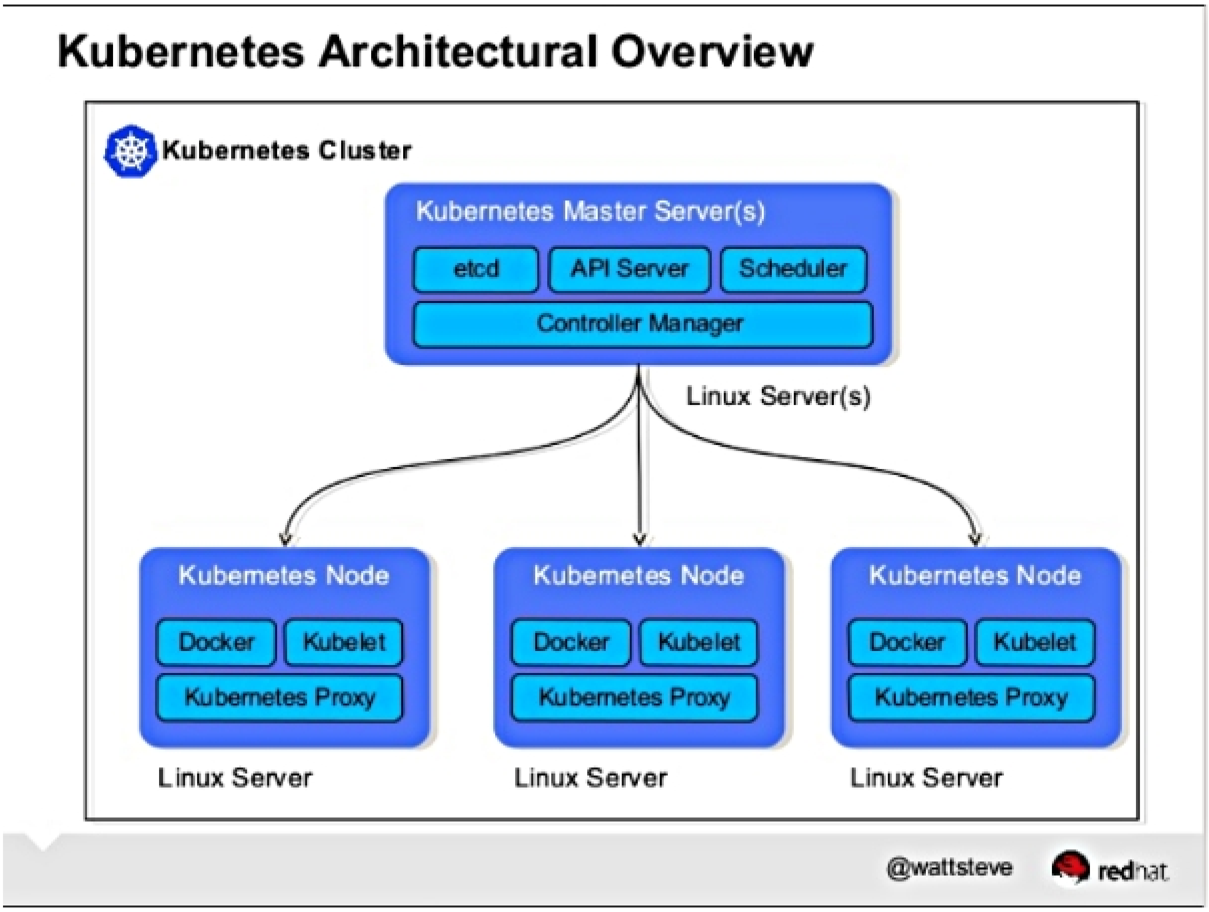
CKA Curriculum Part 8 - Core Concepts

Understand the Kubernetes API primitives

Quite a large subject to cover, but is well documented at https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md

Understand the Kubernetes cluster architecture

- Master Nodes
  - Etcd (unless external)
  - Kube-APIserver
  - Kube-Scheduler
  - Kube-Controller-Manager
- Worker Nodes
  - Some sort of CNI (Flannel, NSX-T, etc)
  - Kube-Proxy
  - Kubelet
  - Container runtime (Docker, RKT, containerd, etc)



Kubernetes Architectural Overview

Kubernetes Cluster

Kubernetes Master Server(s)

etcd

API Server

Scheduler

Controller Manager

Linux Server(s)

Kubernetes Node

Docker

Kubelet

Kubernetes Proxy

Linux Server

Kubernetes Node

Docker

Kubelet

Kubernetes Proxy

Linux Server

Kubernetes Node

Docker

Kubelet

Kubernetes Proxy

Linux Server

@wattsteve

redhat

ETCD

A consistent and highly available key-value store leveraged by Kubernetes to store cluster related data. ETCD can be located on the master nodes, or can be a separate cluster all together.

Kube-APIServer

Everything we do in Kubernetes goes through the API server. Think of this as as the front door to the Kubernetes cluster. This runs on all master nodes, to which a load balancer can be placed in front of.

Kube-Scheduler

Authority for determining which pods run on which nodes. Kube-Scheduler will also respects any constraints or requirements imposed by the pod specification, such as any nodeselector configuration.

Kube-Controller-Manager

Primarily responsible for checking, validating and rectifying current and intended state within the cluster. This includes:

- Reacting to node failure.
- Reacting to when the desired number of pods in a deployment (replication controller) is not satisfied.
- Populating endpoints for services based on defined inclusion criteria (ie labels).

CNI

Responsible for facilitating pod communication. A K8s cluster cannot function with a CNI.

Kube-Proxy

Maintains network rules and provides a level of abstraction by forwarding connections as appropriate

Kubelet

The kubelet takes a set of PodSpecs that are provided through various mechanisms (primarily through the apiserver) and ensures that the containers described in those PodSpecs are running and healthy.

Container Runtime

Usually Docker or Containerd, this provides the platform for containers to run on the host.

Understand Services and other network primitives

Services are categorised into the following types:

ClusterIP - Exposes the service on a IP address that is only accessible internally. Unless explicitly defined in the service yaml file, this is the default type.

NodePort - Exposes the service on each worker node's IP address (non docker bridge) over a static port. This is accessible externally from the cluster.

Load Balancer - Exposes the service externally using a **cloud provider's** load balancer. As implied, the cloud provider must support this functionality.