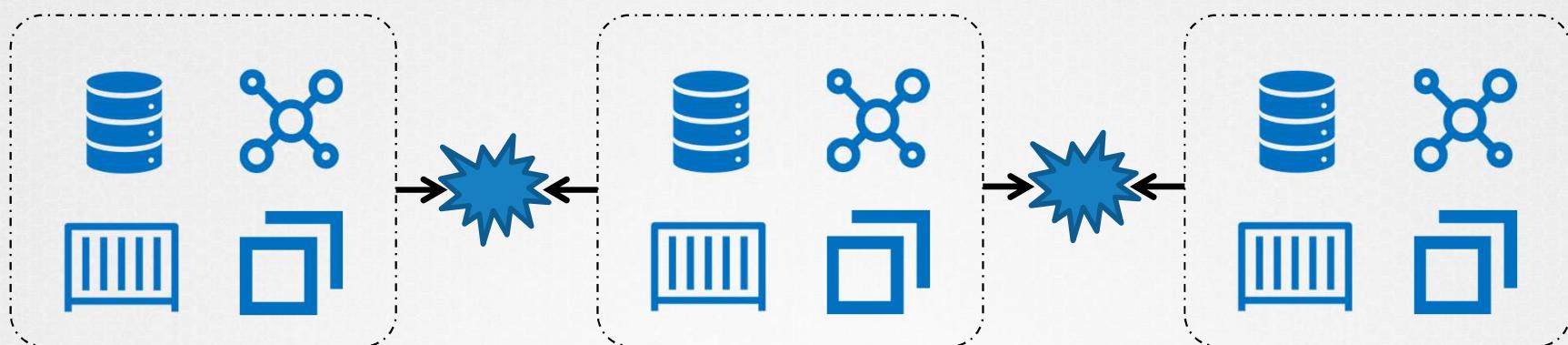
Analogy



|Cloud Native Technologies



|Cloud Native Technologies



Slide 60

nj0 What it means by Language1 language 2...

nimesha jinarajadasa, 2023-05-31T07:50:23.052

JD0 0 It was mentioned that "without common language and guidelines,..." then OCI was introduced.

John Rhey Dayang, 2023-06-01T01:21:15.433

nj0 1 in this context language doesn't mean normal spoken language.

So, this is bit confusing.

let's not add language 1, 2,3, here.

instead show that these technologies can't communicate or
work together.

nimesha jinarajadasa, 2023-06-04T17:13:05.460

Open Standards

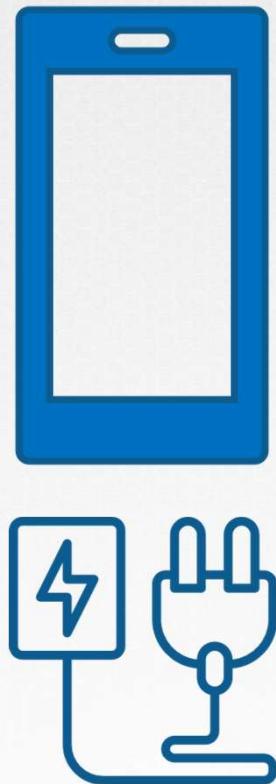
Interoperability
Portability
Vendor Neutrality



