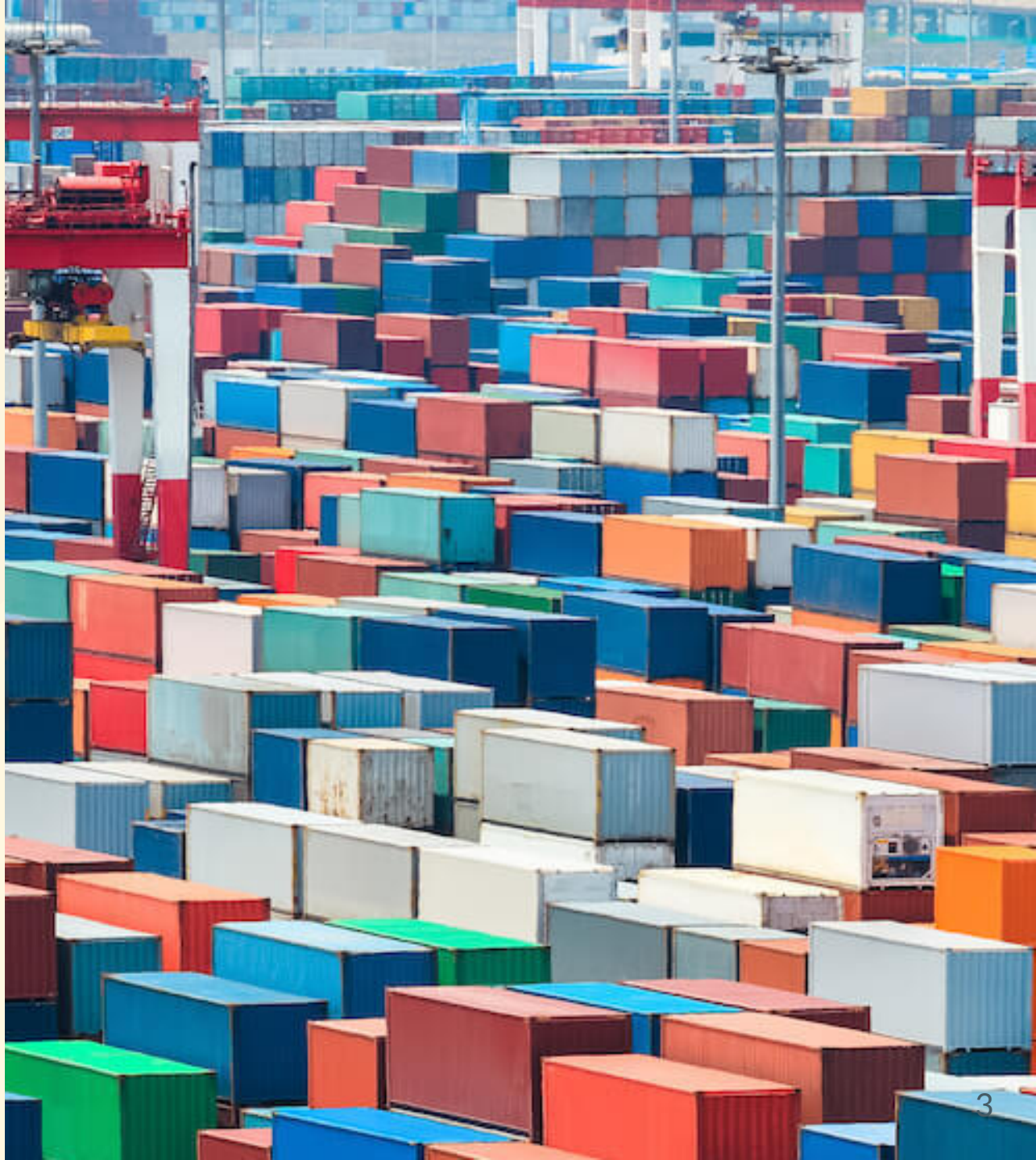
Kubernetes 101

Kubernetes 101

김영우 (Youngwoo Kim)
Data Labs, ICT기술센터, SK Telecom

What is Kubernetes?

