Kubernetes

ผู้ช่วยศาสตราจารย์ ดร.สมเกียรติ โกศลสมบัติ

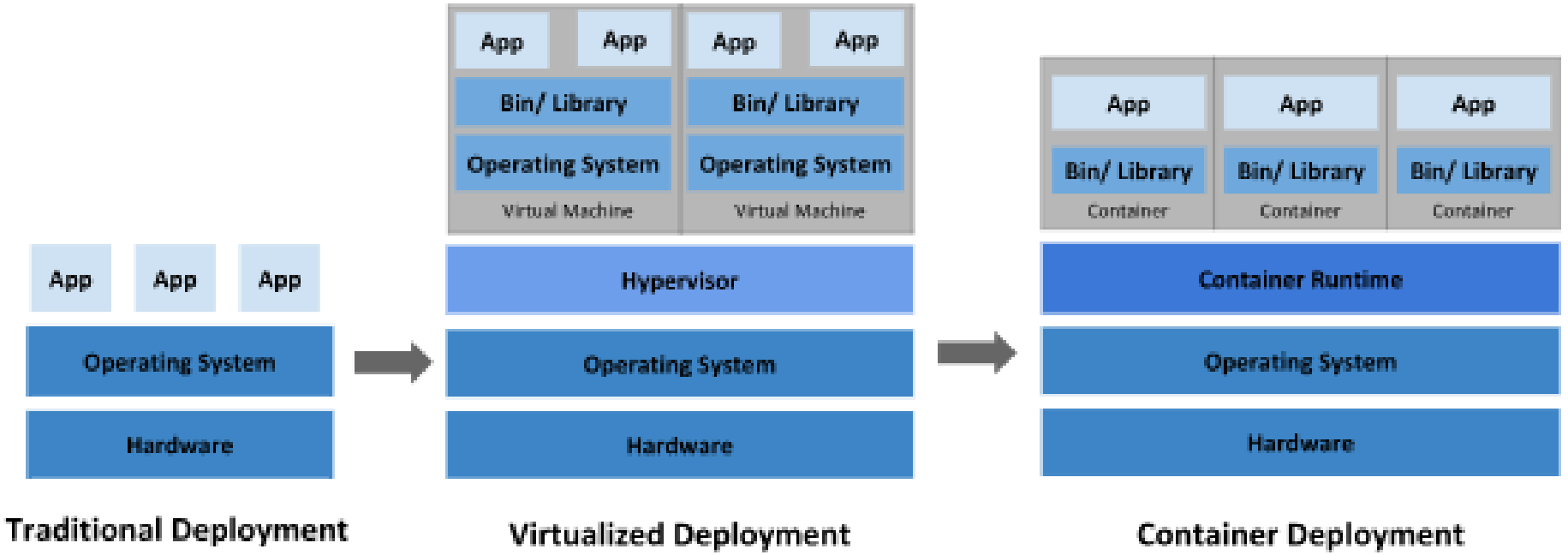
สาขาวิชาวิทยาศาสตร์และนวัตกรรมข้อมูล

วิทยาลัยสหวิทยาการ มหาวิทยาลัยธรรมศาสตร์

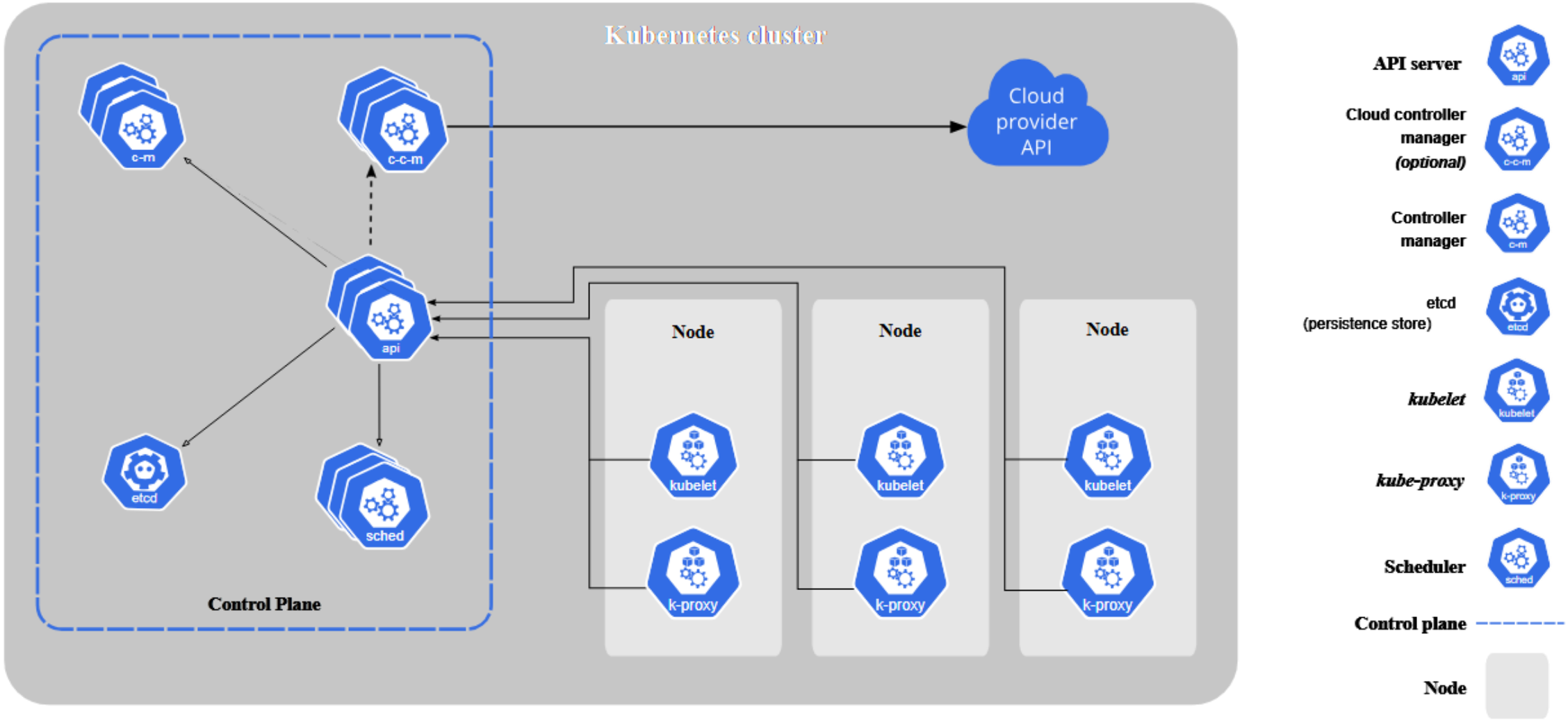
Contents

Introduction to Kubernetes (k8s)

- Kubernetes is a portable, extensible, open source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation.
- The name Kubernetes originates from Greek, meaning helmsman or pilot.
- K8s as an abbreviation results from counting the eight letters between the "K" and the "s".
- Google open-sourced the Kubernetes project in 2014.



Kubernetes Components



Control Plane Components

- kube-apiserver
  - The core component server that exposes the Kubernetes HTTP API
- etcd
  - Consistent and highly-available key value store for all API server data
- kube-scheduler
  - Looks for Pods not yet bound to a node and assigns each Pod to a suitable node.
- kube-controller-manager
  - Runs controllers to implement Kubernetes API behavior.
- cloud-controller-manager (Optional)
  - Integrates with underlying cloud provider(s).