Specifications
Protocols
Formats



Analogy



I Analogy



|Analogy

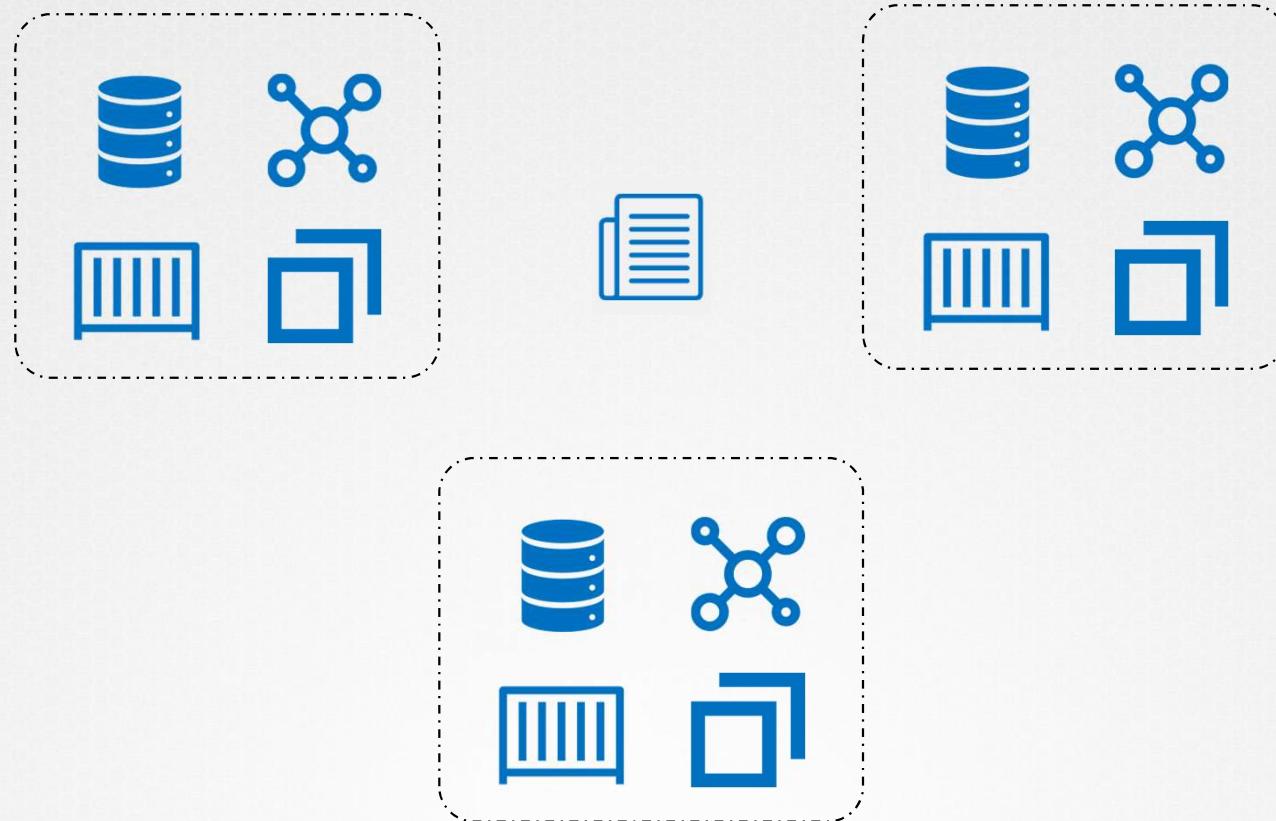


USA

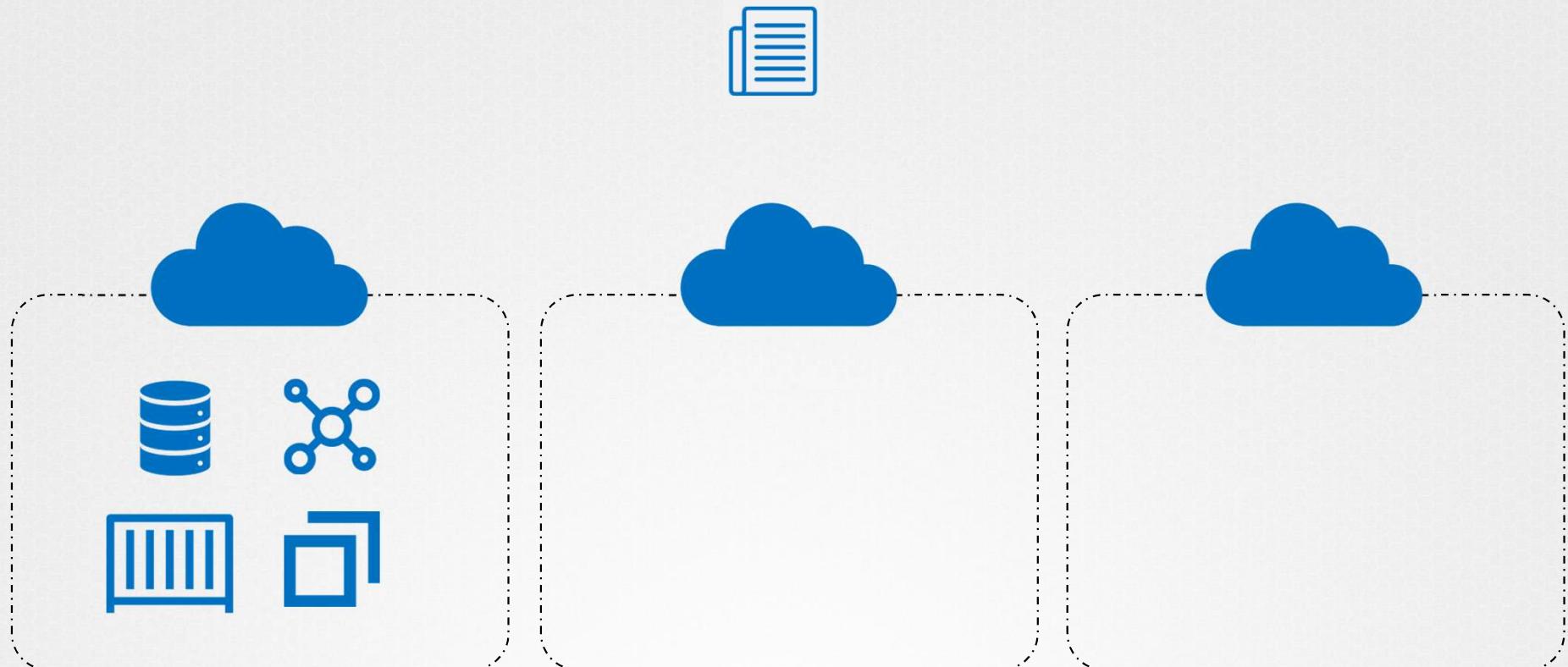


Japan

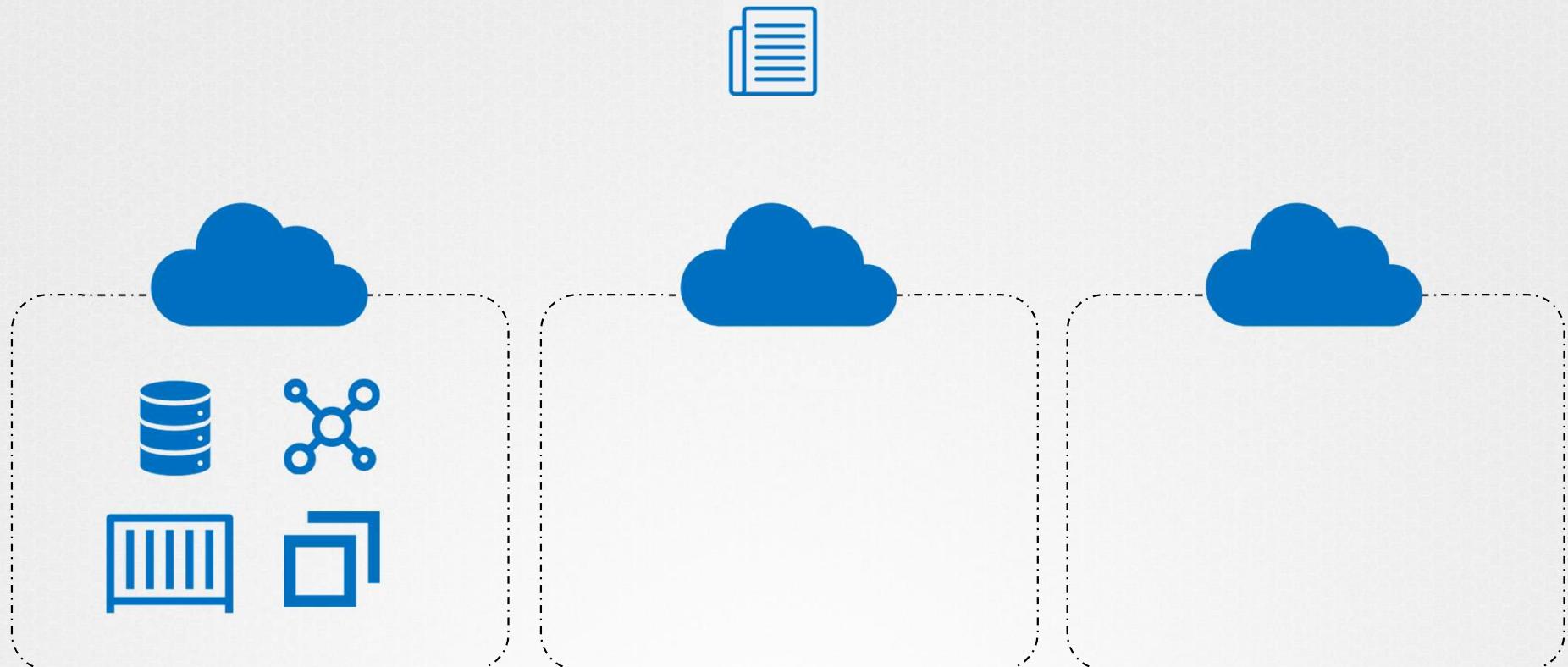
| Necessity of Open Standards



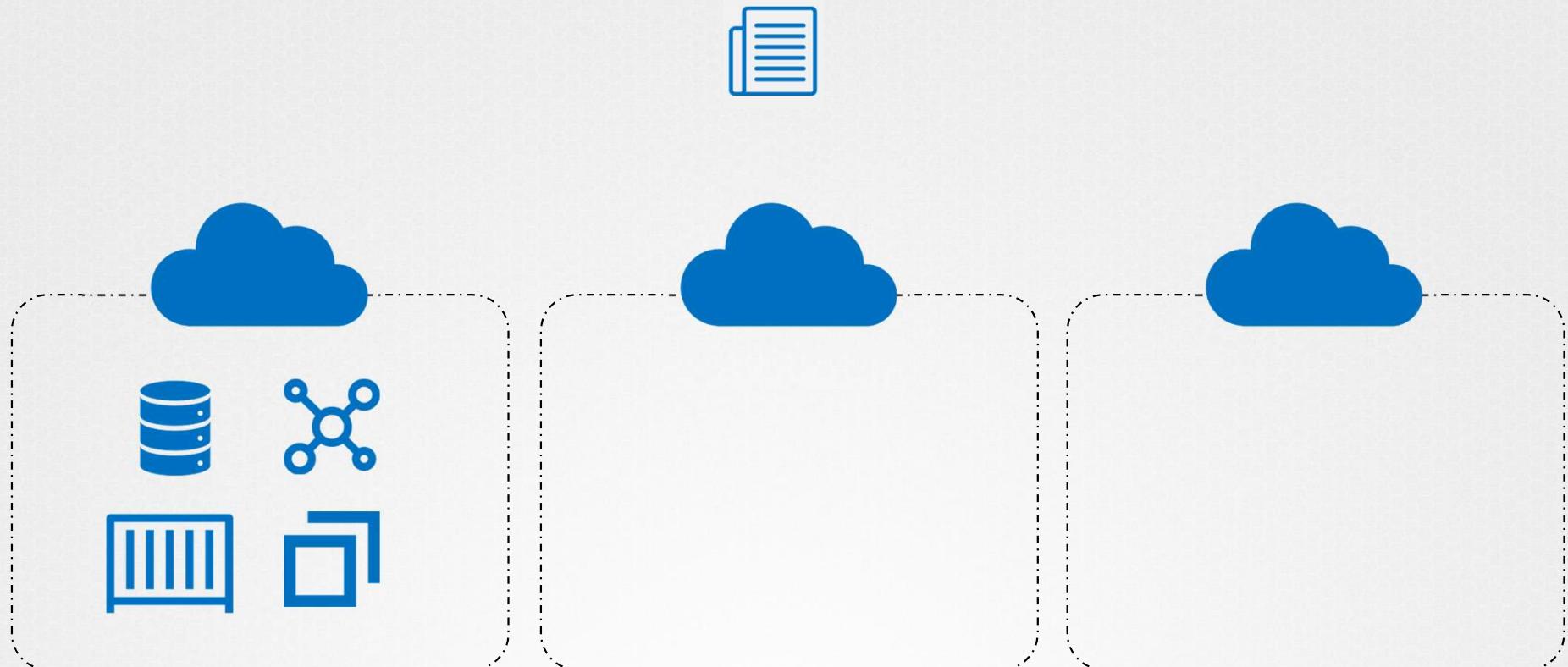
| Necessity of Open Standards



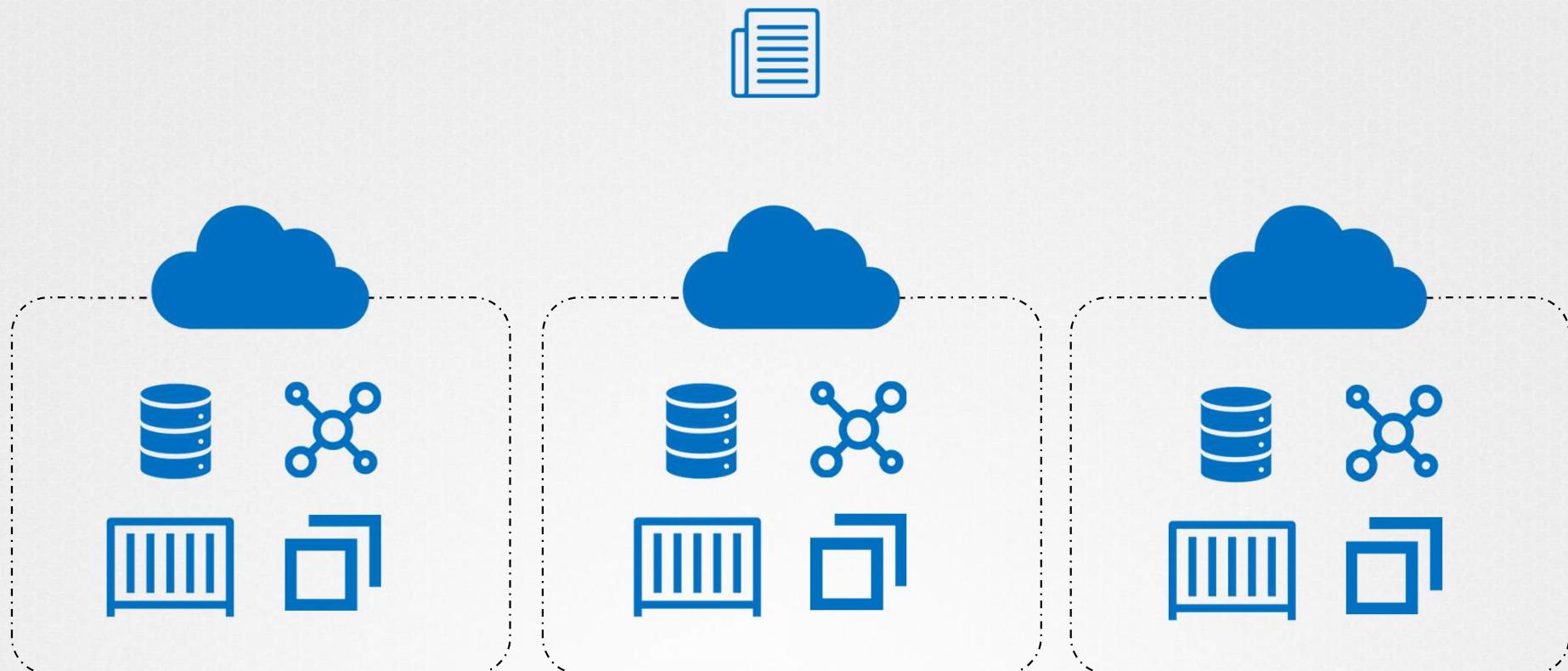
| Necessity of Open Standards



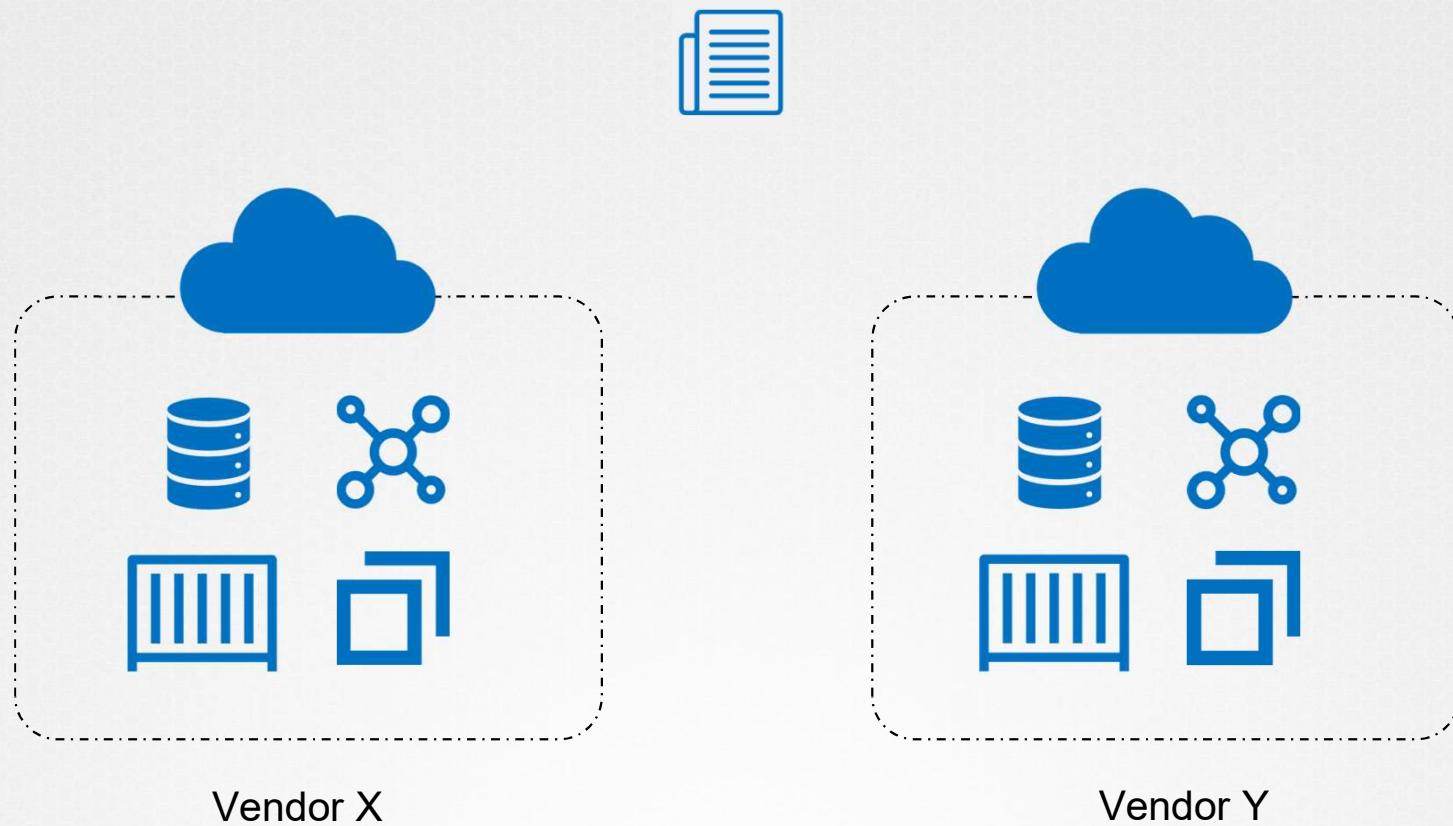
| Necessity of Open Standards



| Necessity of Open Standards



Necessity of Open Standards

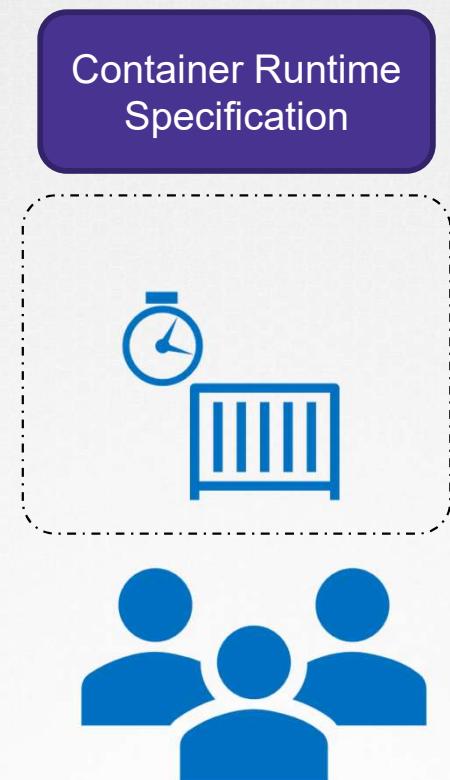


Open Container Initiatives (OCI)



nj0

Open Container Initiatives (OCI)



Slide 72

nj0 Can we use the term container runtime specification.
nimesha jinarajadasa, 2023-05-31T07:52:25.446

nj0 0 not the term container runtime spec. bundle, it should be file system bundle
nimesha jinarajadasa, 2023-05-31T07:53:12.041

JD0 1 This is correct, I added slide/s for the File system bundle just before this slide
John Rhey Dayang, 2023-06-01T01:27:12.156

Open Container Initiatives (OCI)

File System
Bundle



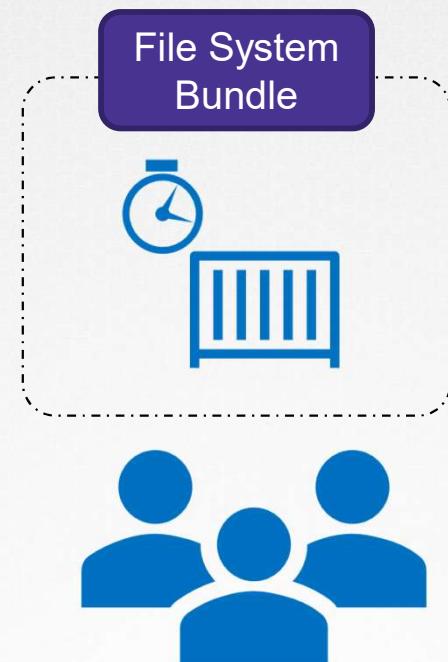
Image



Image Specification

nj0

Open Container Initiatives (OCI)



Slide 74

- nj0** Can we use the term container runtime specification.
nimesha jinarajadasa, 2023-05-31T07:52:25.446
- nj0 0** not the term container runtime spec. bundle, it should be file system bundle
nimesha jinarajadasa, 2023-05-31T07:53:12.041
- JD0 1** This is correct, I added slide/s for the File system bundle just before this slide
John Rhey Dayang, 2023-06-01T01:27:12.156

nj0

Open Container Initiatives (OCI)



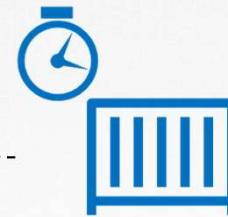
Downloading



Unpacking



Running



Container Runtime
Specification

Slide 75

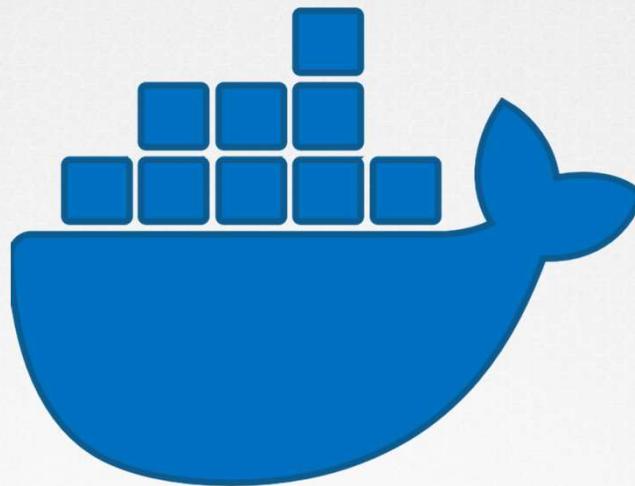
nj0 same here, container runtime specification not bundle.

nimesha jinarajadasa, 2023-05-31T08:12:35.836

JD0 0 Please see response in slide 103

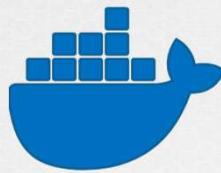
John Rhey Dayang, 2023-06-01T01:27:54.071

Open Container Initiatives (OCI)



Docker

Open Container Initiatives (OCI)



containerd



cri-o



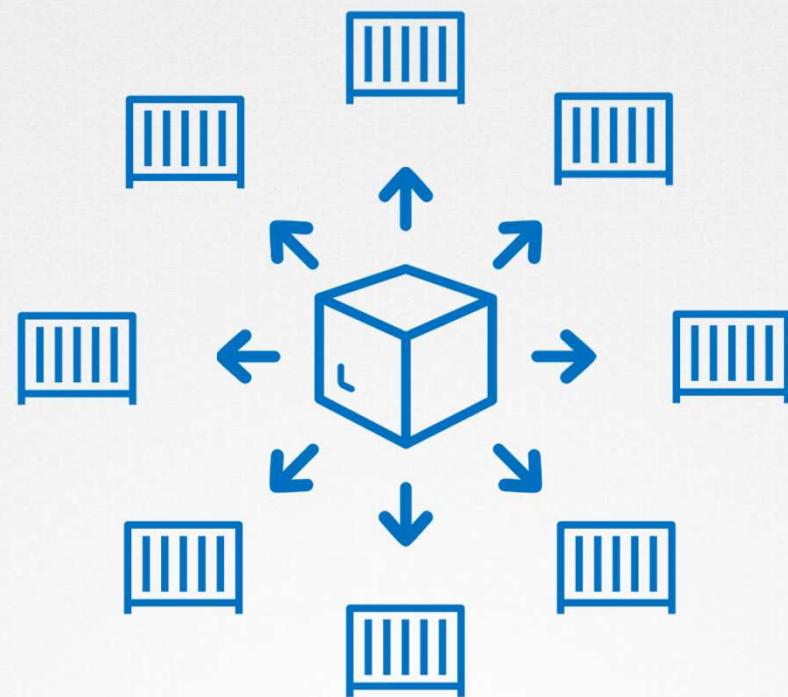
katacontainers



gVisor



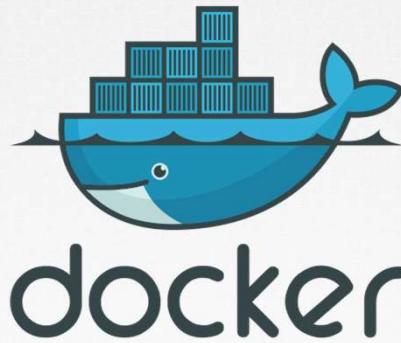
Distribution Specification



Distribution Specification



Amazon Elastic
Container Registry(ECR)

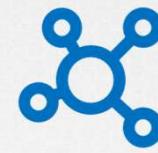


Microsoft
Azure

Kubernetes Open Standards



Runtime



Service
Mesh

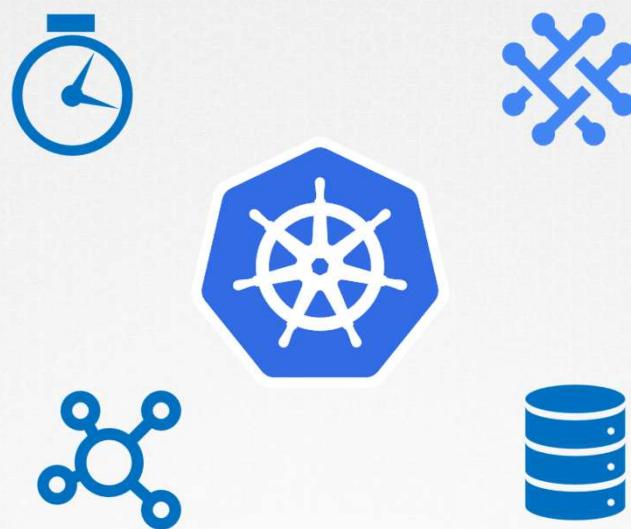


Networking



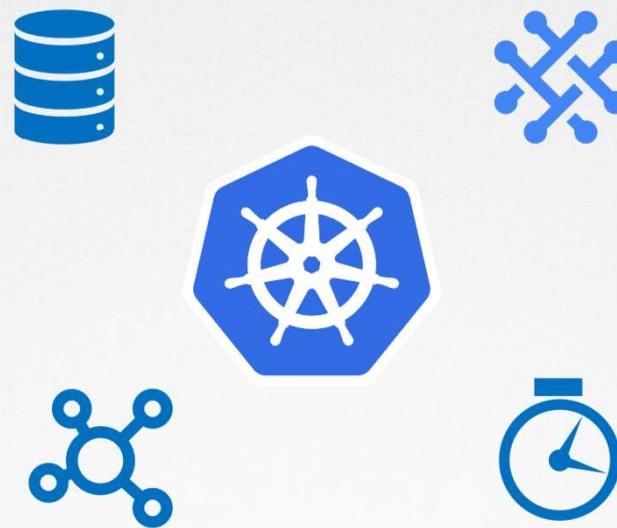
Storage

Kubernetes Open Standards



nj0

I Kubernetes Open Standards

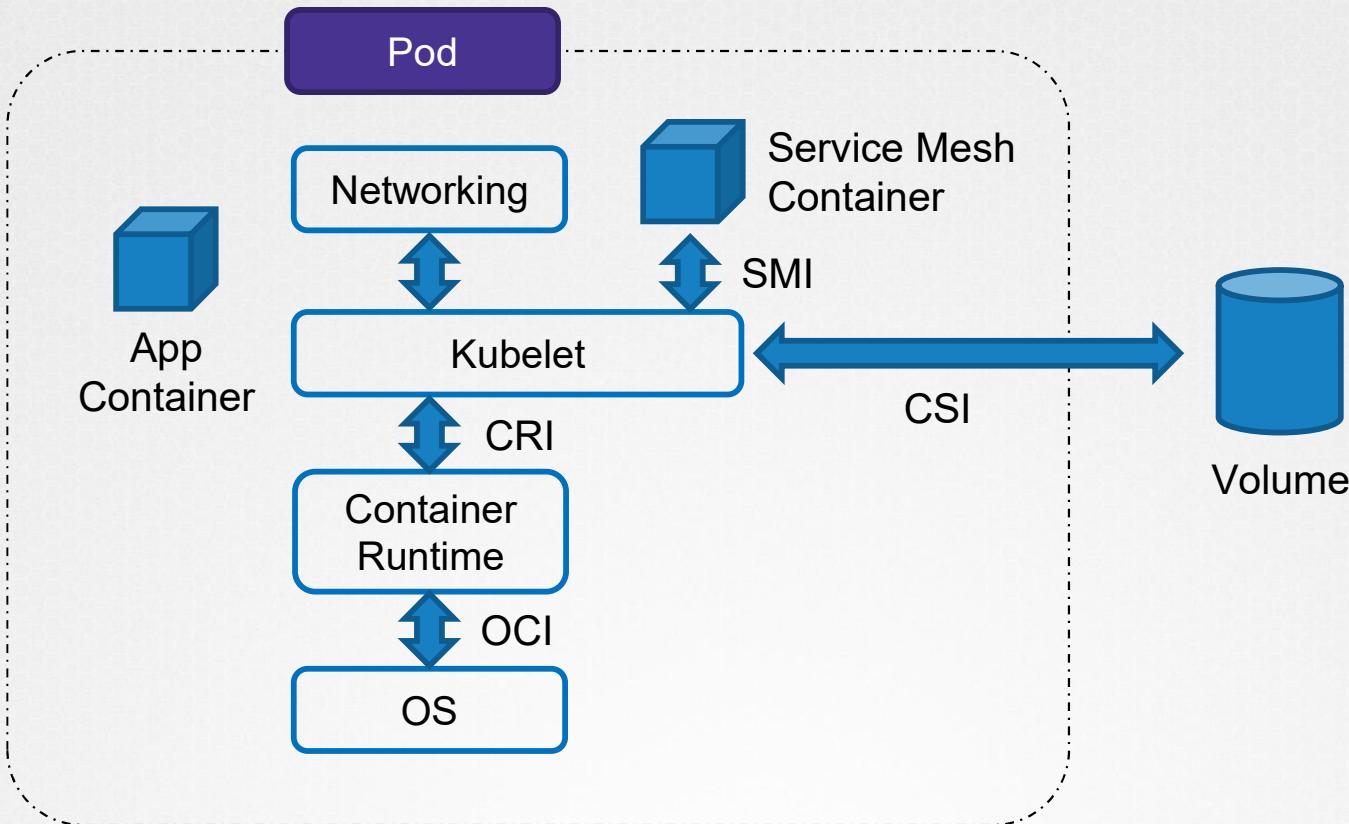


Slide 82

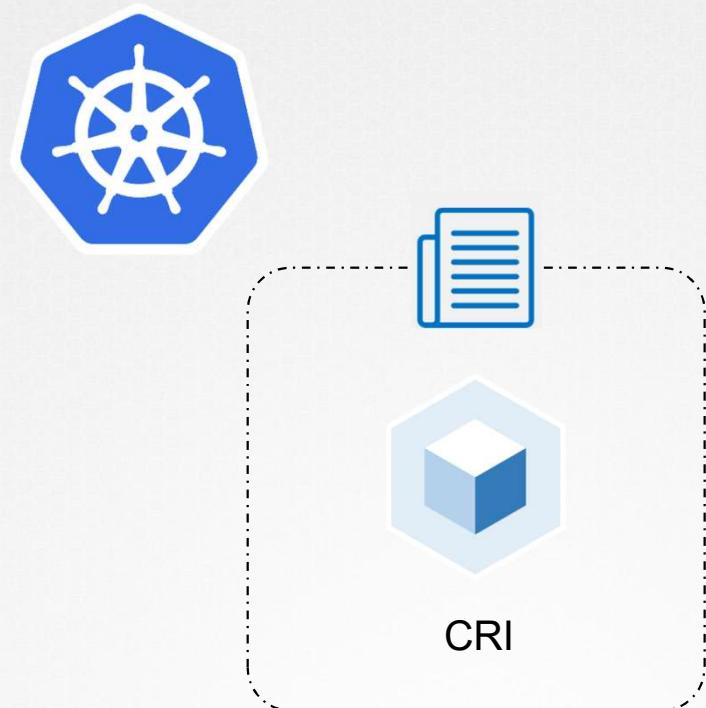
nj0 can we add the image in the script here? maybe improved version of that image.
nimesha jinarajadasa, 2023-05-31T08:15:49.452