- Kubernetes is an open source **container orchestration engine** for automating deployment, scaling, and management of containerized applications. The open source project is hosted by the Cloud Native Computing Foundation



History of Kubernetes

- 2003-2004: Birth of the Borg System
- 2014: Google Introduces Kubernetes
- 2015: The year of Kube v1.0 & CNCF
- 2017: The Year of Enterprise Adoption & Support
- 2018: ……
- 2019: Kubernetes v1.16.2 released (Oct. 2019)

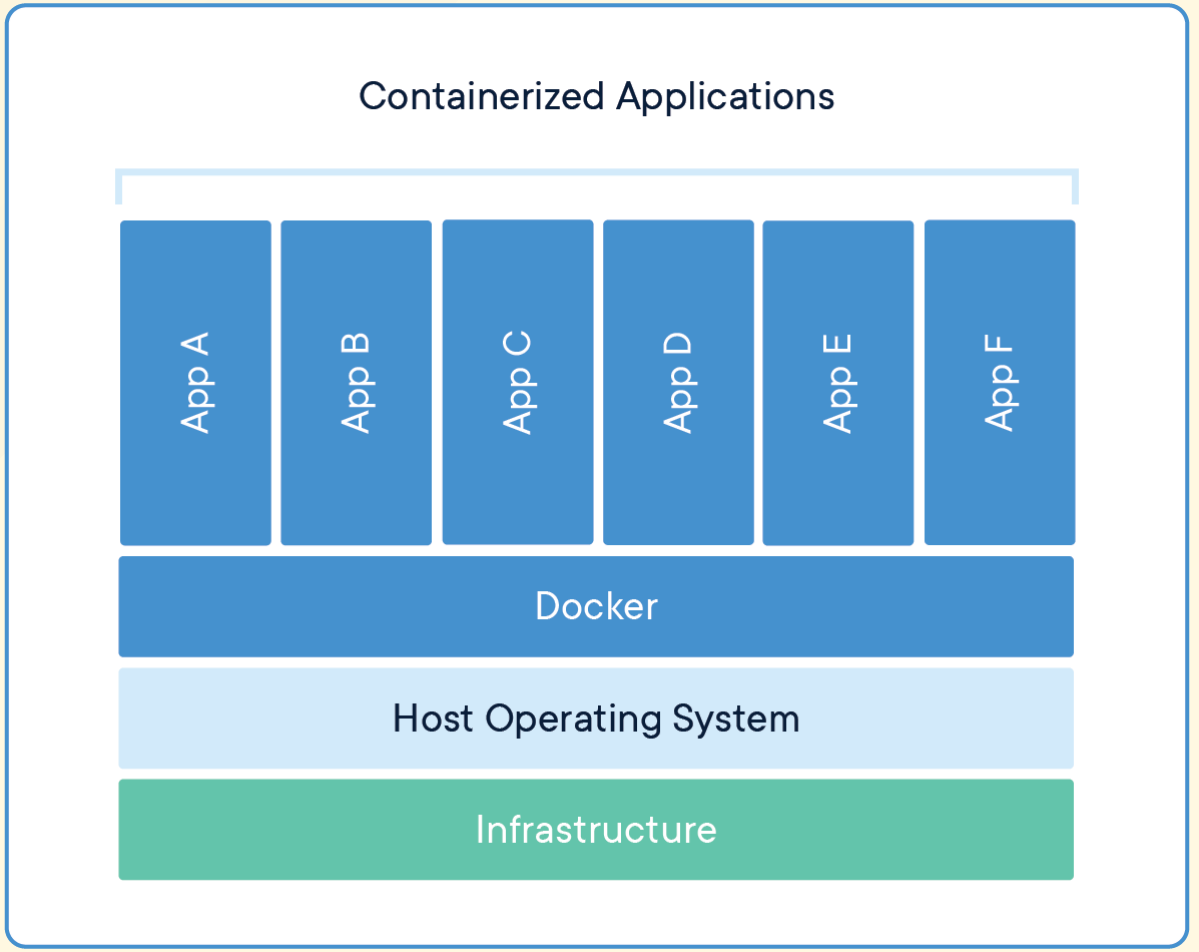
https://blog.risingstack.com/the-history-of-kubernetes/