Node Components

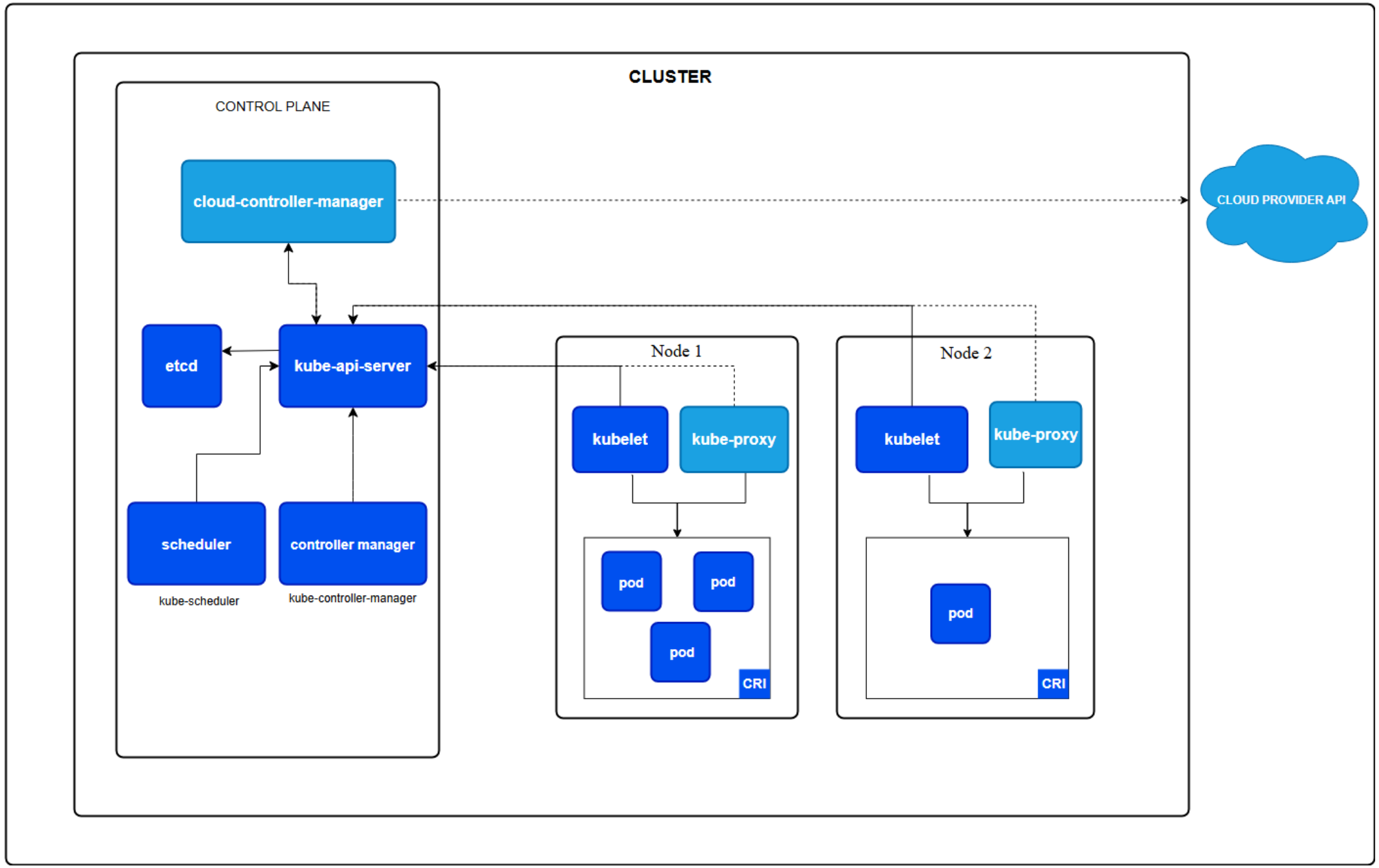
- kubelet
  - Ensures that Pods are running, including their containers.
- Kube-proxy (Optional)
  - Maintains network rules on nodes to implement Services.
- Container runtime
  - Software responsible for running containers. (Pod)

The Kubernetes API

- The Kubernetes API lets you query and manipulate the state of objects in Kubernetes.
- The core of Kubernetes' control plane is the API server and the HTTP API that it exposes.
- Users, the different parts of your cluster, and external components all communicate with one another through the API server.
- The core of Kubernetes' control plane is the API server.
- The API server exposes an HTTP API that lets end users, different parts of your cluster, and external components communicate with one another.
- The Kubernetes API lets you query and manipulate the state of API objects in Kubernetes (for example: Pods, Namespaces, ConfigMaps, and Events).

Cluster Architecture

- A Kubernetes cluster consists of a control plane plus a set of worker machines, called nodes, that run containerized applications.
- Every cluster needs at least one worker node in order to run Pods.
- The worker node(s) host the Pods that are the components of the application workload.
- The control plane manages the worker nodes and the Pods in the cluster.
- In production environments, the control plane usually runs across multiple computers and a cluster usually runs multiple nodes, providing fault-tolerance and high availability.



Kubernetes Cluster components

Control Plane Components

- Control plane components can be run on any machine in the cluster.
- However, for simplicity, setup scripts typically start all control plane components on the same machine, and do not run user containers on this machine.
  - kube-apiserver
  - etcd
  - kube-scheduler
  - kube-controller-manager
  - cloud-controller-manager (Optional)

kube-apiserver

- The API server is a component of the Kubernetes control plane that exposes the Kubernetes API.
- The API server is the front end for the Kubernetes control plane.
- The main implementation of a Kubernetes API server is kube-apiserver.
- kube-apiserver is designed to scale horizontally—that is, it scales by deploying more instances.
- We can run several instances of kube-apiserver and balance traffic between those instance.

etcd

- Consistent and highly-available key value store used as Kubernetes' backing store for all cluster data.
- It is a distributed key-value store that serves as the backbone of the cluster's data storage.
- It is responsible for storing all cluster-related data, including configuration, state, and metadata.

kube-scheduler

- Control plane component that watches for newly created Pods with no assigned node and selects a node for them to run on.
- Factors taken into account for scheduling decisions include: individual and collective resource requirements, hardware/software/policy constraints, affinity and anti-affinity specifications, data locality, inter-workload interference, and deadlines.
- kube-scheduler is a control plane component responsible for assigning newly created Pods to suitable Nodes in a Kubernetes cluster.
- It ensures that each Pod is scheduled on a Node that meets its requirements.