JD0 0 In the next slide
John Rhey Dayang, 2023-06-01T01:54:34.338

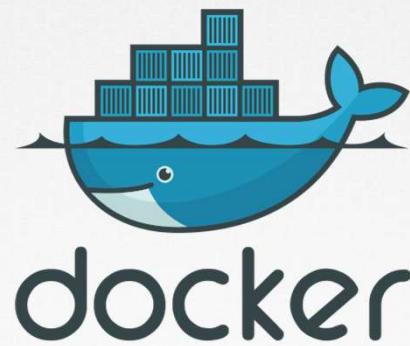
Kubernetes Open Standards



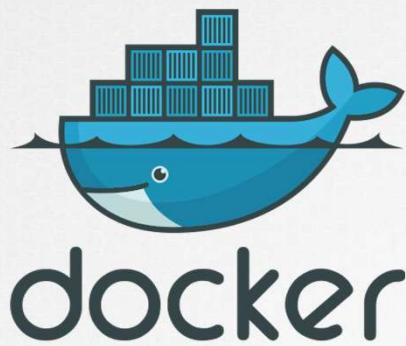
Container Runtime Interface (CRI)



|Container Runtime Interface (CRI)



|Container Runtime Interface (CRI)



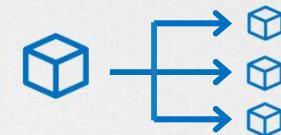
Container Network Interface (CNI)



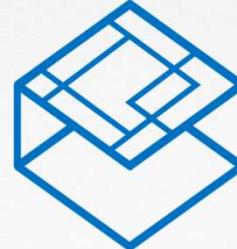
Network
Policies



Service Discovery



Load
Balancing

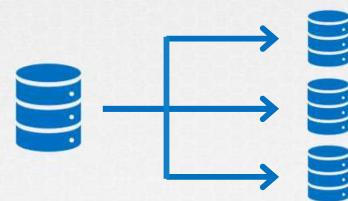


Container Network
Interface

Container Storage Interface (CSI)



Cloud
Storage



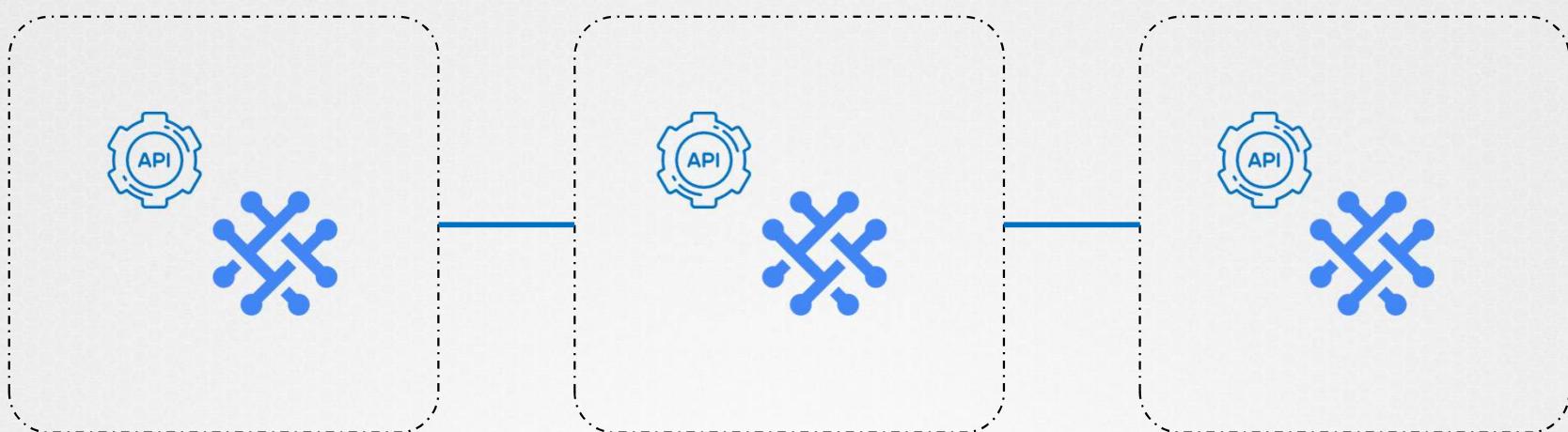
Network-attached Storage



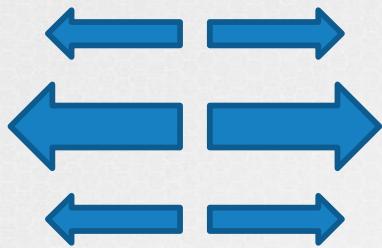
Storage Area Networks



Service Mesh Interface (SMI)



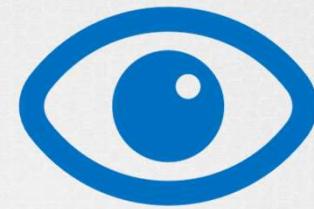
Service Mesh Interface (SMI)



Traffic Management



Security



Observability



Cost Management



nj0

Analogy



Virtual Machines



Volumes



Web
Services

Slide 92

nj0

volumes means, storage.

change this icon.

nimesha jinarajadasa, 2023-06-02T04:12:16.689

Analogy

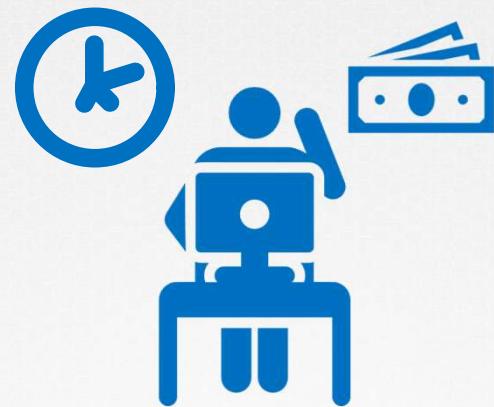


Cloud Cost Management

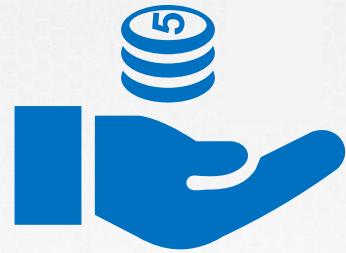
|Cloud Cost Management



|Cloud Cost Management



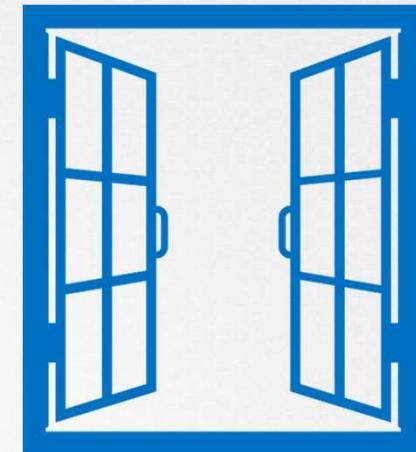
|Cloud Cost Management



|Cloud Cost Management



|Cloud Cost Management



Slide 98

nj0 what these windows mean here?

nimesha jinarajadasa, 2023-06-02T04:13:47.909

nj0 0 ah okay now I got it, its this analogy about ac on while leaving windows open

nimesha jinarajadasa, 2023-06-02T04:16:27.818

Cloud Cost Management Strategies



On-demand



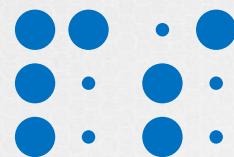
Reserved



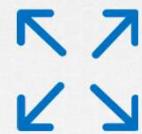
Spot

Right
Infrastructure

On-demand Instances



Prevalent



Flexible



Expensive



On-
demand

Reserved Instances



KodeKloud

Discount

nj0



Reserved

Slide 101

nj0

typo in Discount word

nimesha jinarajadasa, 2023-06-02T04:17:51.772

Reserved Instances



KodeKloud

30% Discount



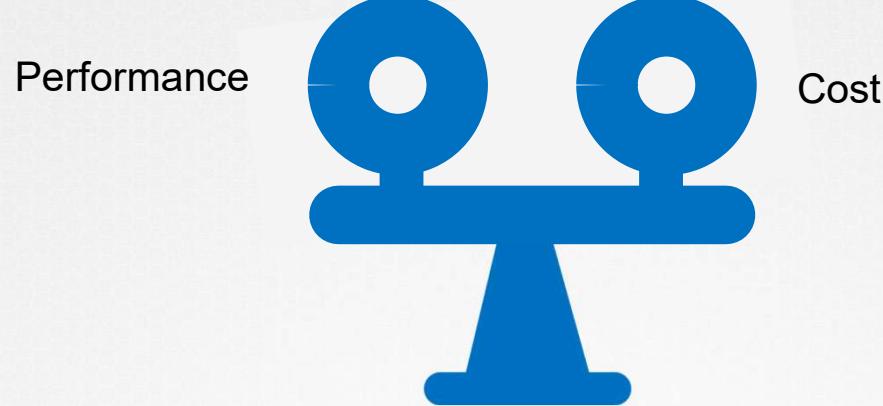
Spot Instances

90%
Discount

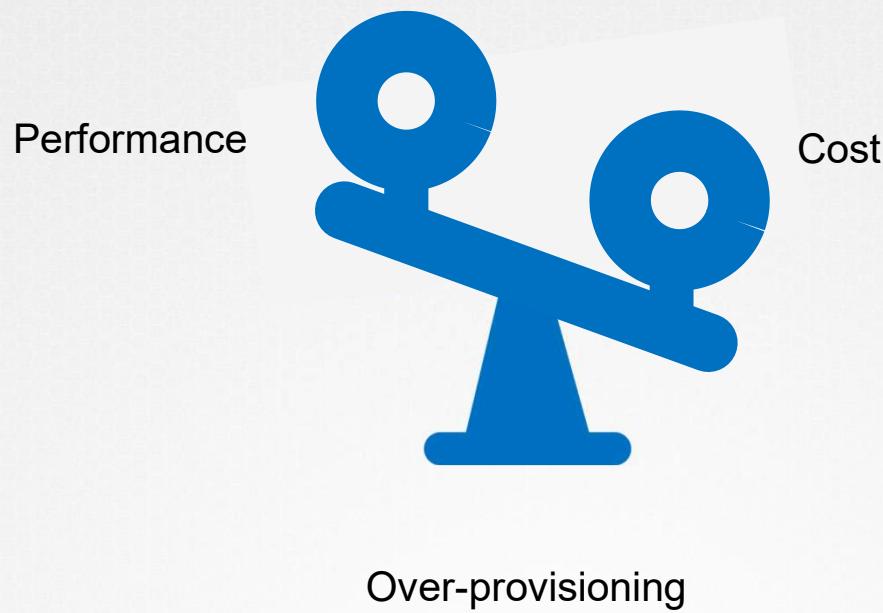


Spot

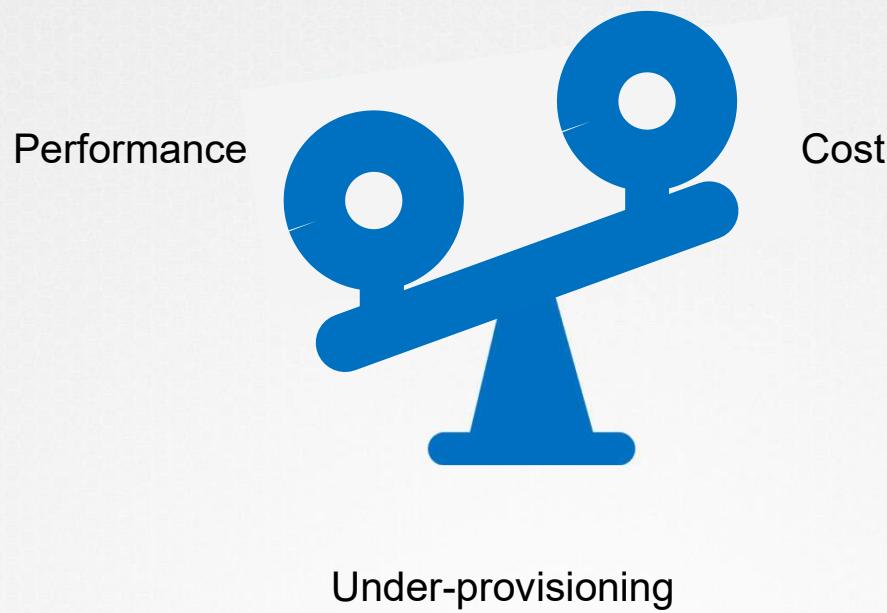
| Managing Performance and Cost through rightsizing



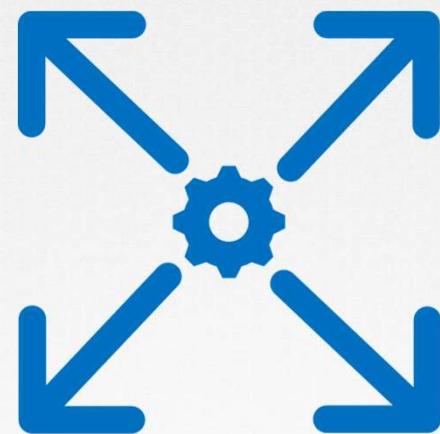
Managing Performance and Cost through rightsizing



Managing Performance and Cost through rightsizing



Managing Performance and Cost through rightsizing



Autoscaling

Managing Performance and Cost through rightsizing



Kubernetes

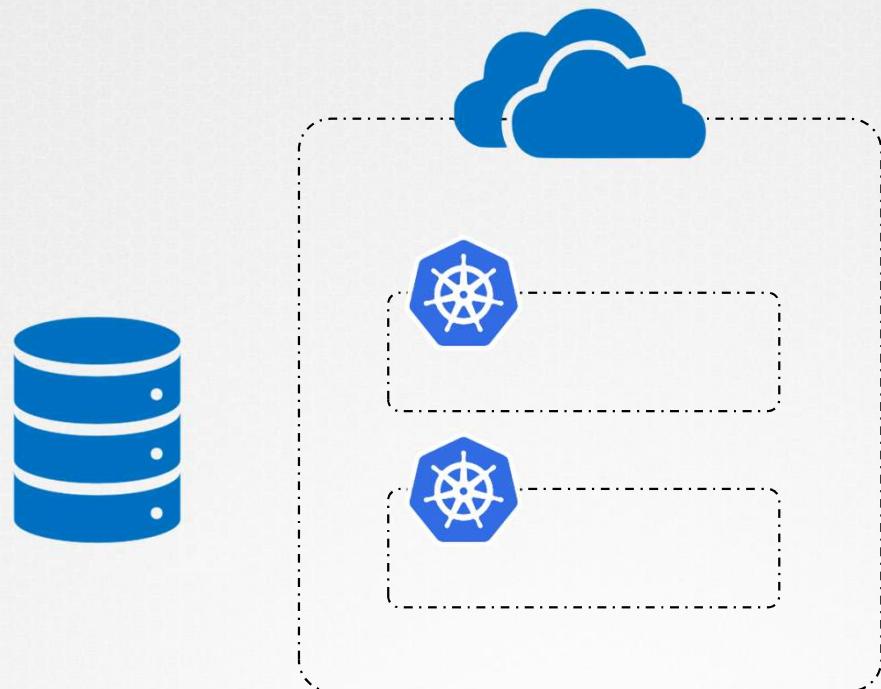
Slide 108

nj0

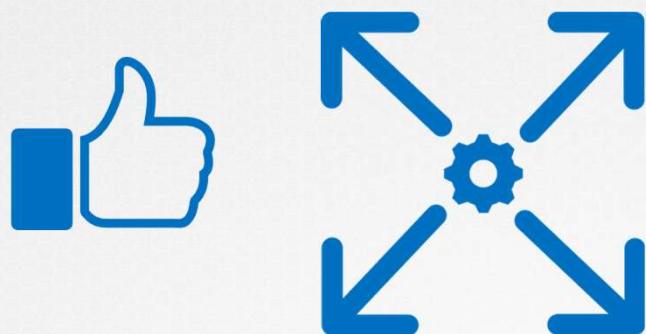
Kubenetes is a one liner

nimesha jinarajadasa, 2023-06-02T04:24:42.225

Managing Performance and Cost through rightsizing



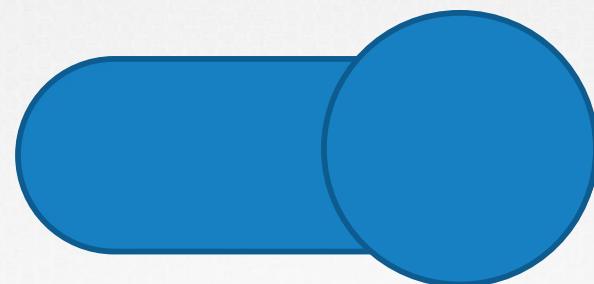
Managing Performance and Cost through rightsizing