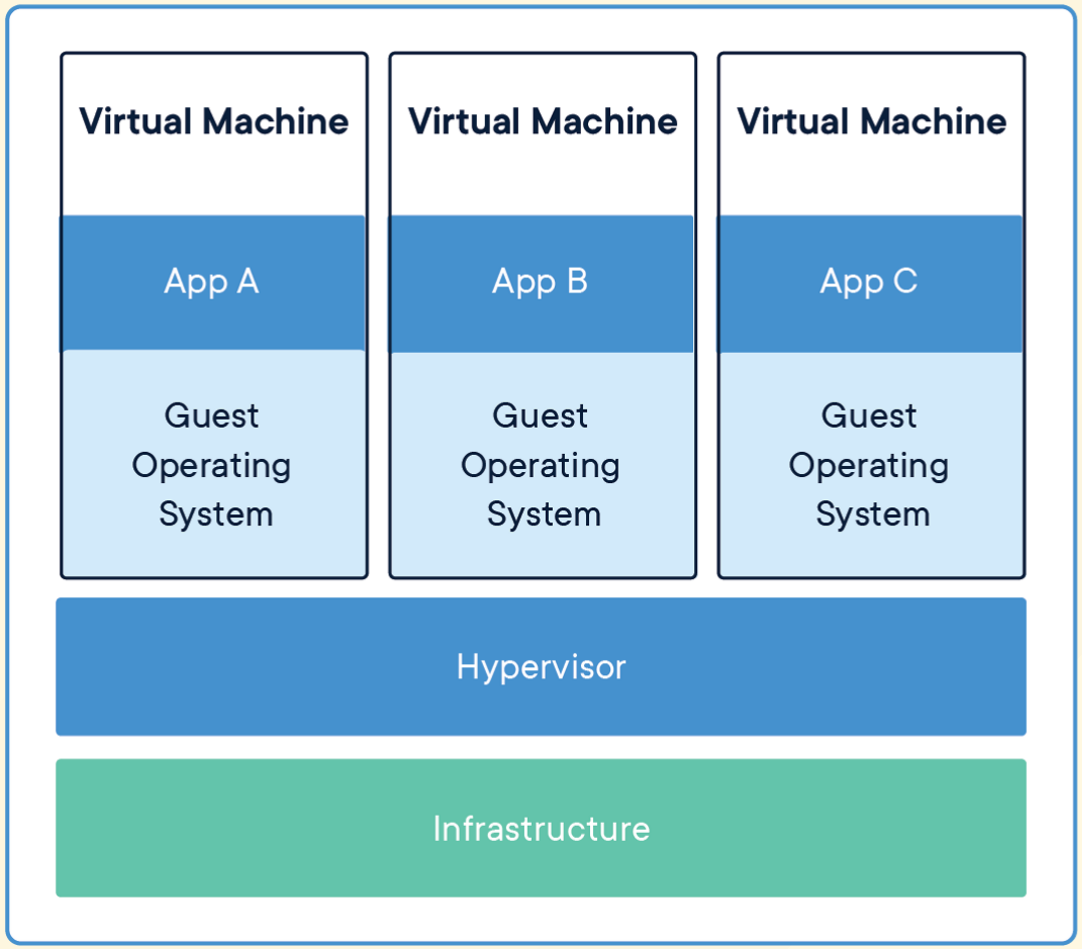
What is Container?

