



kubernetes demo

Open Data Management & the Cloud
(Data Science & Scientific Computing / UniTS – DMG)

Kubernetes Basics (terminology)



★ Cluster:

Collection of hosts (servers) that helps you to aggregate their available resources. That includes ram, CPU, ram, disk, and their devices into a usable pool.

★ Master:

It is a collection of components which make up the control panel of Kubernetes. These components are used for all cluster decisions. It includes both scheduling and responding to cluster events.

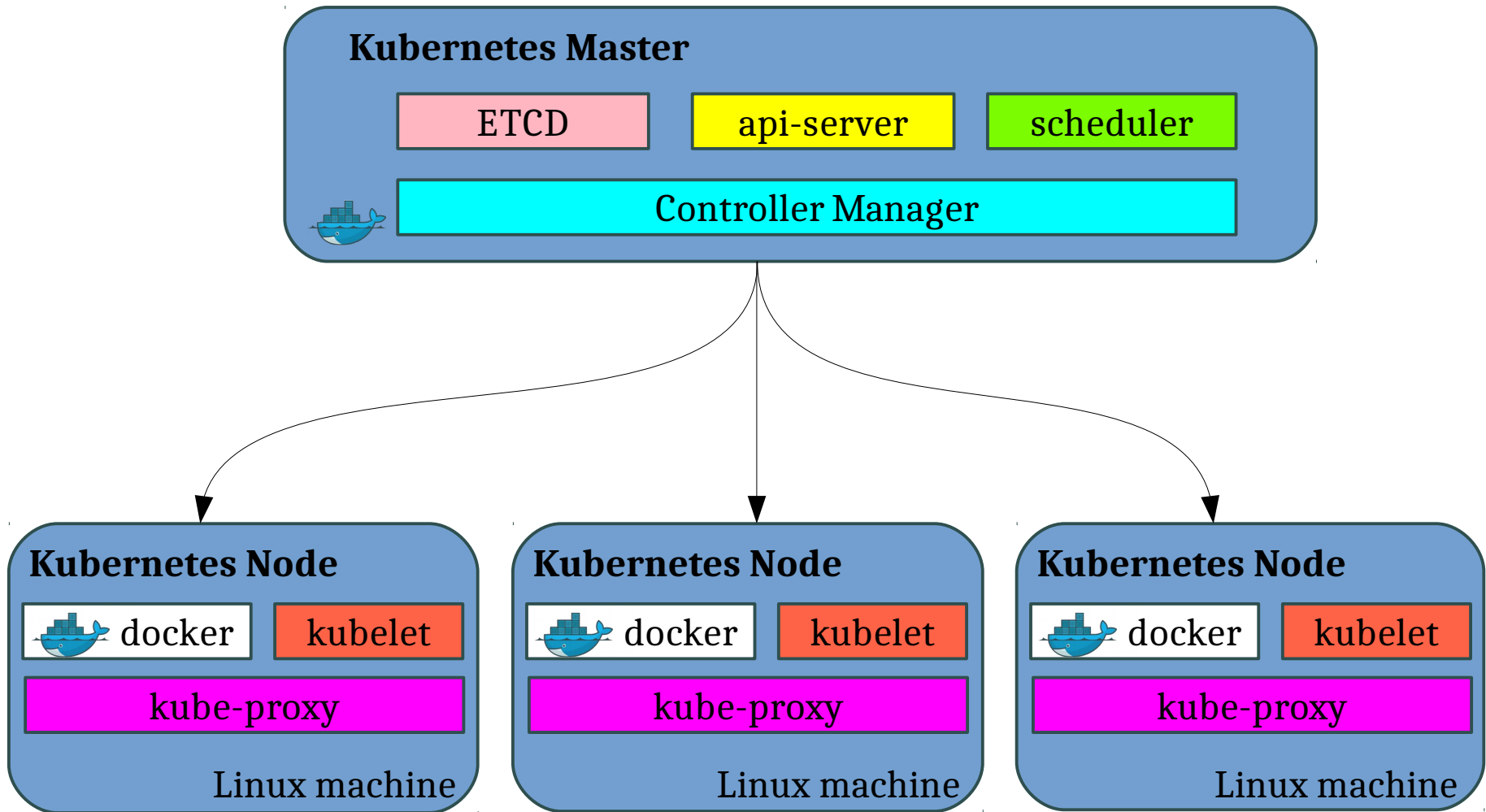
★ Node:

It is a single host which is capable of running on a physical or virtual machine. A node should run both kube-proxy, minikube, and kubelet which are considered as a part of the cluster.

★ Namespace:

It is a logical cluster or environment. It is a widely used method which is used for scoping access or dividing a cluster.

Kubernetes Architecture



Master node



It is the entry point for all kind of administrative tasks. There might be more than one master node in the cluster to check for fault tolerance.

★ API Server

It acts as an entry point for all the REST commands used for executing all operations in the cluster.

★ Scheduler

It schedules the tasks to the slave node. It stores the resource usage information for every slave node. It is responsible for distributing the workload.

It also helps to track how the working load is used on cluster nodes. It helps to place the workload on resources which are available and accept the workload.

★ Controller Manager

Is responsible for most of the controllers that regulates the state of cluster and performs a task. It can be considered as a daemon responsible for collecting and sending information to API server. It works toward getting the shared state of cluster and then make changes to bring the current status of the server to the desired state. The controller manager runs different kind of controllers to handle nodes, endpoints, etc.

The key controllers are replication controller, endpoint controller, namespace controller, and service account controller.

★ Etcd

It is a high availability key value store that can be distributed among multiple nodes. It stores the configuration information which can be used by each of the nodes in the cluster. It is accessible only by the API-server as it may have some sensitive information.

Worker/Slave nodes



Worker nodes contain all the required services to manage the networking between the containers, communicate with the master node, which allows you to assign resources to the scheduled containers.

★ Kubelet

gets the configuration of a Pod from the API server and ensures that the described containers are up and running. The kubelet process then assumes responsibility for maintaining the state of work and the node server. It manages network rules, port forwarding, etc.

★ Docker Container

runs on each of the worker nodes, which runs the configured pods (encapsulated application containers)

★ Kube-proxy

is a proxy-service running on each node. It helps in forwarding the request to correct containers and is capable of performing primitive load balancing. It works as network proxy making sure that the networking environment is predictable and accessible and at the same time it is isolated as well. It manages pods on node, volumes, secrets, creating new containers' health checkup, etc.

★ Pods

is a combination of single or multiple containers that logically run together on nodes

K8s Master and Nodes Components



Kubernetes Master

kube-api-server

exposes
kubernetes API

ETCD

distributed key-value
DB accessible to all

controller manager

multiple kind of
controllers to handle
nodes

scheduler

workload utilization
and pod allocation to
nodes



Kubernetes Node

kube-server

manages pods on
nodes, volumes,
creating/destroying
containers

kube-proxy-server

manages networking
for nodes