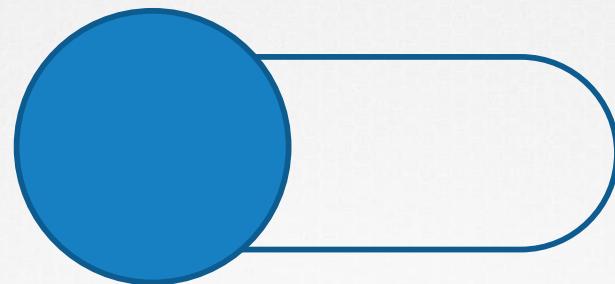
Configuring optimal upper and lower limits

Selecting appropriate performance metrics

| Scheduling non-essential instances and removing unused resources



| Scheduling non-essential instances and removing unused resources



Turn off during non-business hours and weekends

Slide 112

nj0

can we add another slide to conclusion's reading please
at least for these vendors harness, kubecost, densify ..

nimesha jinarajadasa, 2023-06-02T04:37:39.862

Conclusion



Amazon Web Services (AWS)



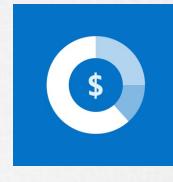
Microsoft Azure



AWS Cost Explorer



Azure Cost Management



Azure Billing

Conclusion



Harness



Kubecost



Cloudability



Densify



Cloudzero

Application Delivery Fundamentals



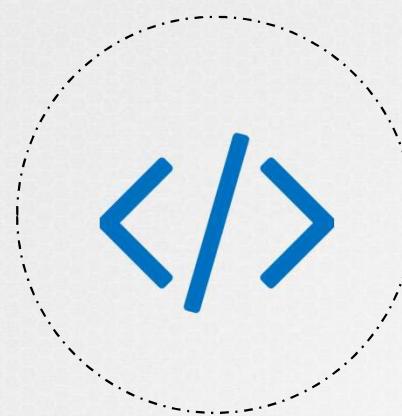
| Analogy



I Analogy



I Scenario in the Past



Building and testing
the code are manual
processes

nj0



Bugs can be introduced
during deployment



The whole process was
long and risky

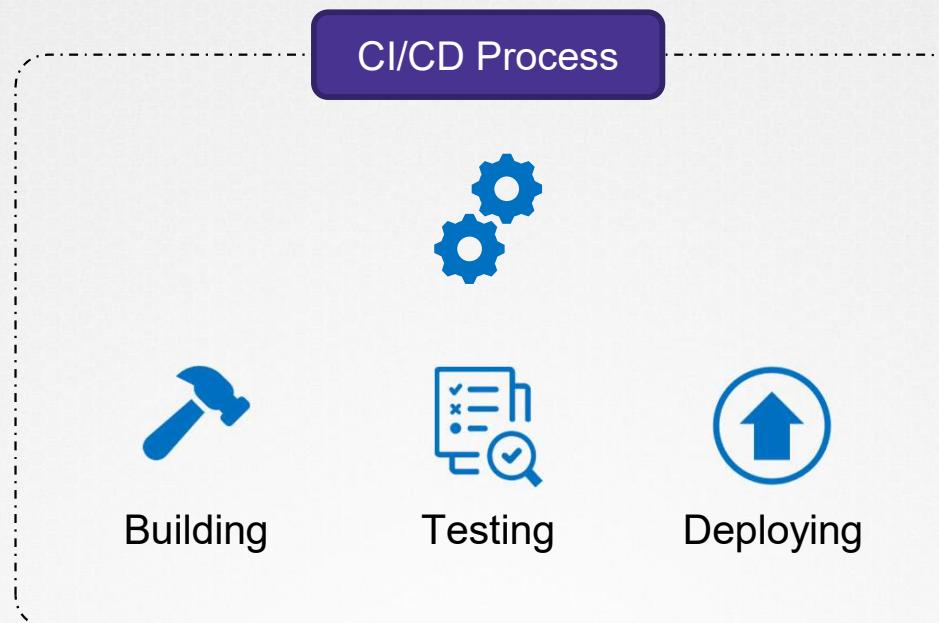
nj0

"Building and testing the code are manual processes."

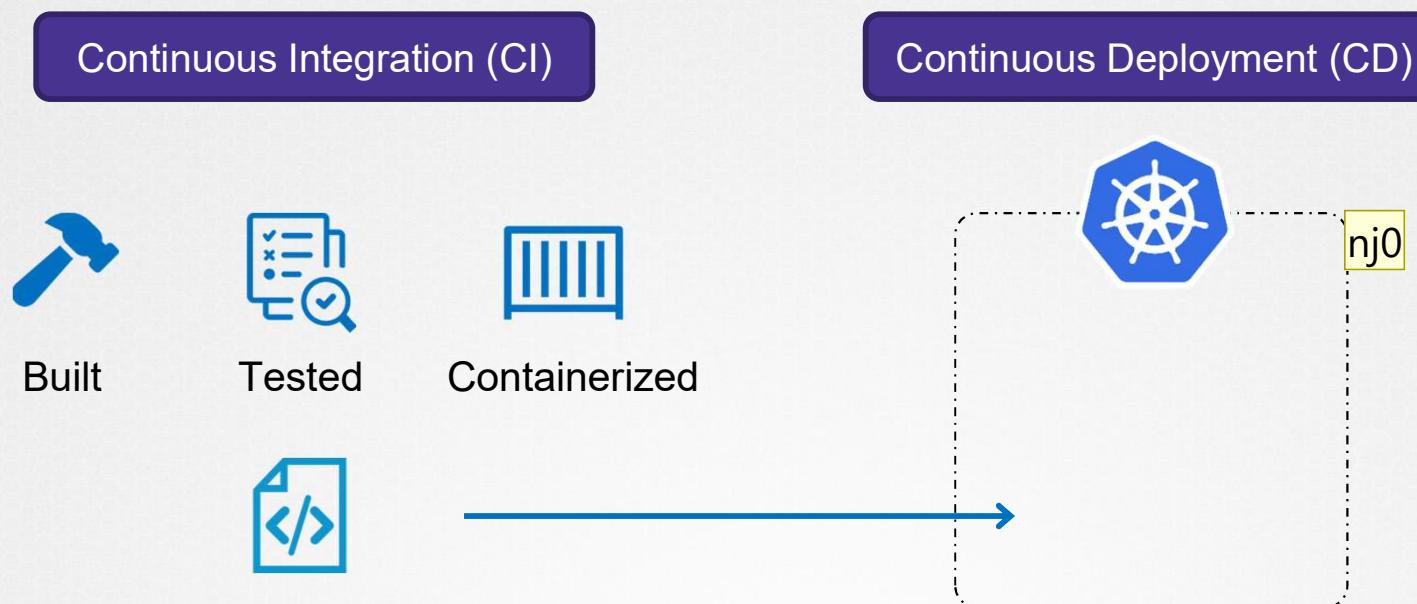
change the statement to this.(under the first icon)

nimesha jinarajadasa, 2023-06-04T17:18:21.303

I Scenario in the Present



I Scenario in the Present



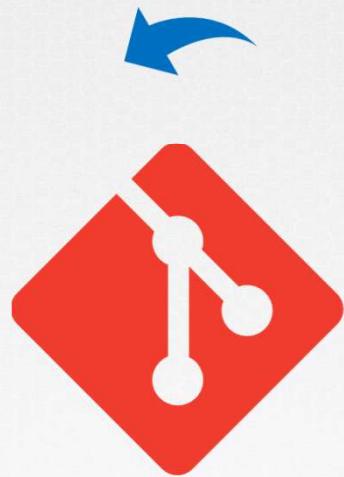
Slide 120

nj0 can we show something like delivering the code to Kubernetes cluster?
with an arrow and code file.

instead of kubernetes icon alone

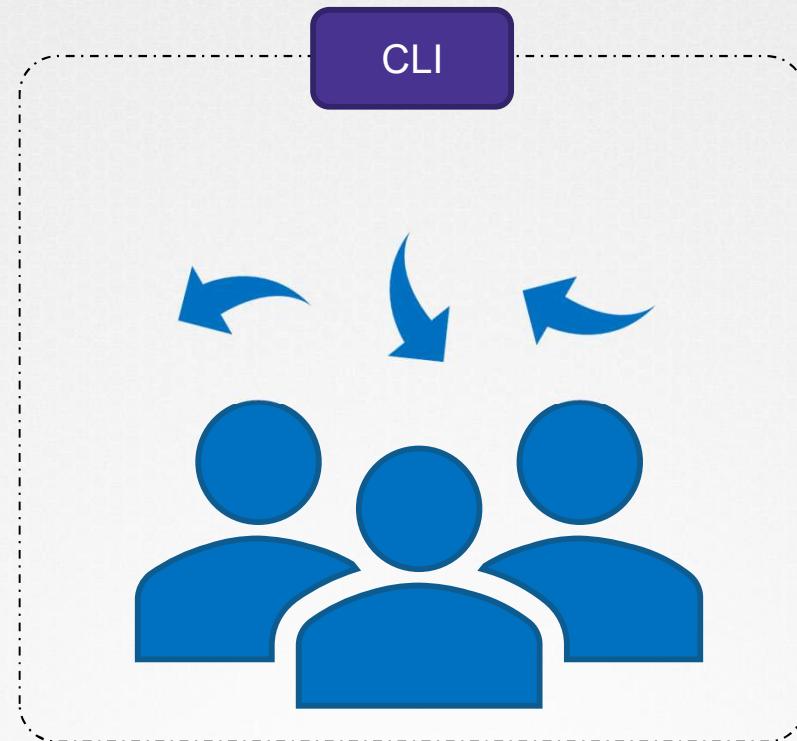
nimesha jinarajadasa, 2023-06-02T04:41:13.189

| GitOps



GitOps

I Manual Changes



| Manual Changes



Slide 123

nj0

where this slide fits?

nimesha jinarajadasa, 2023-06-02T04:45:07.775

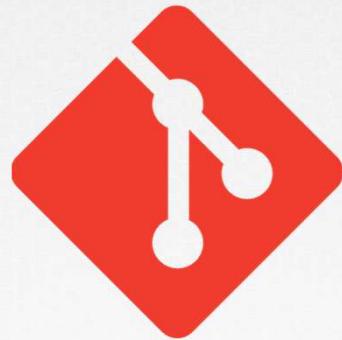
nj0 0

can we add another slide for this disaster recover plan paragraph?

nimesha jinarajadasa, 2023-06-02T04:47:11.982

nj0

| GitOps



GitOps

Slide 124

nj0

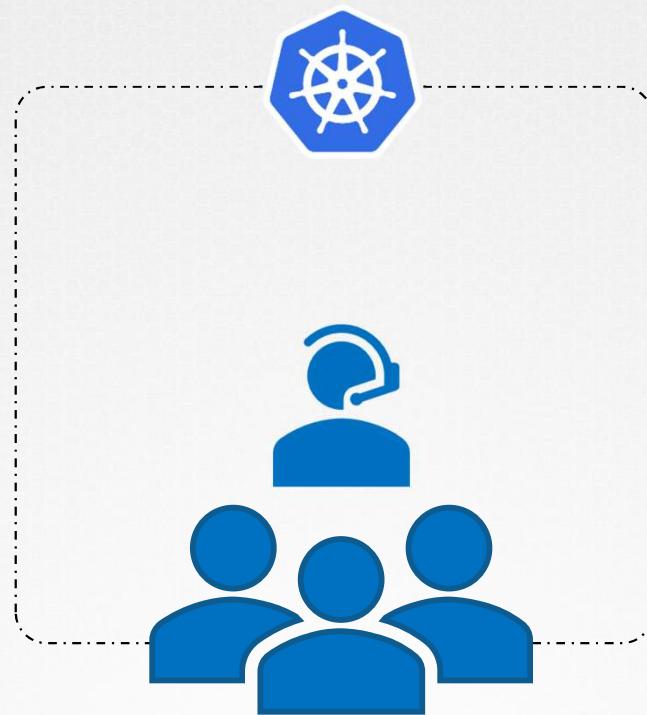
Gitops is one word

nimesha jinarajadasa, 2023-06-02T04:49:52.956

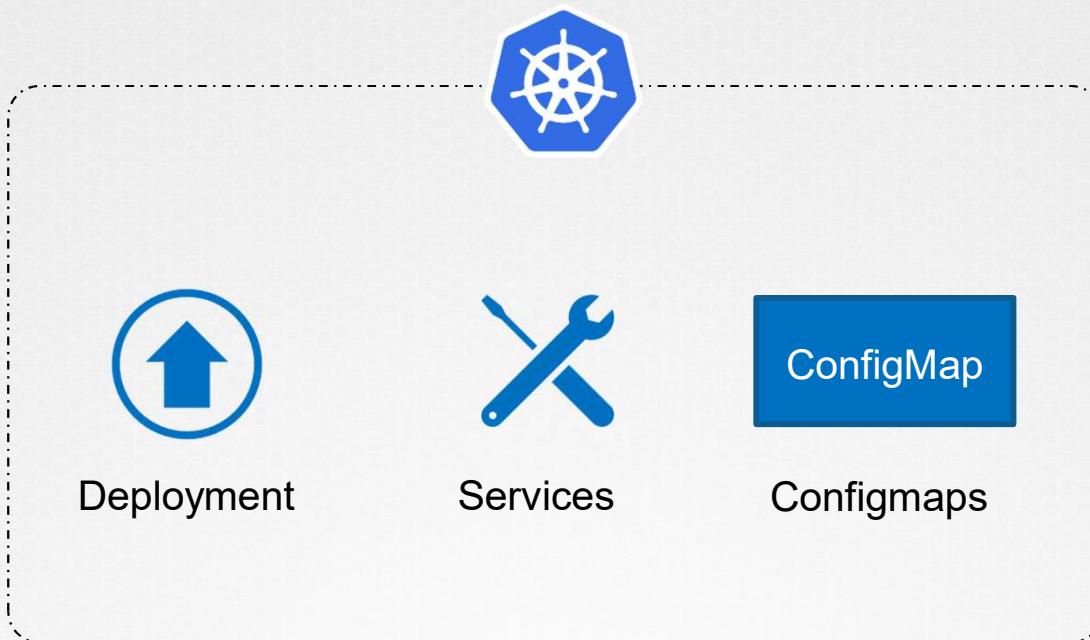
What is GitOps?