- Containers are a method of virtualization that packages an application's code, configurations, and dependencies into building blocks for consistency, efficiency, productivity, and version control. This page gathers resources about containers, including technical definitions and comparisons.

https://www.aquasec.com/wiki/display/containers/What+is+a+Container

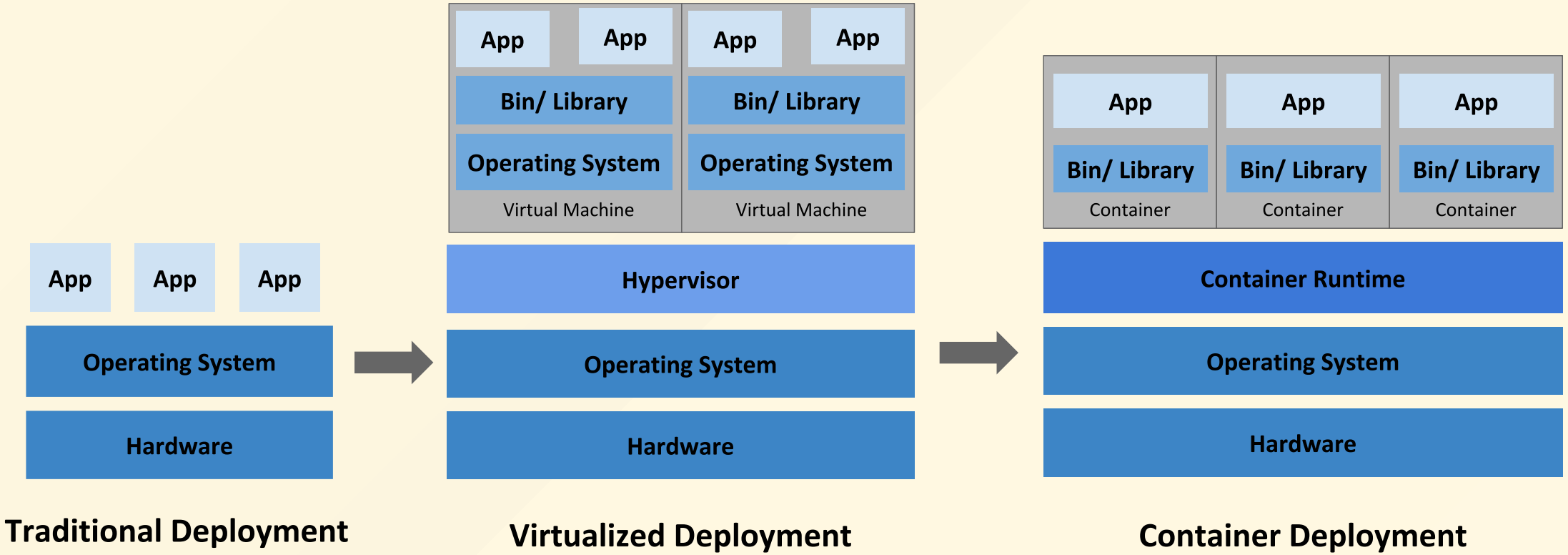
Containers


Containerized Applications
App A
App B
App C
App D
App E
App F
Docker
Host Operating System
Infrastructure



Virtual Machines

What Kubernetes can do



- Service discovery and load balancing
- Storage orchestration
- Automated rollouts and rollbacks
- Automatic bin packing
- Self-healing
- Secret and configuration management

Kubernetes Components

- A cluster is a set of machines, called nodes, that run containerized applications managed by Kubernetes. A cluster has at least one worker node and at least one master node.
- Master components
  - kube-apiserver, etcd, kube-scheduler, kube-controller-manager, cloud-controller-manager
- Node components
  - kubelet, kube-proxy, Container Runtime
- Addons
  - DNS, Dashboard, Container Resource Monitoring, Cluster-level Logging

Kubernetes API

Refer to Kubernetes API reference, https://kubernetes.io/docs/reference/

Kubernetes Objects

- *Kubernetes Objects* are persistent entities in the Kubernetes system.
- A Kubernetes object is a “record of intent”–once you create the object, the Kubernetes system will constantly work to ensure that object exists.
- To work with Kubernetes objects–whether to create, modify, or delete them–you’ll need to use the Kubernetes API. When you use the kubectl command-line interface, for example, the CLI makes the necessary Kubernetes API calls for you.