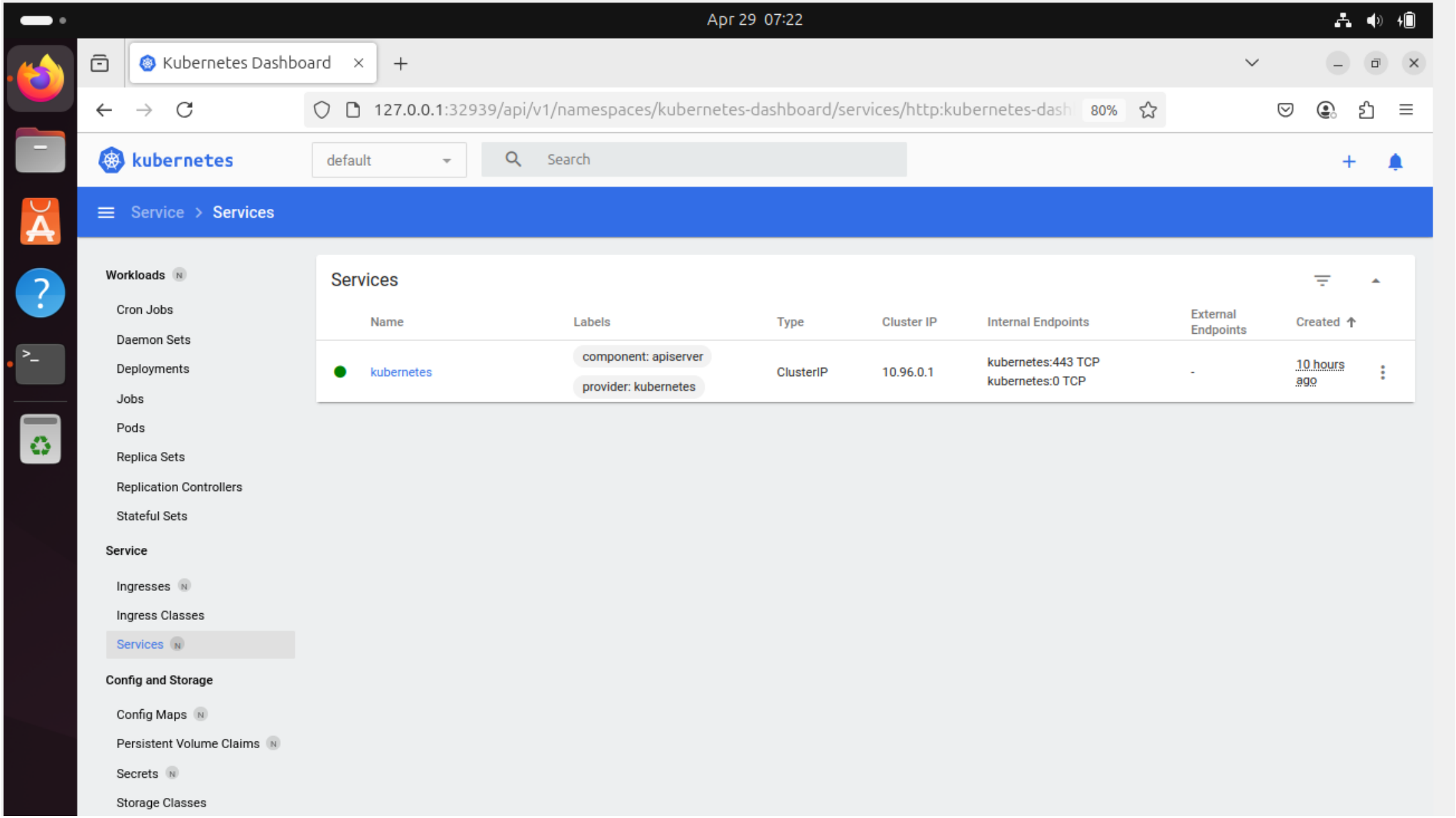
kube-controller-manager

- Control plane component that runs controller processes.
- Logically, each controller is a separate process, but to reduce complexity, they are all compiled into a single binary and run in a single process.
- There are many different types of controllers. Some examples of them are:
  - Node controller: Responsible for noticing and responding when nodes go down.
  - Job controller: Watches for Job objects that represent one-off tasks, then creates Pods to run those tasks to completion.
  - EndpointSlice controller: Populates EndpointSlice objects (to provide a link between Services and Pods).
  - ServiceAccount controller: Create default ServiceAccounts for new namespaces.
- kube-controller-manager is a control plane component in Kubernetes that runs various controllers to maintain the desired state of the cluster.
- It ensures that resources like Nodes, Pods, and Services function as expected.