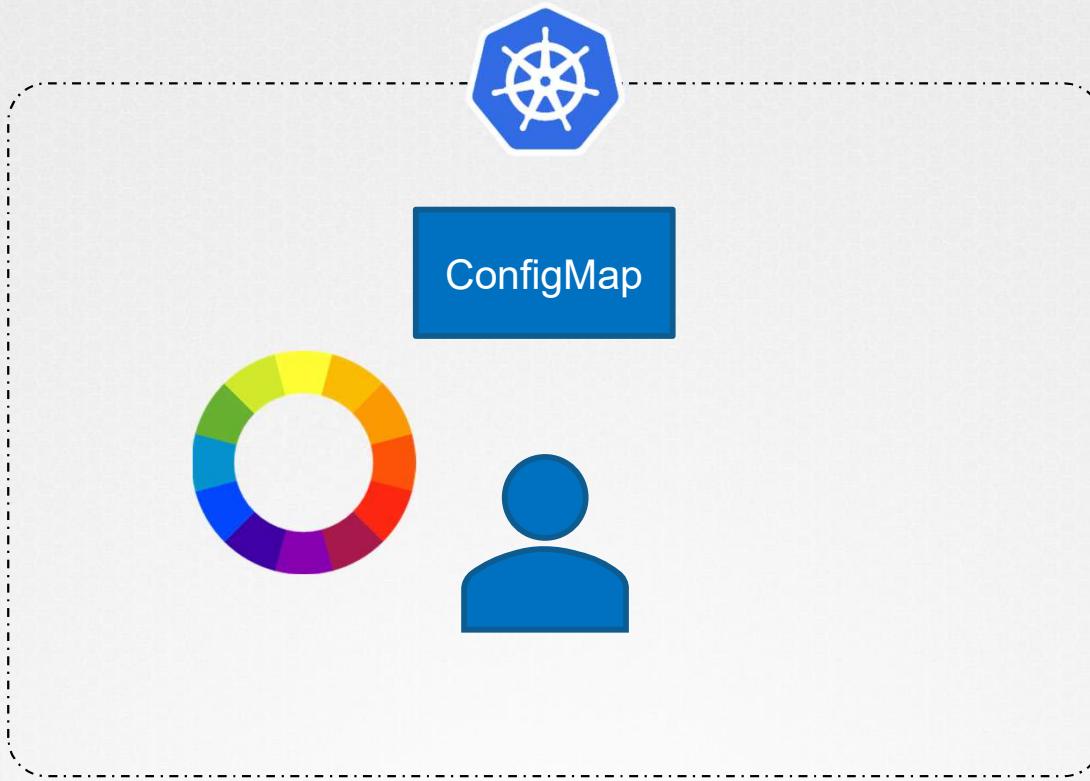
I Analogy



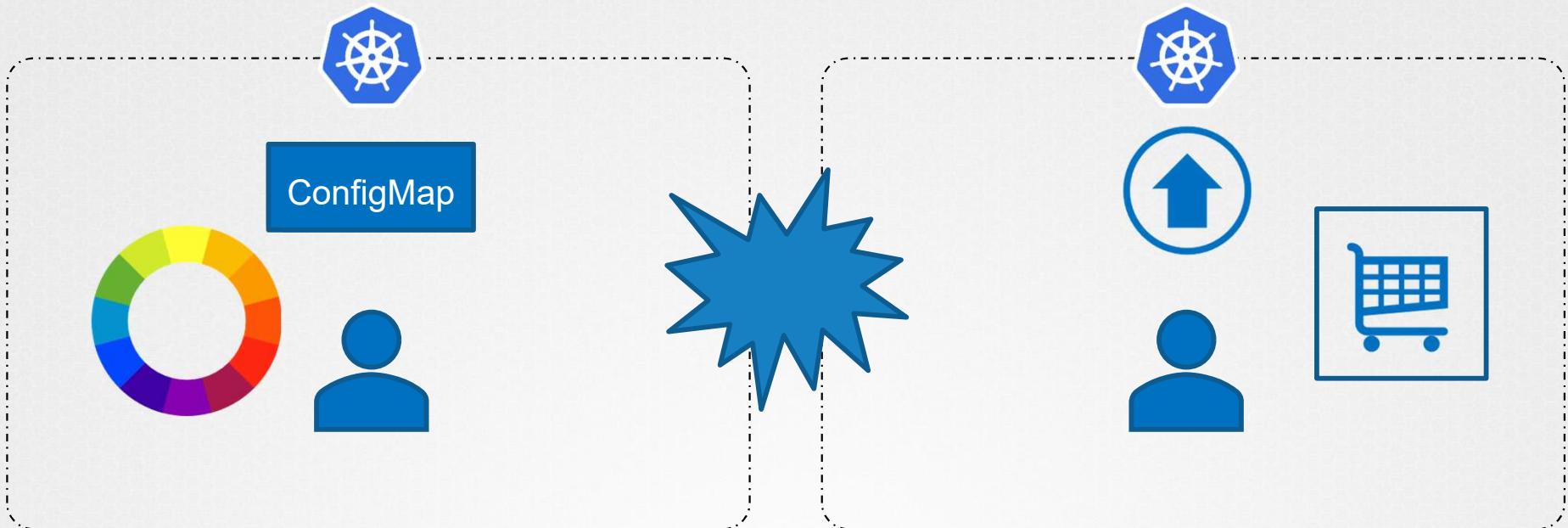
I Analogy



I Analogy



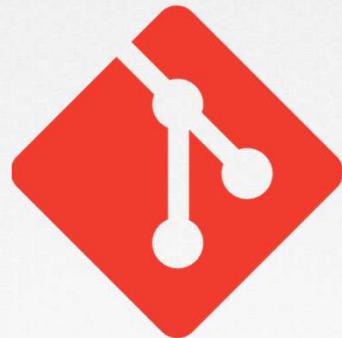
I Analogy



I Analogy

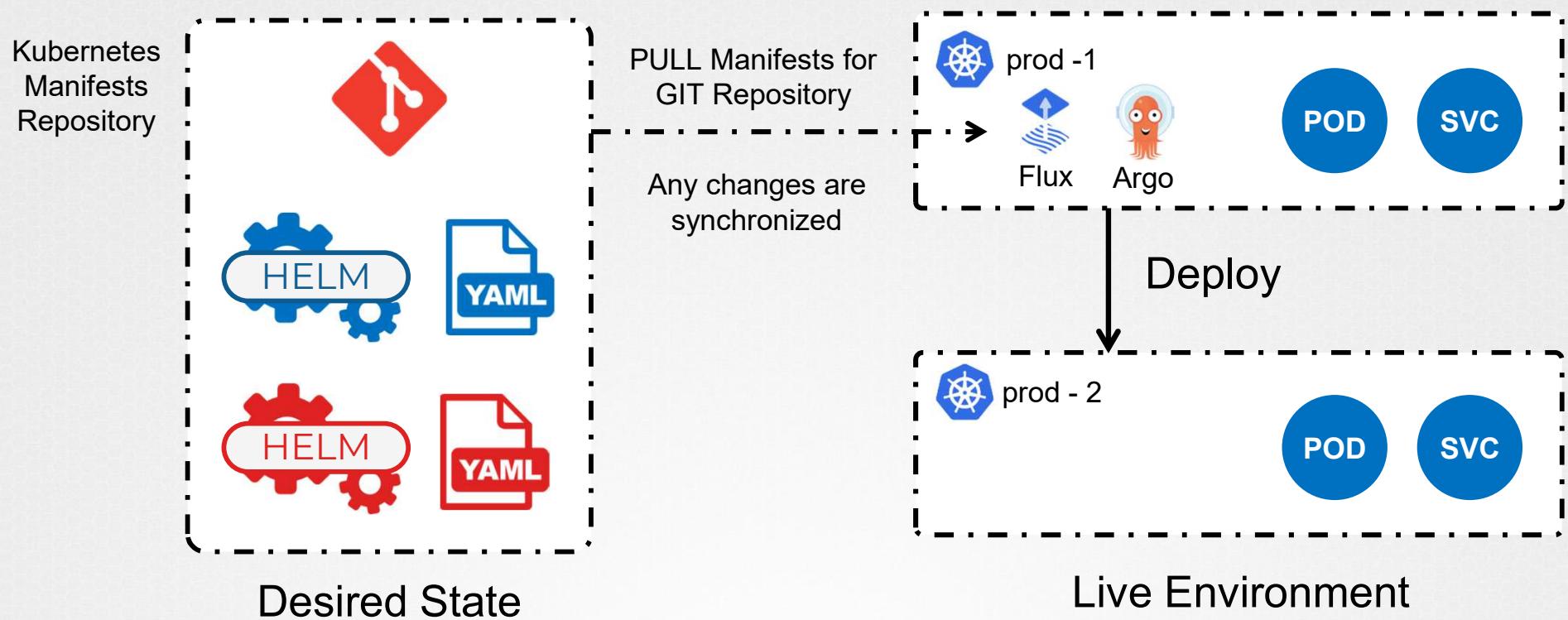


| This is where GitOps comes in

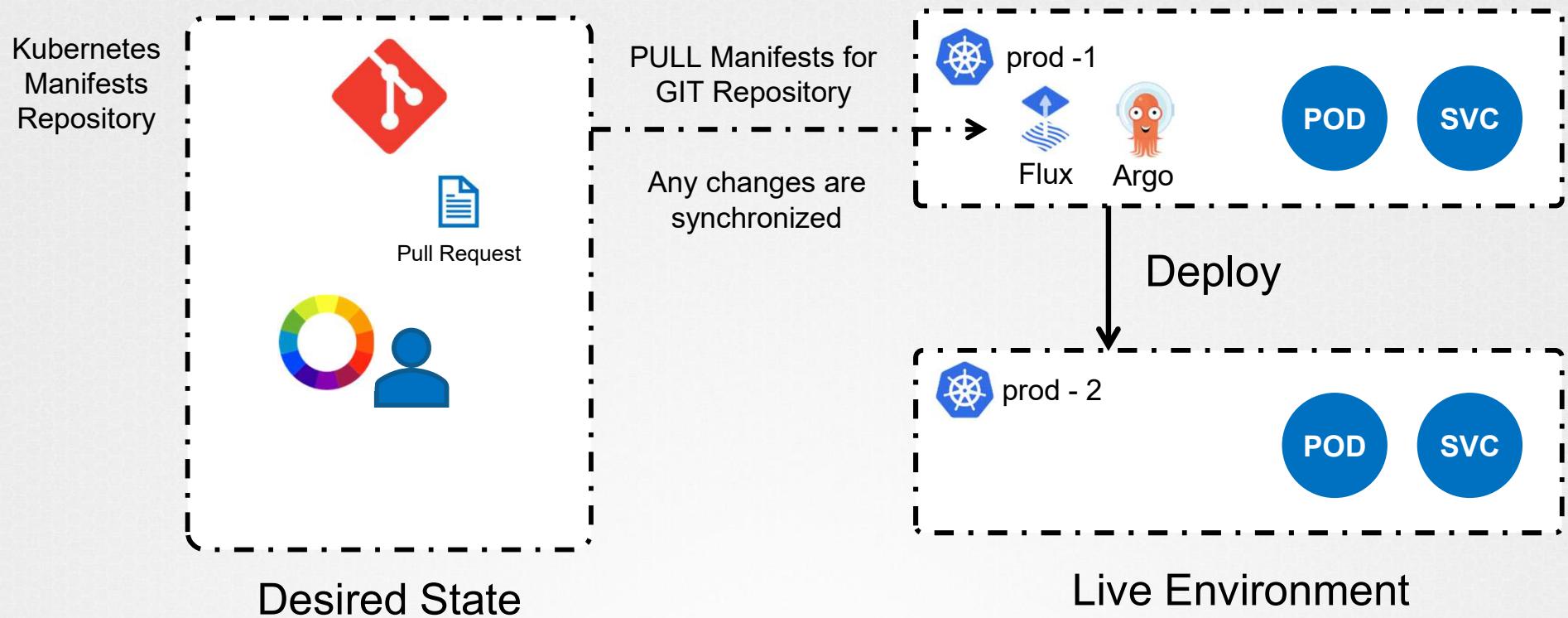


GitOps

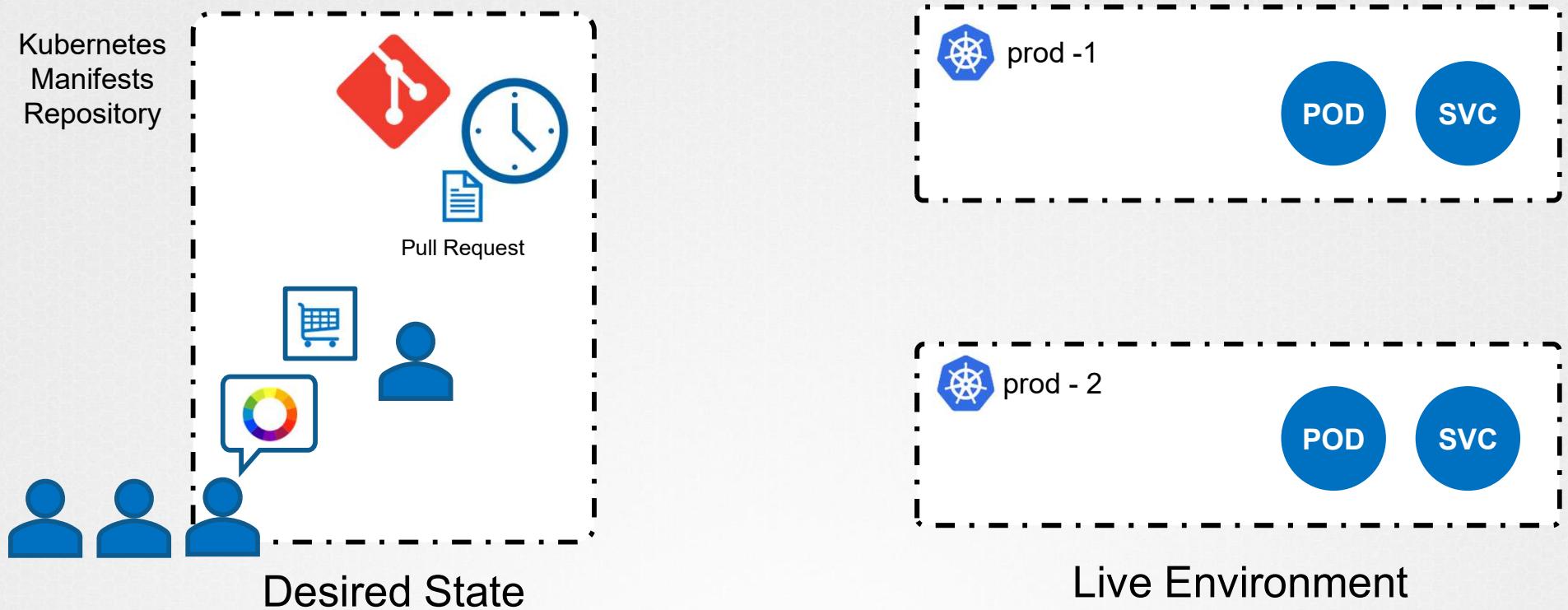
What exactly is GitOps?



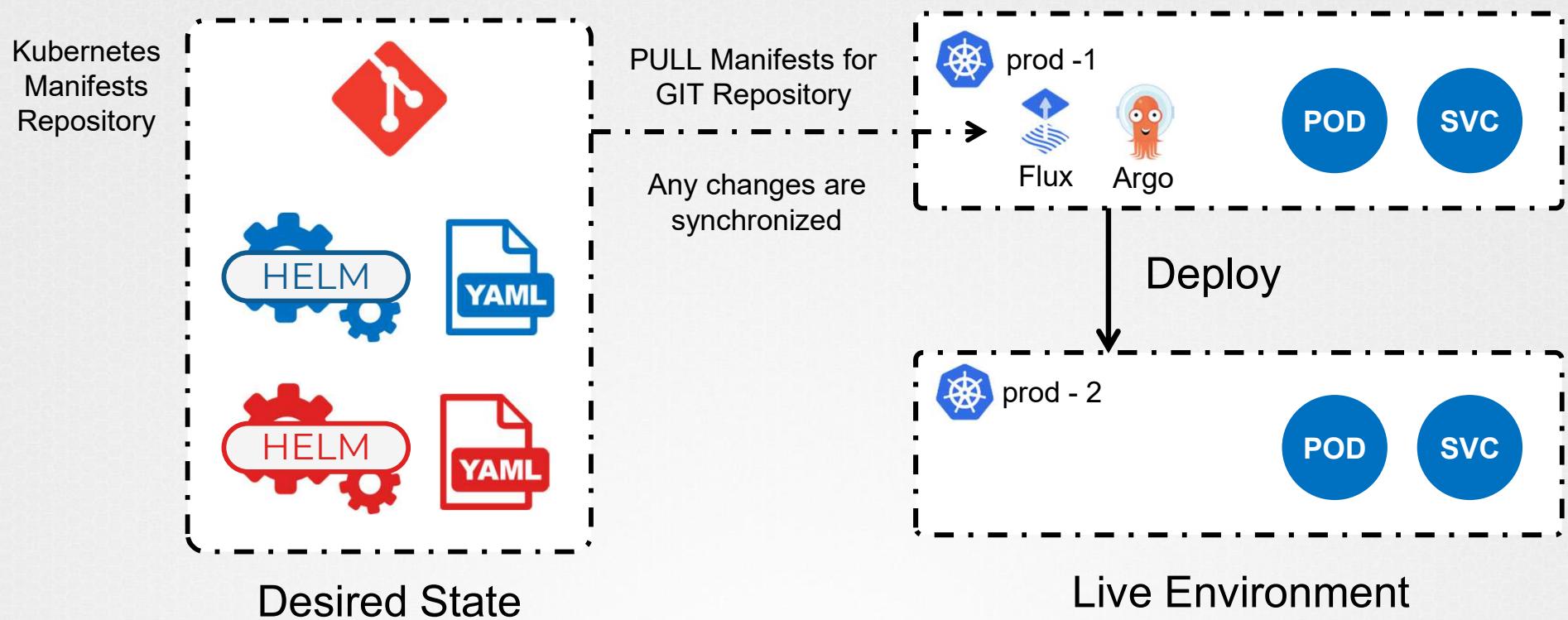
What exactly is GitOps?



What exactly is GitOps?



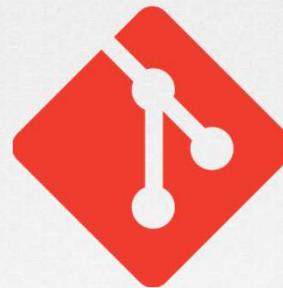
What exactly is GitOps?



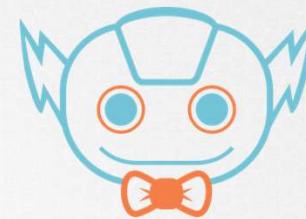
| GitOps Tools and Projects



Flux



Argo



Jenkins X

| GitOps Tools and Projects



Flux

Also FluxCD is a Kubernetes-native Application that syncs Kubernetes state with config files in a Git repository.

- Developed by Weaveworks
- Graduated CNCF Project
- Easy to use
- Focused on CD
- Can only watch one Git repository and one Kubernetes Cluster and Namespace

| GitOps Tools and Projects

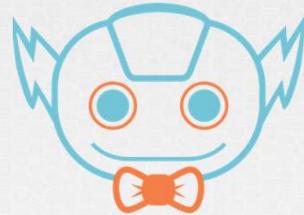


Argo

Also ArgoCD is similar to Flux, it's a declarative GitOps tool designed for Kubernetes.