Object spec and status

- Every Kubernetes object includes two nested object fields that govern the object's configuration: the object *spec* and the object *status*.
  - E.g., A Kubernetes *Deployment* is an object that can represent an application running on your cluster.

- Describing a Kubernetes Object
  - An example .yaml file that shows the required fields and object spec for a Kubernetes Deployment:

```yaml
apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 2 # tells deployment to run 2 pods matching the template
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.7.9
        ports:
        - containerPort: 80
```

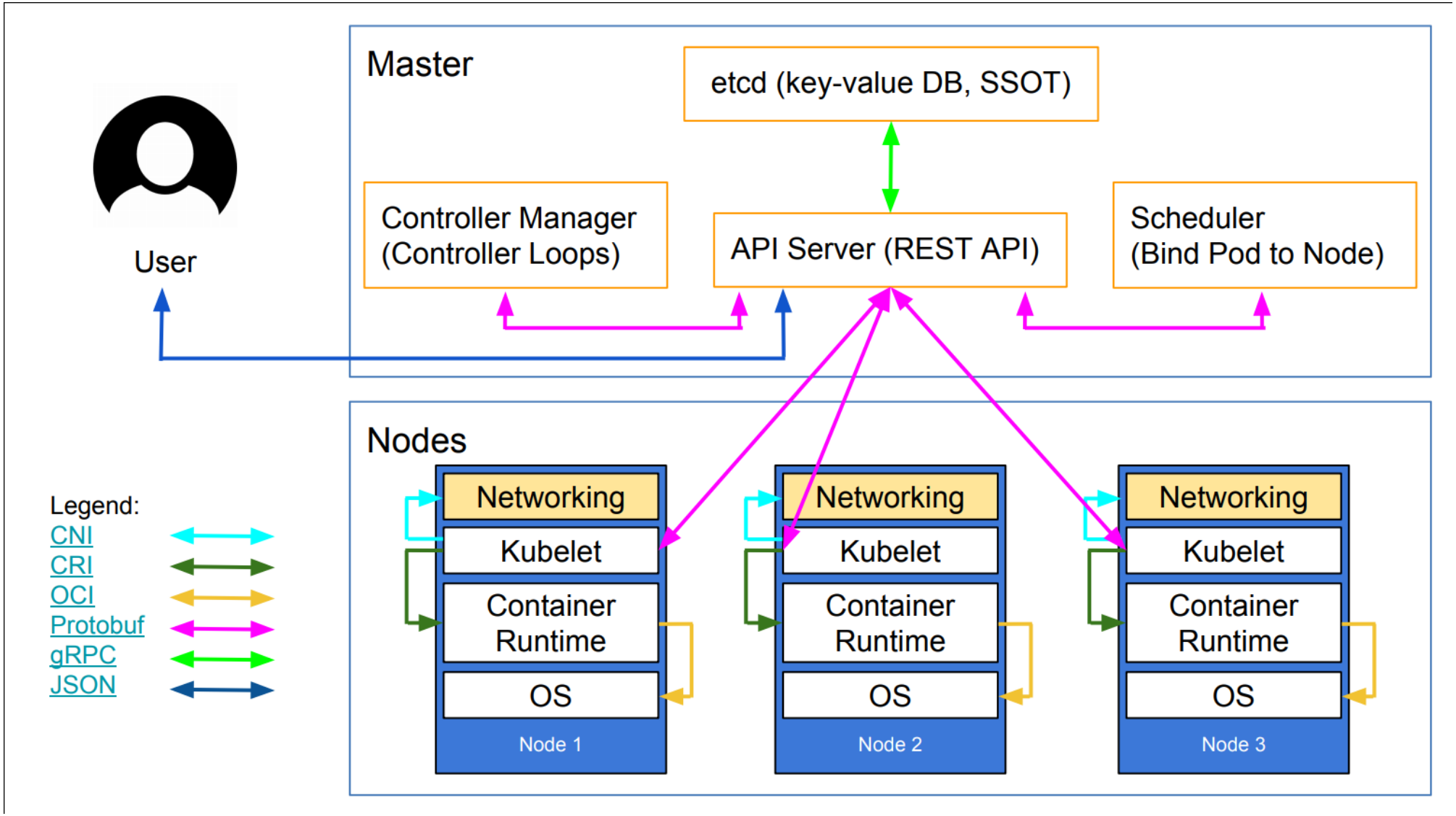
Managing Kubernetes Object

- Kubectl is a command line interface for running commands against Kubernetes clusters.
  - https://kubernetes.io/docs/reference/kubectl/kubectl/

```
$ kubectl version
Client Version: version.Info{Major:"1", Minor:"16", GitVersion:"v1.16.2", ......
Server Version: version.Info{Major:"1", Minor:"16", GitVersion:"v1.16.2", ......
```

Namespaces, Labels and Selectors, Annotations and Field Selectors

Kubernetes Architecture



Getting started with Minikube

- Minukube?
  - Run Kubernetes locally, https://github.com/kubernetes/minikube
- Install Minikube:
  - https://kubernetes.io/docs/tasks/tools/install-minikube/

```
# Install Minikube on MacOS

$ brew install minikube
$ minikube version
minikube version: v1.5.2
commit: 792dbf92a1de583fcee76f8791cff12e0c9440ad
$ minikube start
$ minikube status
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
```

Kubernetes Fundamentals

- Pod
- Controllers
  - ReplicaSet
  - ReplicationController
  - Deployments
  - StatefulSets
  - DaemonSet
  - Garbage Collection
  - TTL Controller for Finished Resources
  - Jobs - Run to Completion
  - CronJob

Services, Load Balancing, and Networking

Storage

Hands-on Labs

How to run a simple Hello World Node.js app on Kubernetes

- Prerequisites
  - Minikube
  - Node.js