cloud-controller-manager (Optional)

- A Kubernetes control plane component that embeds cloud-specific control logic.
- The cloud controller manager lets you link your cluster into your cloud provider's API and separates out the components that interact with that cloud platform from components that only interact with your cluster.
- The cloud-controller-manager only runs controllers that are specific to your cloud provider.

```
ubuntu@ubuntu-VirtualBox:~$ minikube kubectl -- get pods -A
NAMESPACE              NAME                                         READY   STATUS    RESTARTS      AGE
kube-system            coredns-668d6bf9bc-v2bnv                     0/1     Running   1 (32s ago)   9h
kube-system            etcd-minikube                                1/1     Running   1 (32s ago)   9h
kube-system            kube-apiserver-minikube                      1/1     Running   1 (32s ago)   9h
kube-system            kube-controller-manager-minikube             1/1     Running   1 (32s ago)   9h
kube-system            kube-proxy-z8h58                             1/1     Running   1 (32s ago)   9h
kube-system            kube-scheduler-minikube                      1/1     Running   1 (32s ago)   9h
kube-system            storage-provisioner                          1/1     Running   2 (32s ago)   9h
kubernetes-dashboard   dashboard-metrics-scraper-5d59dccf9b-gk7wz   1/1     Running   1 (32s ago)   9h
kubernetes-dashboard   kubernetes-dashboard-7779f9b69b-zmsxw        1/1     Running   1 (32s ago)   9h
```


Apr 29 07:22
Kubernetes Dashboard
127.0.0.1:32939/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dash
80%
kubernetes
default
Search
Service > Services
Workloads
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Service
Ingresses
Ingress Classes
Services
Config and Storage
Config Maps
Persistent Volume Claims
Secrets
Storage Classes
Services
Name
Labels
Type
Cluster IP
Internal Endpoints
External Endpoints
Created
kubernetes
component: apiserver
provider: kubernetes
ClusterIP
10.96.0.1
kubernetes:443 TCP
kubernetes:0 TCP
-
10 hours ago

Node Components

- Node components run on every node, maintaining running pods and providing the Kubernetes runtime environment.
  - kubelet
  - kube-proxy (Optional)
  - container runtime

kubelet

- An agent that runs on each node in the cluster.
- It makes sure that containers are running in a Pod.
- The kubelet takes a set of PodSpecs that are provided through various mechanisms and ensures that the containers described in those PodSpecs are running and healthy.
- The kubelet doesn't manage containers which were not created by Kubernetes.
- kubelet is a node agent in Kubernetes that runs on every worker node and ensures that Pods and their containers are running properly.
- It communicates with the Kubernetes API Server to receive tasks and report node status.

kube-proxy (Optional)

- kube-proxy is a network proxy that runs on each node in your cluster, implementing part of the Kubernetes Service concept.
- kube-proxy maintains network rules on nodes.
- These network rules allow network communication to your Pods from network sessions inside or outside of your cluster.

container runtime

- A fundamental component that empowers Kubernetes to run containers effectively.
- It is responsible for managing the execution and lifecycle of containers within the Kubernetes environment.

Minikube Start

- https://kubernetes.io/docs/tutorials/hello-minikube/
- https://minikube.sigs.k8s.io/docs/start/?arch=%2Fwindows%2Fx86-64%2Fstable%2F.exe+download
- https://minikube.sigs.k8s.io/docs/drivers/
- https://minikube.sigs.k8s.io/docs/drivers/docker/

References

- https://help.ubuntu.com/community/ServerGUI
- https://kubernetes.io/docs/home/
- https://minikube.sigs.k8s.io/docs/start/?arch=%2Fwindows%2Fx86-64%2Fstable%2F.exe+download
- https://kubernetes.io/docs/tutorials/kubernetes-basics/
- https://kubernetes.io/docs/tutorials/hello-minikube/
- https://minikube.sigs.k8s.io/docs/drivers/virtualbox/
- https://kubernetes.io/docs/tasks/tools/install-kubectl-linux/
- https://minikube.sigs.k8s.io/docs/start/?arch=%2Fwindows%2Fx86-64%2Fstable%2F.exe+download
- https://docs.docker.com/engine/install/ubuntu/
- https://minikube.sigs.k8s.io/docs/drivers/docker/