- Originally developed by Intuit
- Graduated CNCF Project
- Kubernetes-native
- Uses pull-based Model
- Focused on CD
- Can only monitor multiple Git repository and multiple Kubernetes Namespace

| GitOps Tools and Projects



Jenkins X

A Kubernetes-focused GitOps Tool
that covers the entire CI/CD Process

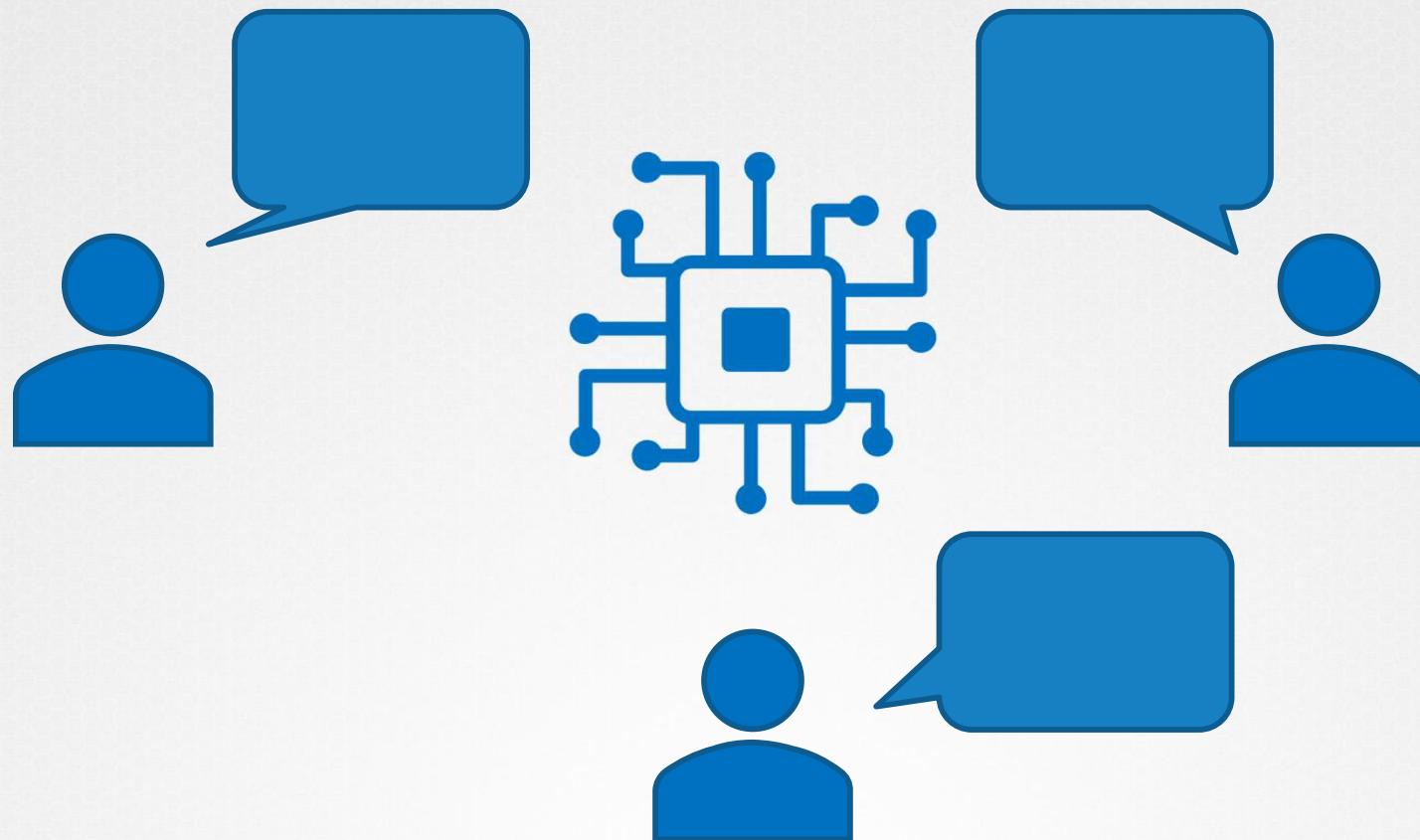
- A collection of other open-source tools.
- Relatively simple to deploy
- More complex than Argo and Flux



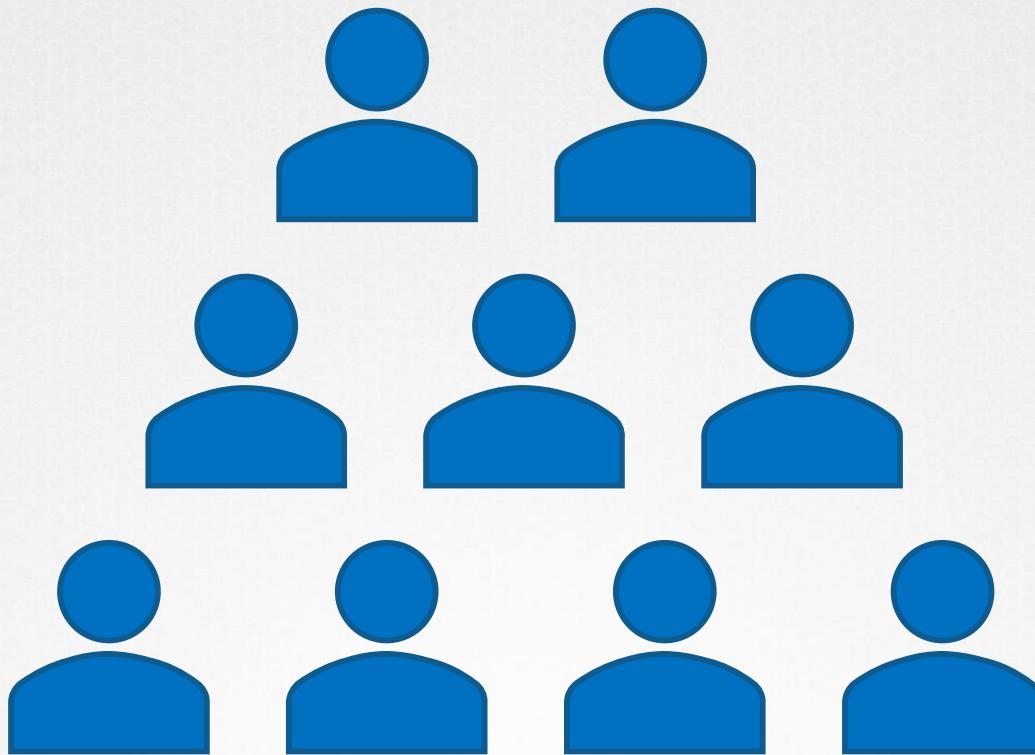
GitOps Principles



I Analogy



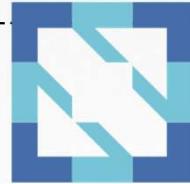
I Governing Body



| Cloud Native Computing Foundation (CNCF)



Kubernetes



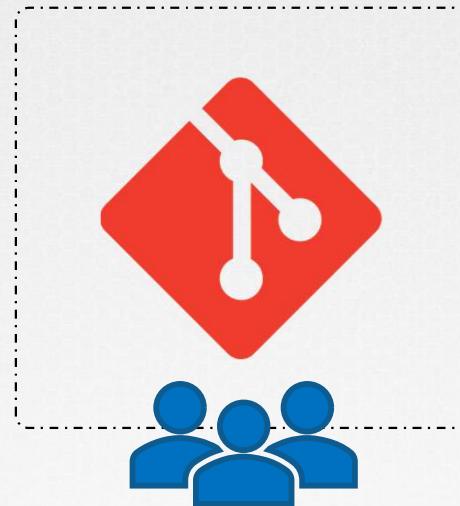
Cloud Native Computing
Foundation

| GitOps Working Group



Cloud Native Computing
Foundation

GitOps Working Group



GitOps Working Group

GitOps Working Group



Cloud Native Computing
Foundation



GitOps Working Group

GitOps Working Group



App Delivery
SIG



GitOps Working Group

GitOps Working Group



Open GitOps Project



Experts

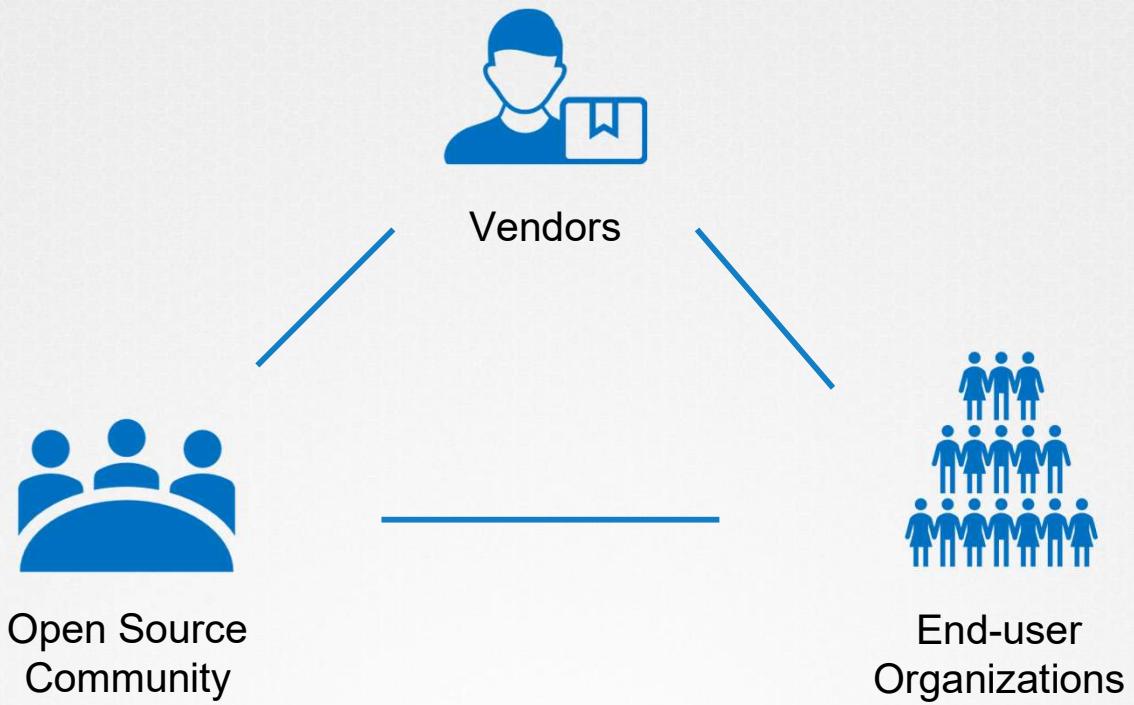


GitOps Working Group



Stakeholders

GitOps Working Group



Four Core Principles



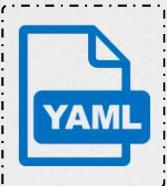
Four Core Principles



A screenshot of a terminal window titled "Terminal". The command \$ cat deployment.yaml is run, displaying the following YAML configuration:

```
$ cat deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 3
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
```

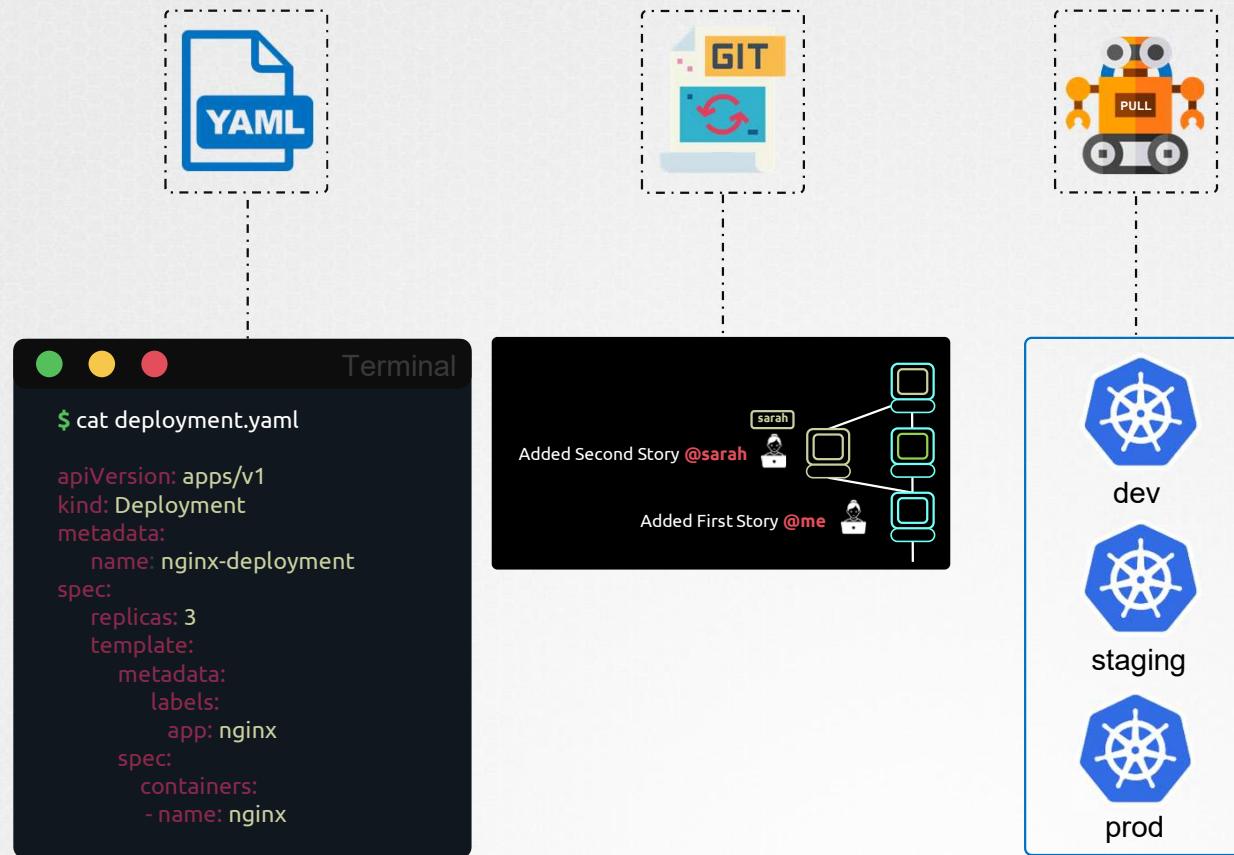
Four Core Principles



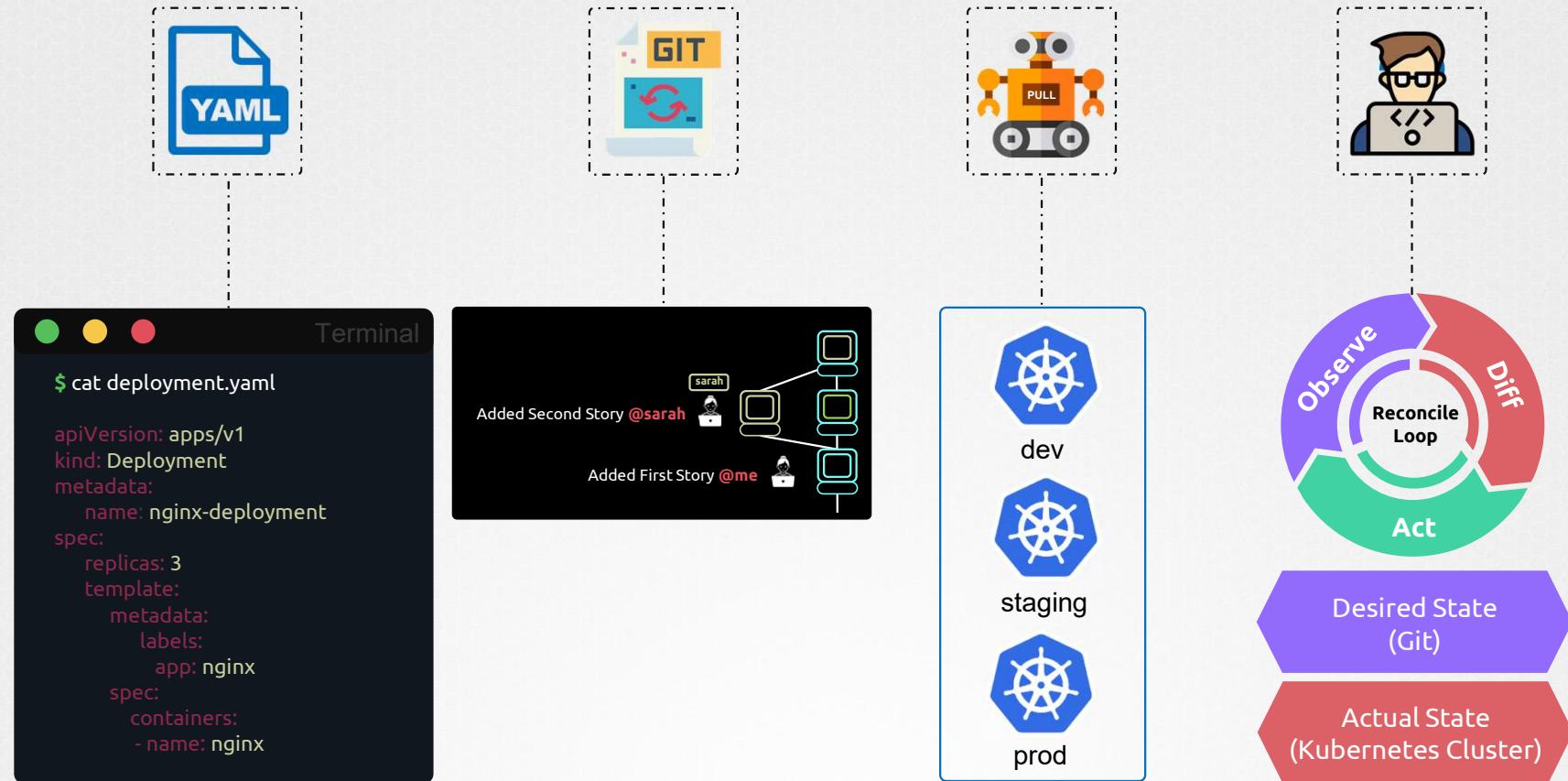
```
Terminal
$ cat deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 3
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
```