Hello World Node.js application

- Source code of *Hello World* Node.js application (server.js)

```
var http = require('http');

var handleRequest = function(request, response) {
  console.log('Received request for URL: ' + request.url);
  response.writeHead(200);
  response.end('Hello World!');
};
var www = http.createServer(handleRequest);
www.listen(8080);
```

- Sanity check for the app:

```
$ HOSTNAME=`hostname` node server.js
$ curl http://localhost:8080
```

Dockerize Node.js application

- Build & Push Docker image:

```
$ docker build -t hello-node .

......

Successfully built e43978e8832c
Successfully tagged hello-node:latest

$ docker images | grep hello
hello-node                  latest              e43978e8832c        6 minutes ago       655MB
```

```
$ docker login
Username: youngwookim
Password:
Login Succeeded

$ docker tag hello-node youngwookim/hello-node
$ docker push youngwookim/hello-node
```

Create a Kubernetes Deployment and Service

```
$ kubectl create -f hello_node-deployment.yaml
deployment.apps/hello-node created

$ kubectl get deployments
NAME         READY   UP-TO-DATE   AVAILABLE   AGE
hello-node   1/1     1            1           9m28s
```

```
$ kubectl create -f hello_node-service.yaml
service/hello-node-service created
```

Verify the Node.js app

```
$ minikube service hello-node-service --url
http://IP:PORT
```

Browse http://IP:PORT

Or run via curl:

```
$ curl $(minikube service hello-node-service --url)
```

Cloud Native Landscape

https://landscape.cncf.io/

참고

- https://www.ibm.com/cloud/garage/content/course/kubernetes-101/
- https://github.com/contino/kubernetes-101
- https://www.aquasec.com/wiki/display/containers/Kubernetes+Guide
- https://container.training/
- https://dzone.com/articles/the-complete-kubernetes-collection-tutorials-and-tools

Image Credits

- https://www.docker.com/resources/what-container
- https://selleo.com/blog/kubernetes-101
- https://container.training/kube-halfday.yml.html

FIN