Certified Kubernetes Administrator (CKA) Exam Curriculum

This document provides the curriculum outline of the Knowledge, Skills and Abilities

that a Certified Kubernetes Administrator (CKA) can be expected to demonstrate.

# CKA Curriculum

## 25% - Cluster Architecture, Installation & Configuration

### **Manage role based access control (RBAC)**

### **Use Kubeadm to install a basic cluster**

### **Manage a highly-available Kubernetes cluster**

### **Provision underlying infrastructure to deploy a Kubernetes cluster**

### **Perform a version upgrade on a Kubernetes cluster using Kubeadm**

### **Implement etcd backup and restore**

## 15% - Workloads & Scheduling

### **Understand deployments and how to perform rolling update and rollbacks**

### **Use ConfigMaps and Secrets to configure applications**

### **Know how to scale applications**

### **Understand the primitives used to create robust, self-healing, application deployments**

### **Understand how resource limits can affect Pod scheduling**

### **Awareness of manifest management and common templating tools**

## 20% - Services & Networking

### **Understand host networking configuration on the cluster nodes**

### **Understand connectivity between Pods**

* Pod ip addreses are not persistent across restarts

#### CNI

* Container Network Interface
* plugins
* Assignes IP addreses to Pods and services
* Ensures L3 connectivity without relying on NAT withing K8s cluster

#### Kube-Proxy

* <https://mayankshah.dev/blog/demystifying-kube-proxy/>
* Uses iptables rules to forward requests to pods
* Generally independent of CNI
* Service Proxying
* Installed as daemonset which means it runs on each node in cluster
* Watches service and Endpoint/EndpointSlices and updates the routing rules on its host nodes to allow communicating over services
* Iptables
  + Linux command
  + Firewall utility to Control packets that enter the kernel
  + Program how linux kernel should treat packets
* IPVS
  + IP Virtual Server
  + Does L4 load balancing and scaling because iptables doesn’t do this
* Modes of execution
  + Iptables
    - default
  + Ipvs
  + Userspace
  + kernelspace

### **Understand ClusterIP, NodePort, LoadBalancer service types and endpoints**

#### ClusterIP

#### NodePort

#### LoadBalancer

#### Services

* Allow you to assign a single virtual IP address to a set of PODS
* Proxy / LB traffic to PODS
* Keeps track of pod ips

#### Endpoints

* 1:1 relationship with services
* Created for keeping trac of Pod IP services are proxying for
* Always kept insync with Pod IP
* Somewhat of a lookup table for pods

#### EndpointSlices

* Assist with endpoint scaling

### **Know how to use Ingress controllers and Ingress resources**

### **Know how to configure and use CoreDNS**

* DNS within K8s
* Allows pods to communicate via DNS Name instead of constantly changing IP

### **Choose an appropriate container network interface plugin**

## 10% - Storage

### **Understand storage classes, persistent volumes**

### **Understand volume mode, access modes and reclaim policies for volumes**

### **Understand persistent volume claims primitive**

### **Know how to configure applications with persistent storage**

## 30% - Troubleshooting

### **Evaluate cluster and node logging**

### **Understand how to monitor applications**

### **Manage container stdout & stderr logs**

### **Troubleshoot application failure**

### **Troubleshoot cluster component failure**

### **Troubleshoot networking**