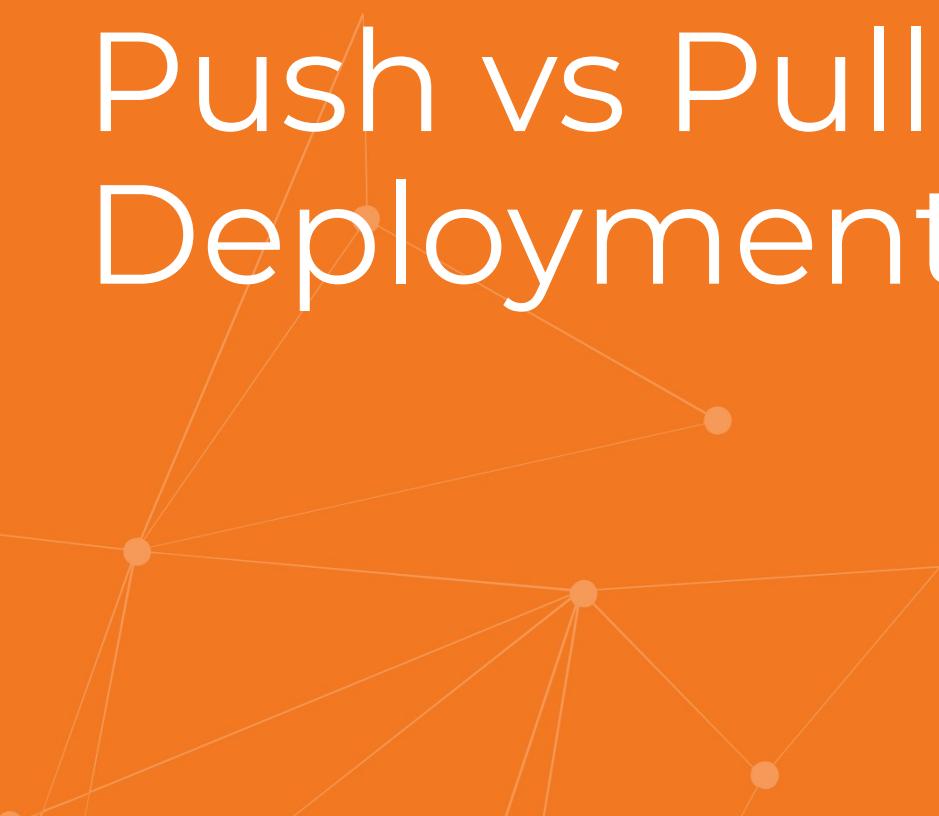
Four Core Principles



Four Core Principles



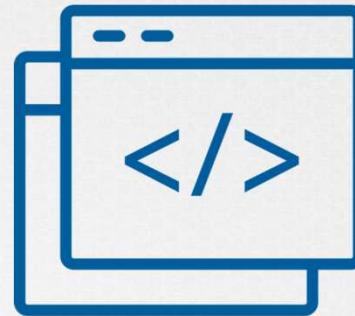
Push vs Pull-based Deployments



|Analogy

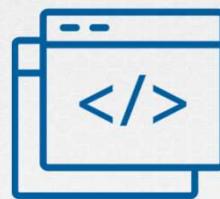


IAnalogy



New Version
Application

Analogy



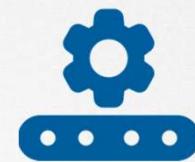
New Version Application



Building

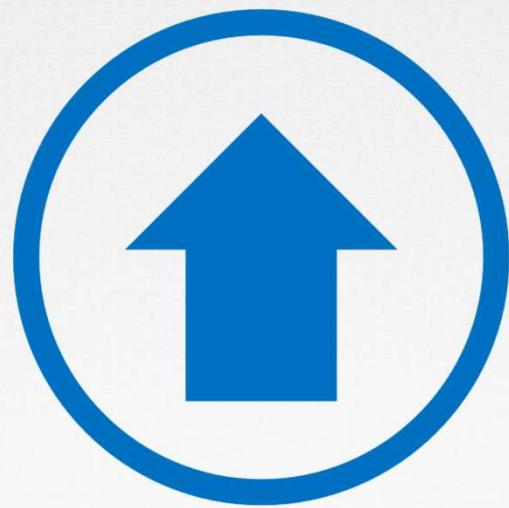


Testing



Production

IAnalogy



Deployment

Analogy



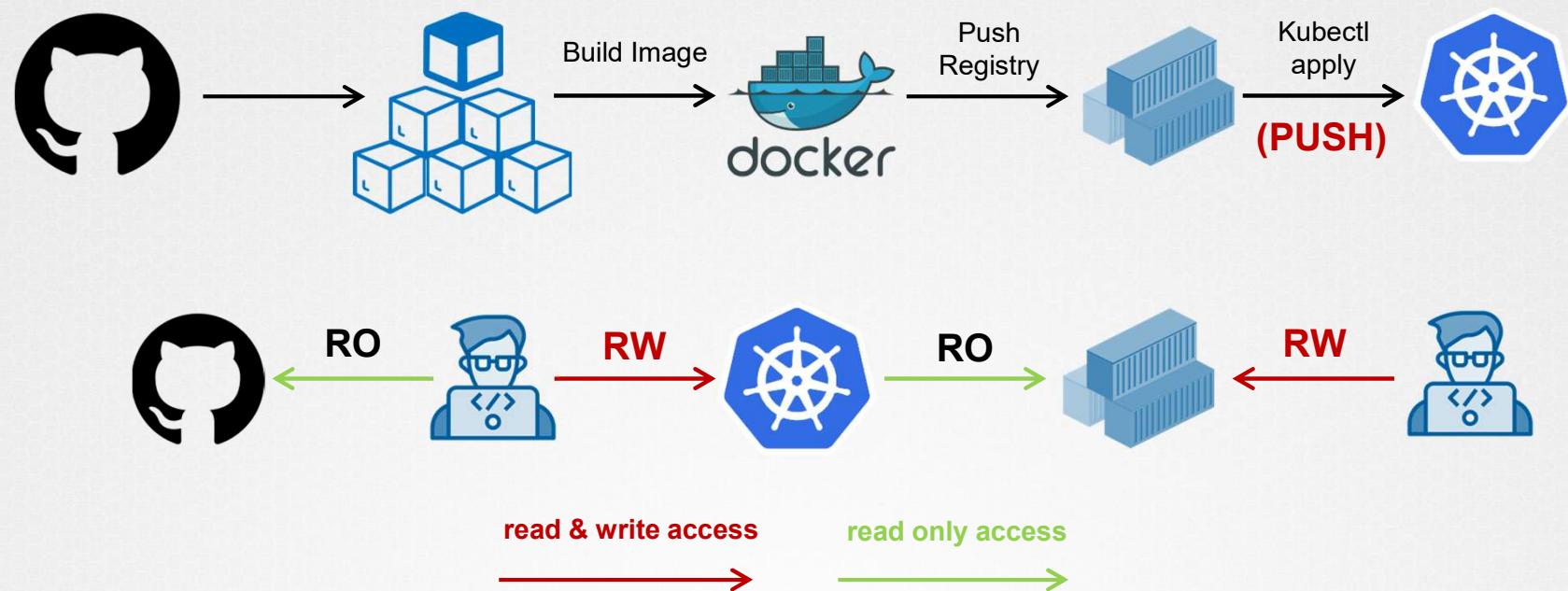
Pull-based

Analogy

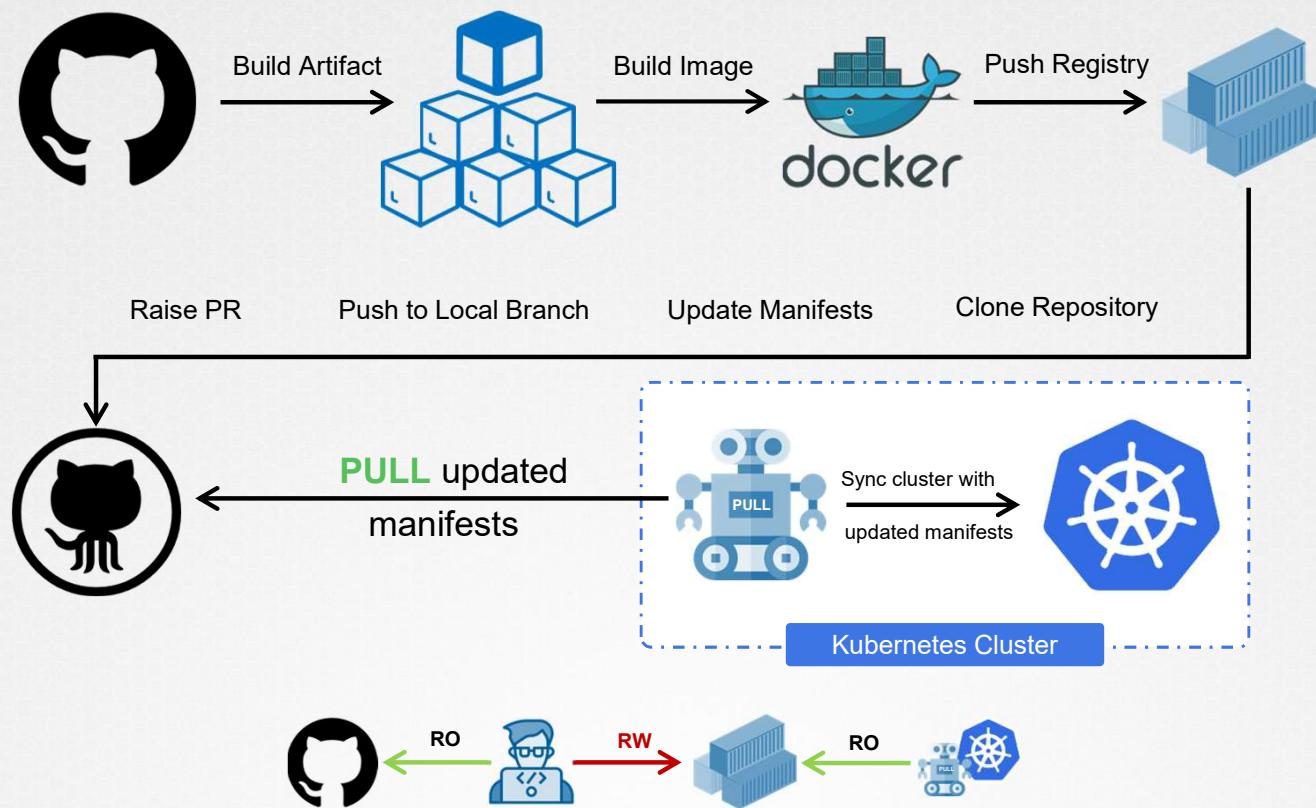


Push-based

Push-based Deployment



Pull-based Deployment



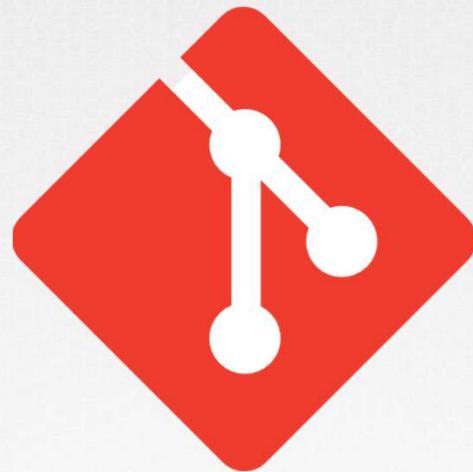
I Main Benefit of Pull Approach



CI/CD with GitOps

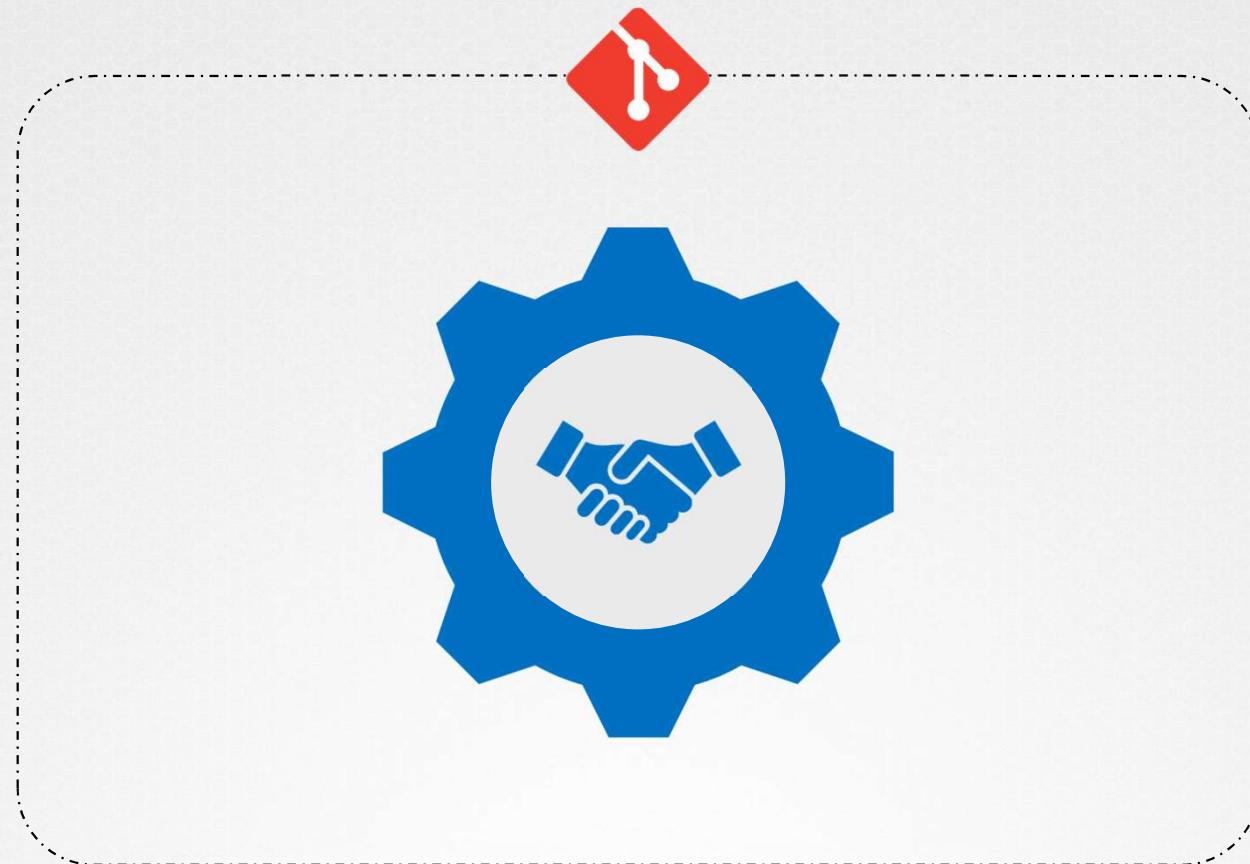


IAnalogy

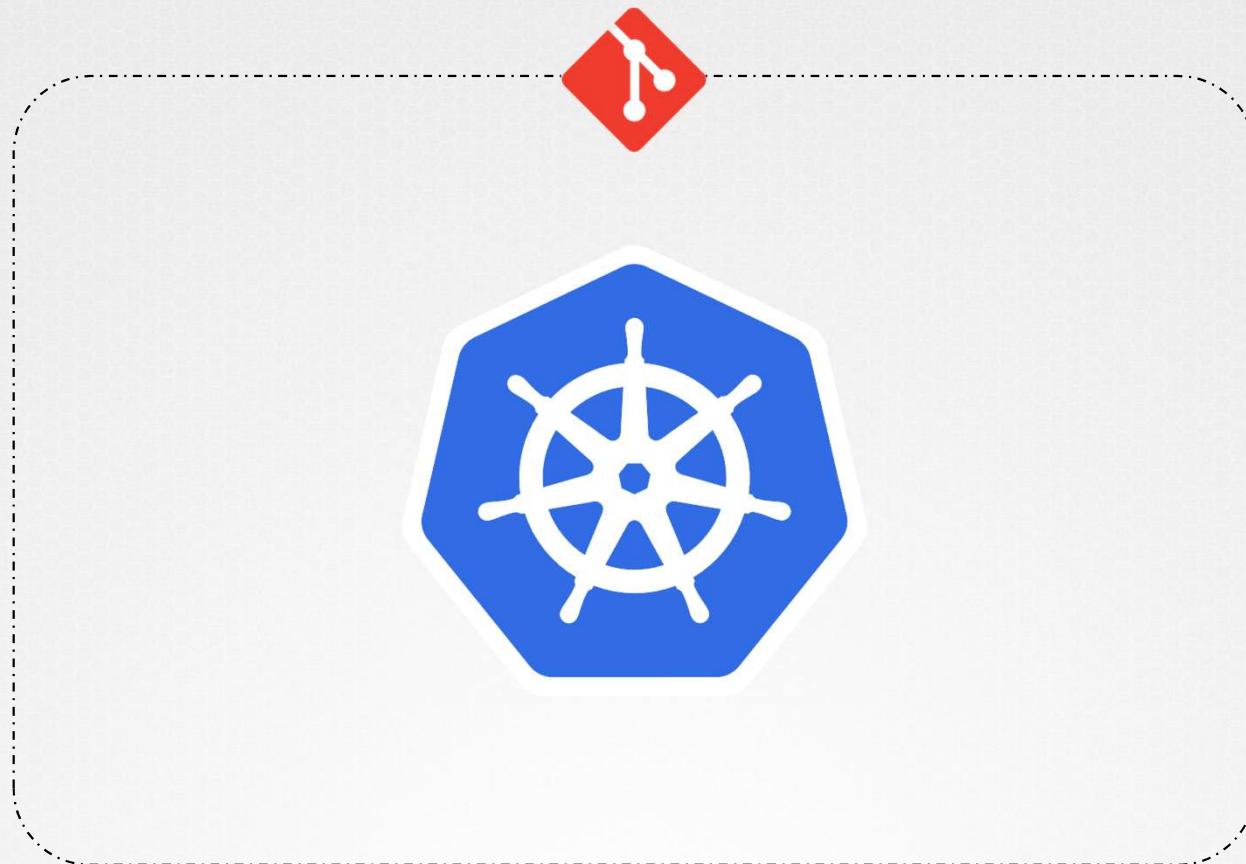


GitOps

Analogy



Analogy



Analogy



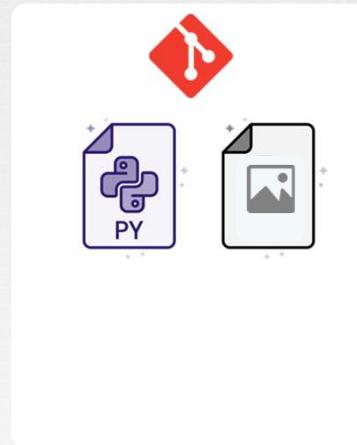
Declarative Configuration



Automated Delivery

In GitOps

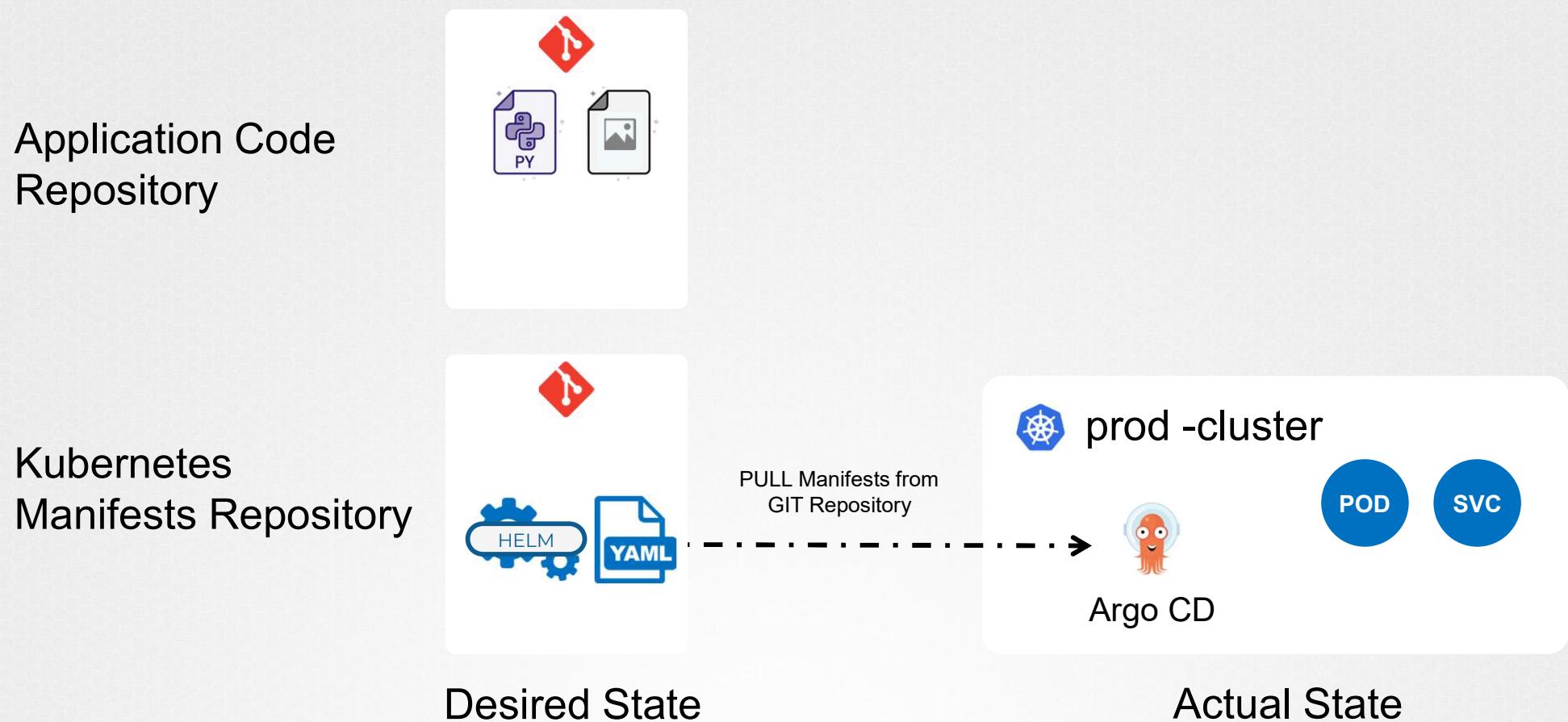
Application
Code Repository



Kubernetes Manifests
Repository



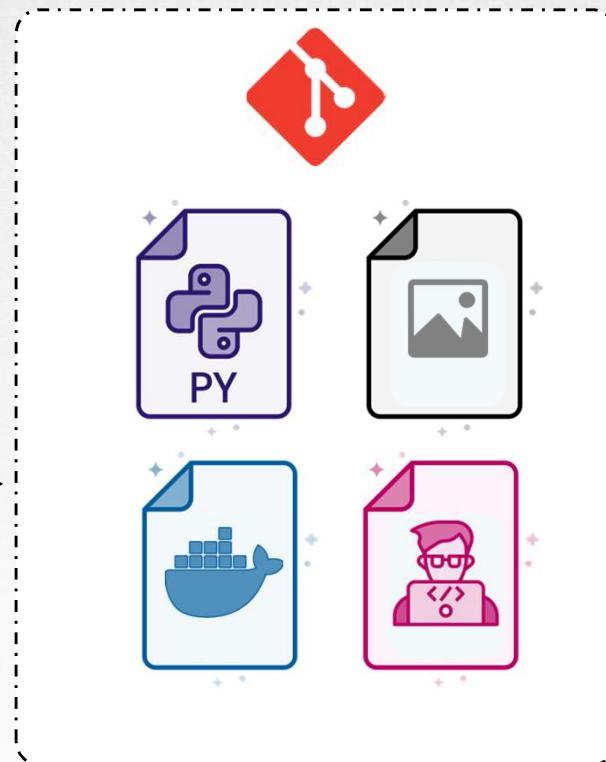
Implementing GitOps



In GitOps

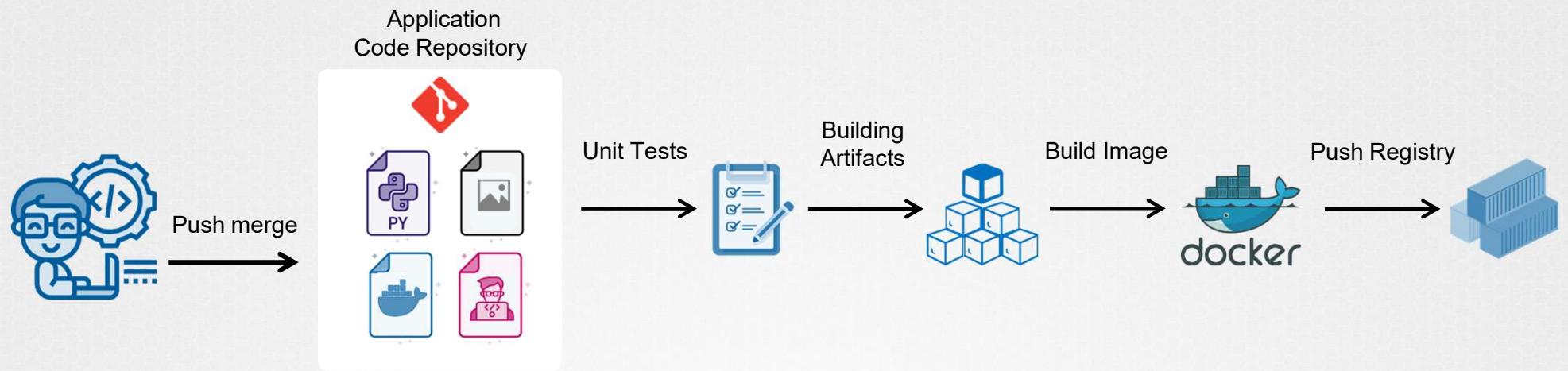


Push Merge

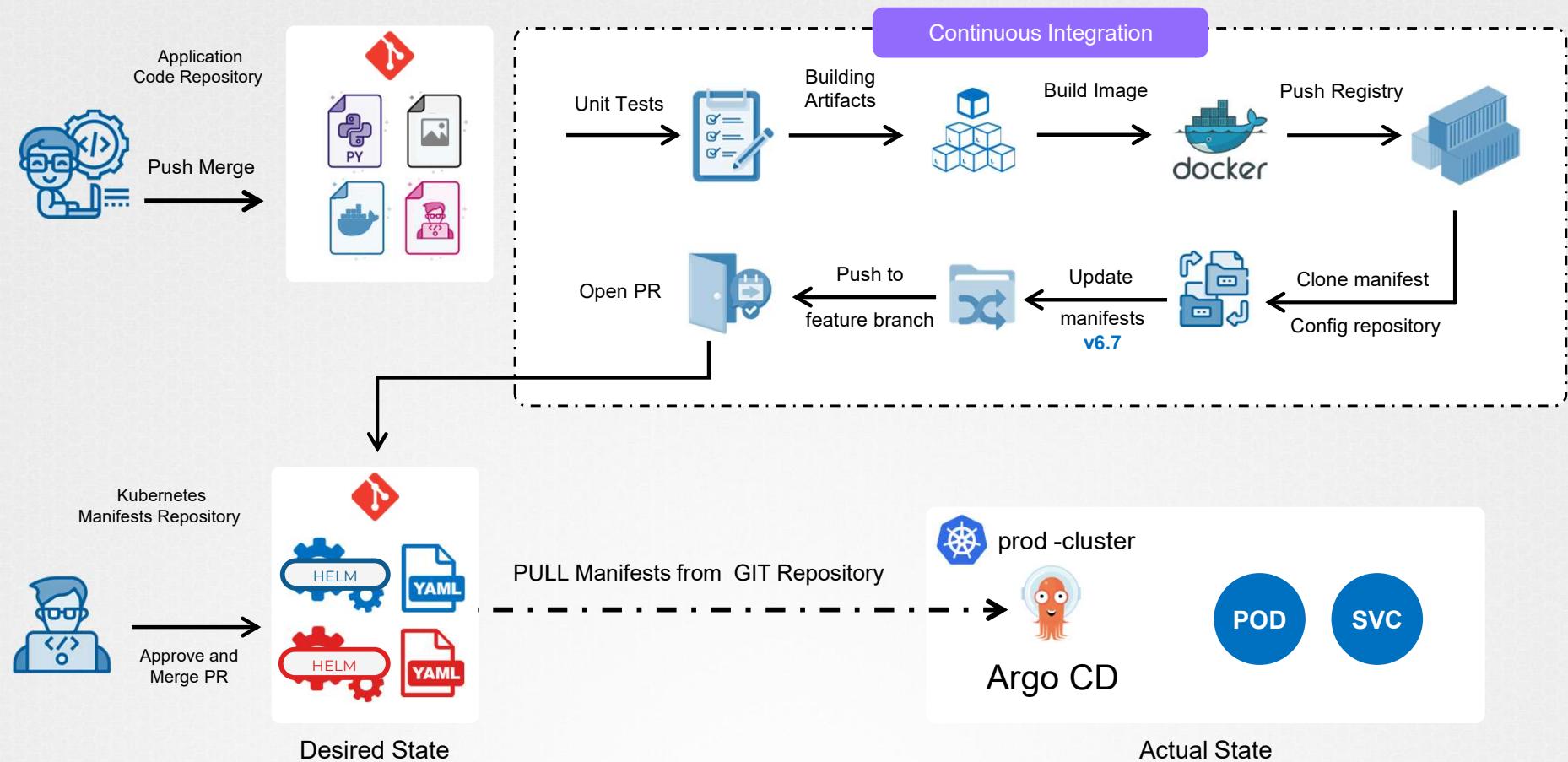


Application Code
Repository

