

Designed for cluster administrators, this course covers essential Kubernetes setup and maintenance. It provides a solid foundation for the CKA exam while emphasizing real-world skills. Unlike many competitors, we focus on practical, project-based learning. Participants who complete the course will be well-prepared for both the exam and actual work environments.

This course is engineered for cluster administrators, offering comprehensive training in installing, setting up, and configuring a Kubernetes cluster. While it serves as an excellent primer for the CKA exam, its true strength lies in preparing participants for real-world scenarios. Unlike many competitors, we prioritize hands-on, project-centric skills. Participants who actively complete the course will be well-prepared for the CKA exam and, more importantly, will have the specialized expertise to make a meaningful impact in real-world operations.

For those that are seeking certification, and wonder how this course maps to the CKA requirements:

- **Cluster Architecture, Installation & Configuration:** this is the core of this course
- **Workloads & Scheduling:** basics covered in [Core Kubernetes](#), define namespace wide resource limit defaults and scheduling details (e.g., affinity, toleration's etc.) are covered during this course.
- **Service & Networking:** basics covered in [Core Kubernetes](#) while CNI and network policies are covered in this course
- **Storage:** mostly covered in [Core Kubernetes](#). During this course we explore on-prem storage solutions
- **troubleshooting:** application troubleshooting is part of [Core Kubernetes](#), during this course we cover general cluster and nodes related problems

During the workshop, students will set up and configure a 5-node kubernetes cluster with a HA control-plane.

Cluster Architecture

- Recap of k8s architecture (**Master** and **Worker Nodes**)
- Recap **control plane** components (APIServer, Controller-Manager, Scheduler and etcd)
- Choosing a **container runtime environment** (docker, containerd, cri-o)
- Discuss the role of **Container Network Interface** (CNI)
- A deeper dive into the role of **etcd**
- **Planning** a cluster (e.g, size, machine types, etc)

Installation

- Discuss different installation techniques (kubeadm, kops, ...)
- Using **kubeadm** to install a cluster
- Set up a **control plane**
- Using your **own image registry** for the control-plane (and cluster)
- Discuss **Admission Plugins**
- Installing a **CNI** implementation

- Set up an initial **worker nodes**
- **Adding masters** to your cluster
- **Configuring** kubeadm using YAML
- Setup a **HA etcd cluster**
- understand **etcd**
- Introduce **raft** (consensus algorithm)
- Topologies for **Highly Available clusters** (HA) (stacked vs external etcd nodes)

Configuring Role-Based Access Control (RBAC)

- recap **RBAC**
- Planning user-access
- User access **workflows**
- Managing cluster administrators and **users** using RBAC
- Limiting object and namespace **access**
- Using OpenID (**OIDC**) to authenticate cluster users
- Integrating with **LDAP**

Configuration

- Configuring your k8s installation
- Introduce etcd admin tools **etcdctl** and **etcdutl**
- **Backing-up** your etcd configuration
- **Scaling** worker nodes (adding/removing nodes)
- Discuss **Upgrading** strategies
- **Upgrade a cluster** to a new k8s version
- General Configuration tuning Best Practices
- Installation and configuration troubleshooting

Monitoring

Monitoring using Prometheus/Grafana

- Using **Prometheus**
- **Installing and configuring** Prometheus/Grafana (basics)
- Discuss **federation** and scalability-
- Understand prometheus **Service Discovery**
- Using prometheus **relabeling**
- Understand the structure of a **metric**
- Understand the different **metric types** (counter, gauge, histogram, ...)
- Adding **Service Monitors**
- Using **Grafana** dashboards
- **Importing** existing dashboards

Log aggregation with Loki

- introduce **Loki**
- Understand the Architecture of Loki
- Appreciate the **indexing behaviour** of Loki
- Discuss the benefits of using Loki over other logging solutions
- Discuss Loki **configuration** (promtail, storage, scraping, ...)
- Setting up Loki using helm

Storage

- Recap **volumes**
- Reiterate the different between **static** and **dynamic storage provisioning**
- Setting up **storage classes**
- Discuss **on-prem dynamic storage** solutions
- Setting up a distributed storage solution (**Rook/Ceph**)
- **Monitoring** storage

Workloads & Scheduling

Resources

- recap of container resources (requests/limits)
- setting up **default resource limits** per namespace
- Define and manage resource **sonatas**
- Monitoring workload resource

Pod Scheduling

- Introduce the **node selection** process
- Scheduling pods to nodes using **node selectors**
- Keeping pods away from nodes using **taints** and **tolerations**
- Inviting pods to nodes using **affinity**
- Co-locating (or not) pods using **podAffinity** and **podAntiAffinity**

Networking

- Recap networking objects (services, ingress, ...)
- Discuss **kubeproxy**
- Using **LoadBalancer** service types for on-prem clusters

CoreDNS

- overview of *core-dns*
- recap of *DNS records* in kubernetes (A/AAAA/SRV/CNAME/...)
- Understanding the *default configuration*
- Configure **upstream nameservers**

- Configure **logging**

Kube proxy

- recap the role of kube-proxy
- explore different **modes** (iptables, ipvs)
- understand how kube-proxy uses **iptables**
- discuss issues with using **iptables**
- **performance** tuning for iptables mode
- configure and explore **IPVS**
- use different IPVS **schedulers**

MetalLB

- Introduce **MetalLB** as a load-balancer for on-prem clusters
- Understand the architecture of MetalLB
- Pros and cons of **Layer-2** vs **BGP**
- Confusing IP pools for **MetalLB**
- Using **MetalLB CRDs**

Network Policies

- Explain **Network Policies**
- Understand the default behaviour in kubernetes
- Protecting your Network
- Protecting your Pods
- Allowing and Denying traffic **Ingress** and **Egress*
- Isolating **namespaces**
- Allowing based on **pods** and/or **namespaces**
- Allowing based on **IP Addresses**
- **Port** level access
- Policy **Patterns**
- **Encrypting secrets at rest** (etcd)

Technology stacks (Discussion)

- Recap Cluster Observability and **monitoring**
- Recap Cluster **logging** (ELK/Elastic Stack, Kibana, Loki, Grafana)
- Recap Resource and **performance monitoring** (Prometheus, Grafana)
- Reiterate the importance of **gitops**
- Discuss solutions for **Secrets**
- Tools for **Managing certificates**
- Using **replication** for namespace-bound objects
- Other tools