

Certified Kubernetes Network Engineer (CKNE) Training

OEM: Linux Foundation / CNCF • Duration: 4 Days (32 hrs) • Code: CKNE

COURSE MODULES & TOPICS

Module 1: Core Infrastructure and CNI (20%)

- Introduction to Kubernetes networking fundamentals
- The Container Network Interface (CNI) specification and its role
- CNI plugin installation, configuration, and lifecycle management
- Popular CNI plugins: Calico, Cilium, Flannel, Weave, Multus
- IP Address Management (IPAM) and pod CIDR allocation strategies
- Node-level IP allocation and cluster-wide IPAM design
- Linux packet-level networking primitives: iptables, ipvs, eBPF
- Linux troubleshooting tools: tcpdump, netstat, ss, ip, ethtool
- Pod connectivity diagnostics and end-to-end packet flow tracing
- Multi-interface pod configuration using Multus CNI
- Network namespace management and inspection
- Understanding veth pairs, bridges, and virtual networking
- Hands-on Lab: Deploy and validate a CNI plugin from scratch
- Hands-on Lab: Diagnose pod-to-pod connectivity issues

Module 2: Service Networking and DNS (25%)

- Kubernetes Service types: ClusterIP, NodePort, LoadBalancer, ExternalName
- Layer 4 service configuration and endpoint slices
- kube-proxy modes: iptables, ipvs, nftables
- Alternative kube-proxy replacements: Cilium eBPF, Calico BGP
- Service networking internals: virtual IPs, connection tracking
- CoreDNS architecture and configuration
- CoreDNS customization: plugins, zones, forwarding, caching
- Custom DNS policies for pods and namespaces
- Service discovery patterns in Kubernetes
- Troubleshooting service network traffic issues
- Gateway API concepts and object model (GatewayClass, Gateway, HTTPRoute, TCPRoute)
- Gateway API traffic management: header-based routing, weighted routing, mirroring
- Ingress controllers vs Gateway API — when to use which
- Hands-on Lab: Configure services, endpoints, and troubleshoot service networking
- Hands-on Lab: Deploy and configure Gateway API for advanced routing

Module 3: Advanced Traffic Management (20%)

- Traffic engineering principles for Kubernetes clusters
- LLM and AI workload traffic optimization (large payloads, streaming)

- Network routing implementation: BGP, static routes, ECMP
- Egress gateway configuration and centralized outbound traffic control
- Egress policies and network address translation (NAT) strategies
- Cross-cluster service discovery and multi-cluster networking
- Service mesh integration: Istio, Linkerd, Cilium Service Mesh
- Traffic splitting, canary deployments, and A/B testing at the network layer
- Load balancing algorithms: round-robin, least-connections, session affinity
- External load balancer integration (MetalLB, cloud LBs)
- Hands-on Lab: Configure egress gateway and traffic policies
- Hands-on Lab: Implement cross-cluster service discovery

Module 4: Network Security and Policy (25%)

- Kubernetes NetworkPolicy fundamentals — ingress and egress rules
- NetworkPolicy selectors: podSelector, namespaceSelector, ipBlock
- Default deny policies and zero-trust networking principles
- Advanced NetworkPolicy with Calico, Cilium (L7 policies, DNS-based rules)
- Encryption at node level: WireGuard, IPsec, MACsec
- Encryption at pod level: mTLS via service mesh
- TLS certificate lifecycle: cert-manager, ACME (Let's Encrypt), rotation
- Managing certificate authorities for internal PKI
- Pod-level authentication: SPIFFE, SPIRE identity framework
- Pod-level authorization and workload identity
- Network segmentation strategies for multi-tenant clusters
- Compliance considerations: PCI DSS, HIPAA, SOC 2 in Kubernetes networking
- Threat models: pod-to-pod attacks, DNS spoofing, ARP spoofing
- Hands-on Lab: Implement network policies for zero-trust segmentation
- Hands-on Lab: Configure mTLS between services using service mesh

Module 5: Observability (15%)

- Network observability principles for Kubernetes
- Network health metrics: latency, throughput, packet loss, jitter
- Prometheus metrics for network components (CNI, kube-proxy, ingress)
- Grafana dashboards for network monitoring
- End-to-end performance tracing across services
- Distributed tracing: OpenTelemetry, Jaeger, Zipkin
- eBPF-based observability tools: Hubble, Pixie, Falco (network aspects)
- Traffic audit logging and flow logs
- Cloud provider VPC flow logs integration
- Alerting on network anomalies and degraded performance
- Debugging network performance issues in production
- Hands-on Lab: Deploy Prometheus + Grafana for network monitoring
- Hands-on Lab: Trace a slow request across services using OpenTelemetry

Module 6: CKNE Certification Exam Preparation

- CKNE exam format and structure overview
- Review of all 5 exam domains with weighted emphasis
- Domain 1 (20%) — Core Infrastructure and CNI recap
- Domain 2 (25%) — Service Networking and DNS recap
- Domain 3 (20%) — Advanced Traffic Management recap
- Domain 4 (25%) — Network Security and Policy recap
- Domain 5 (15%) — Observability recap
- Performance-based exam task strategies (hands-on kubectl commands)
- Time-management tips for the 2-hour proctored exam
- Common pitfalls and how to avoid them
- Full-length mock CKNE exam (hands-on lab environment)
- Mock exam review and gap analysis
- CKNE exam registration guidance (training.linuxfoundation.org)
- Post-course study plan for weak knowledge areas
- Course wrap-up and Koenig Certificate of Completion