In GitOps



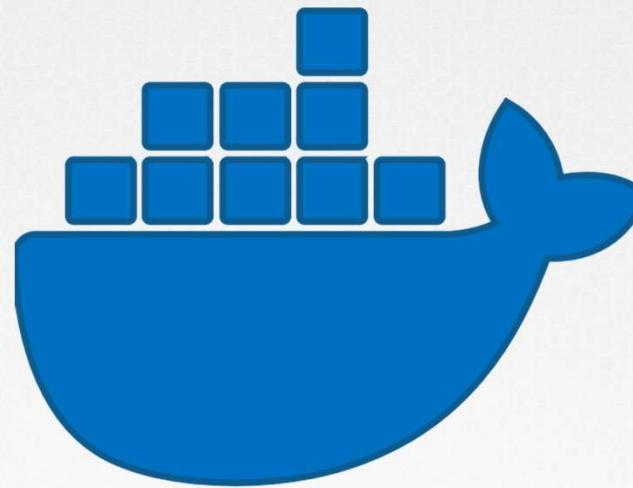
In GitOps



Runtime - CRI



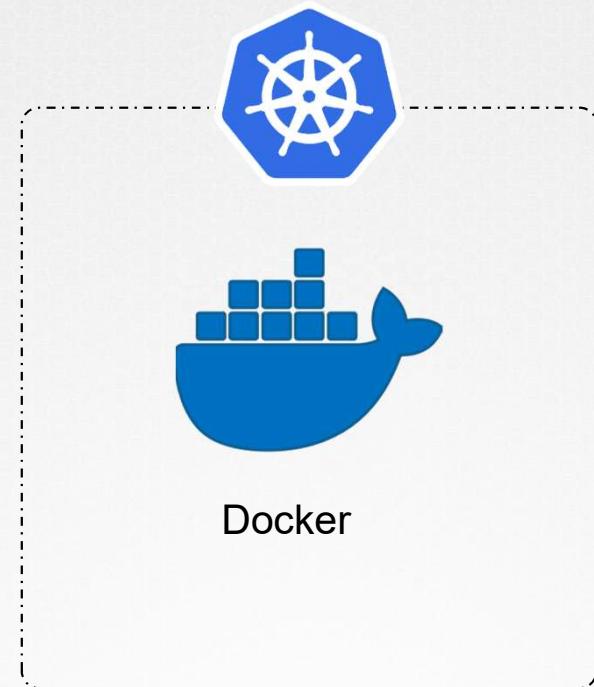
| Analogy



Docker

nj0

| Analogy



nj0

Docker is one line word

nimesha jinarajadasa, 2023-06-02T06:26:11.176

I Analogy



Rkt



Containerd

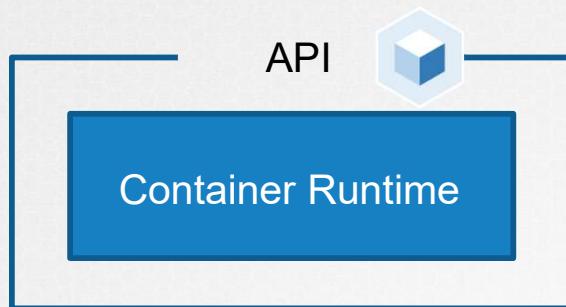


Container Runtime Interface



nj0

Container Runtime Interface (CRI)



Slide 179

- nj0** remove the dashed line named container
cos, containers run on top of container runtime
nimesha jinarajadasa, 2023-06-02T06:35:27.549
- JDO 0** Done
John Rhey Dayang, 2023-06-03T06:14:16.510

nj0

| Container Runtime Interface (CRI)

Container Runtime



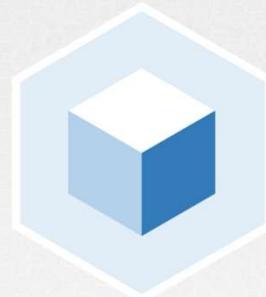
Slide 180

nj0

same here remove the container dashed line. container runs on top of container run time

nimesha jinarajadasa, 2023-06-02T06:36:29.676

Container Runtime Interface (CRI)



The CRI defines the gRPC Protocol that the Kubernetes Kubelet uses to interact with container runtimes



Images

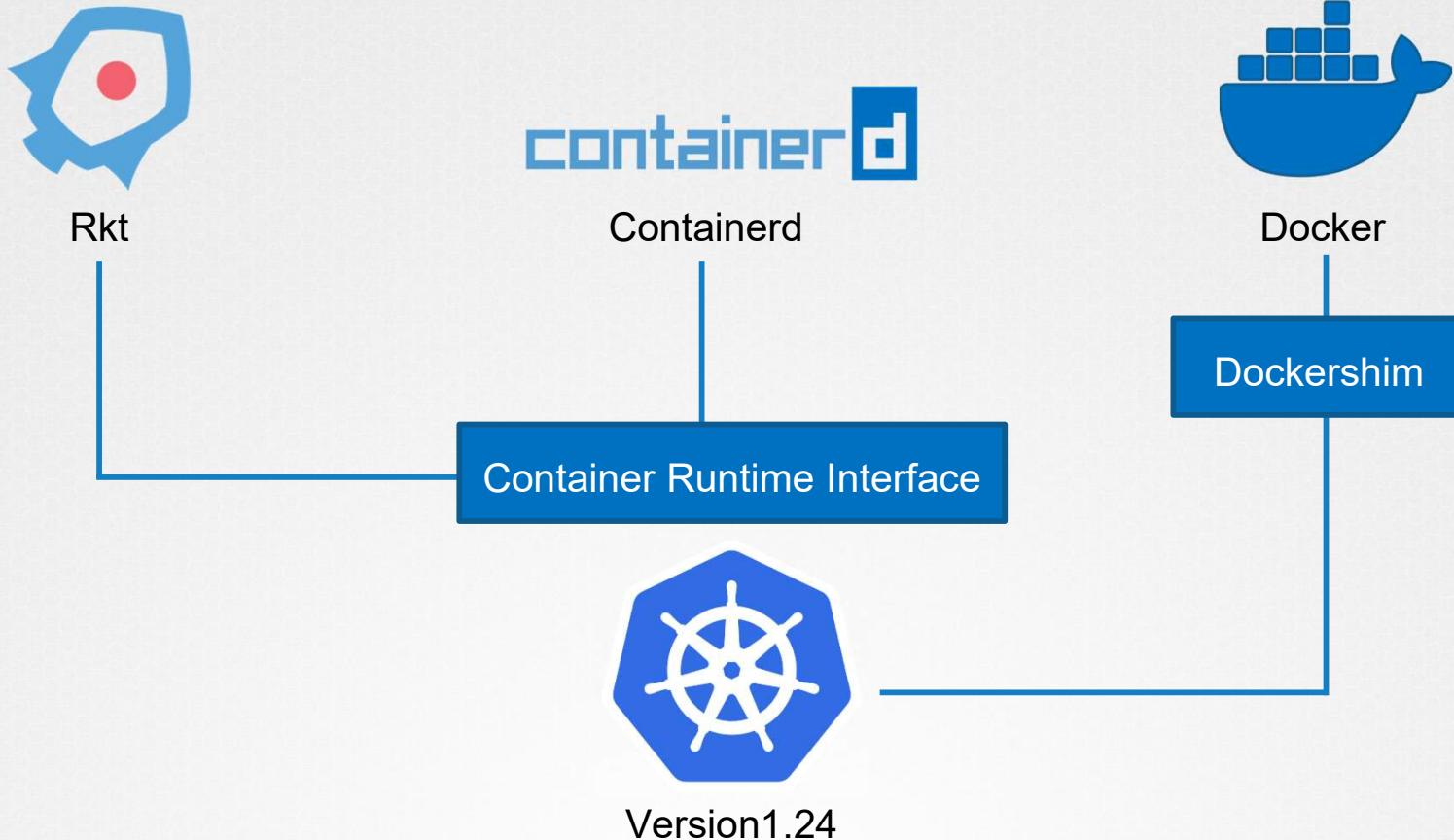


Container

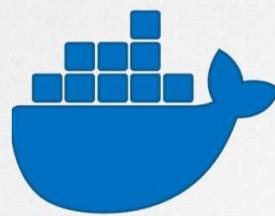


Networking

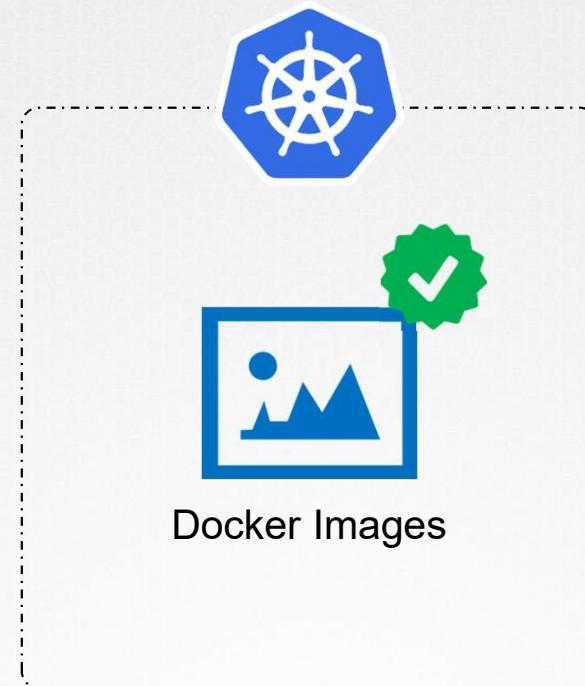
I Analogy



Docker Removal



Docker



Open Container Initiative



containerd



cri-o

