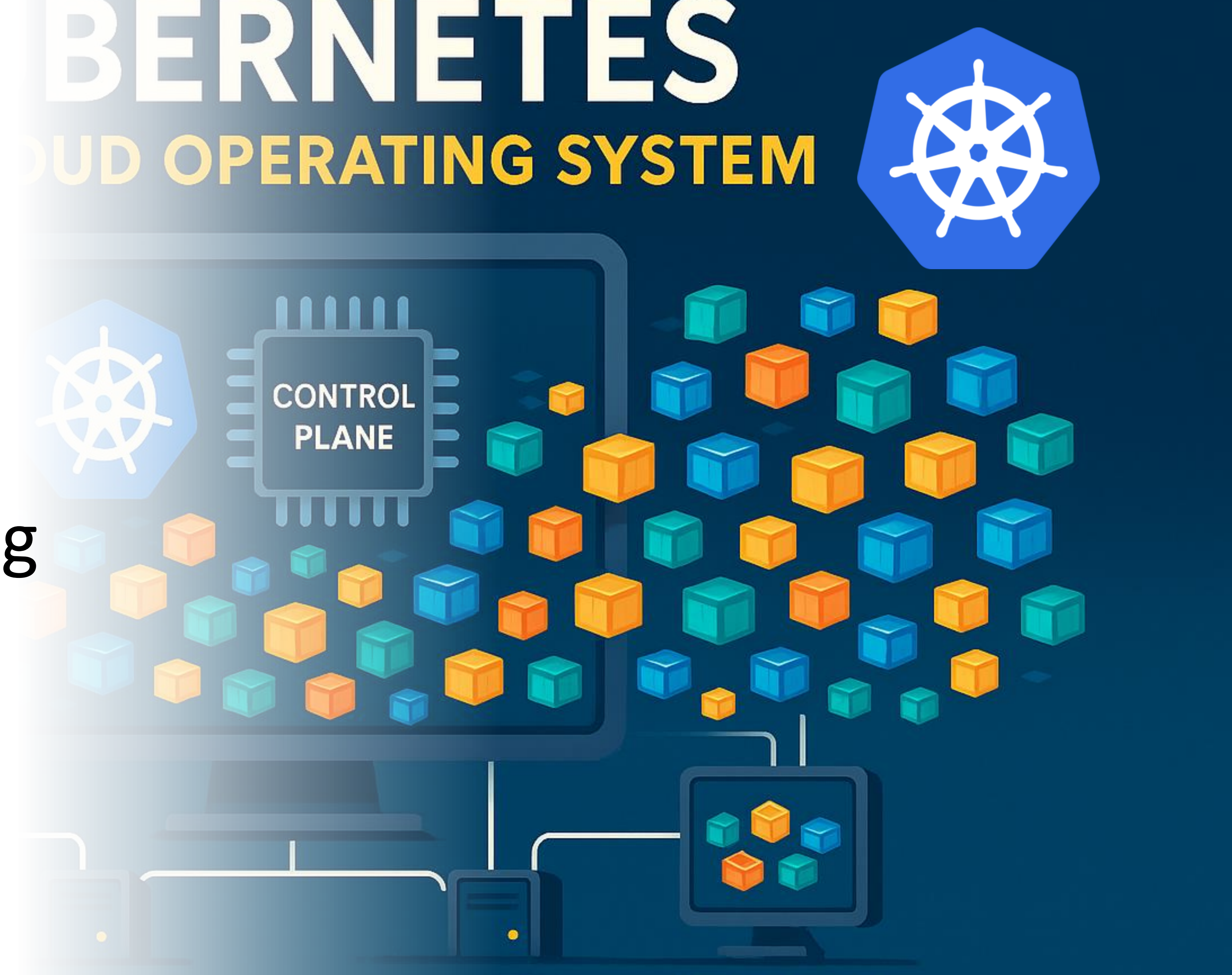
CS-UY 3913
Container
Operating Systems



K8 Networking

Instructor: Magdy Salem

Agenda

- Kubernetes Networking Model
- Pod-to-Pod Communication
- Service-to-Pod Communication
- CNI (Container Network Interface) Plugins
- Network Policies
- DNS and CoreDNS

Kubernetes Networking Model

- Every Pod gets a unique IP address
- Flat network model: all Pods can reach each other without NAT
- Containers within a Pod share network namespace
- Communication is direct unless restricted by Network Policies

Pod-to-Pod Communication

- Pods can communicate directly using IP addresses
- Communication works across nodes in the cluster
- Overlay networks handle routing between Pods
- Examples: Calico, Flannel, Cilium handle pod networking

Service-to-Pod Communication

- Services provide stable DNS and IP for accessing Pods
- Kube-proxy manages virtual IPs and routes traffic to Pod endpoints
- Round-robin load balancing between matched Pods
- Uses iptables, IPVS, or eBPF under the hood

CNI (Container Network Interface)

- Standard for configuring container network interfaces
- Each CNI plugin configures IP, routes, DNS, etc.
- Popular CNI plugins: Calico, Flannel, Weave, Cilium
- Plugins are pluggable and customizable

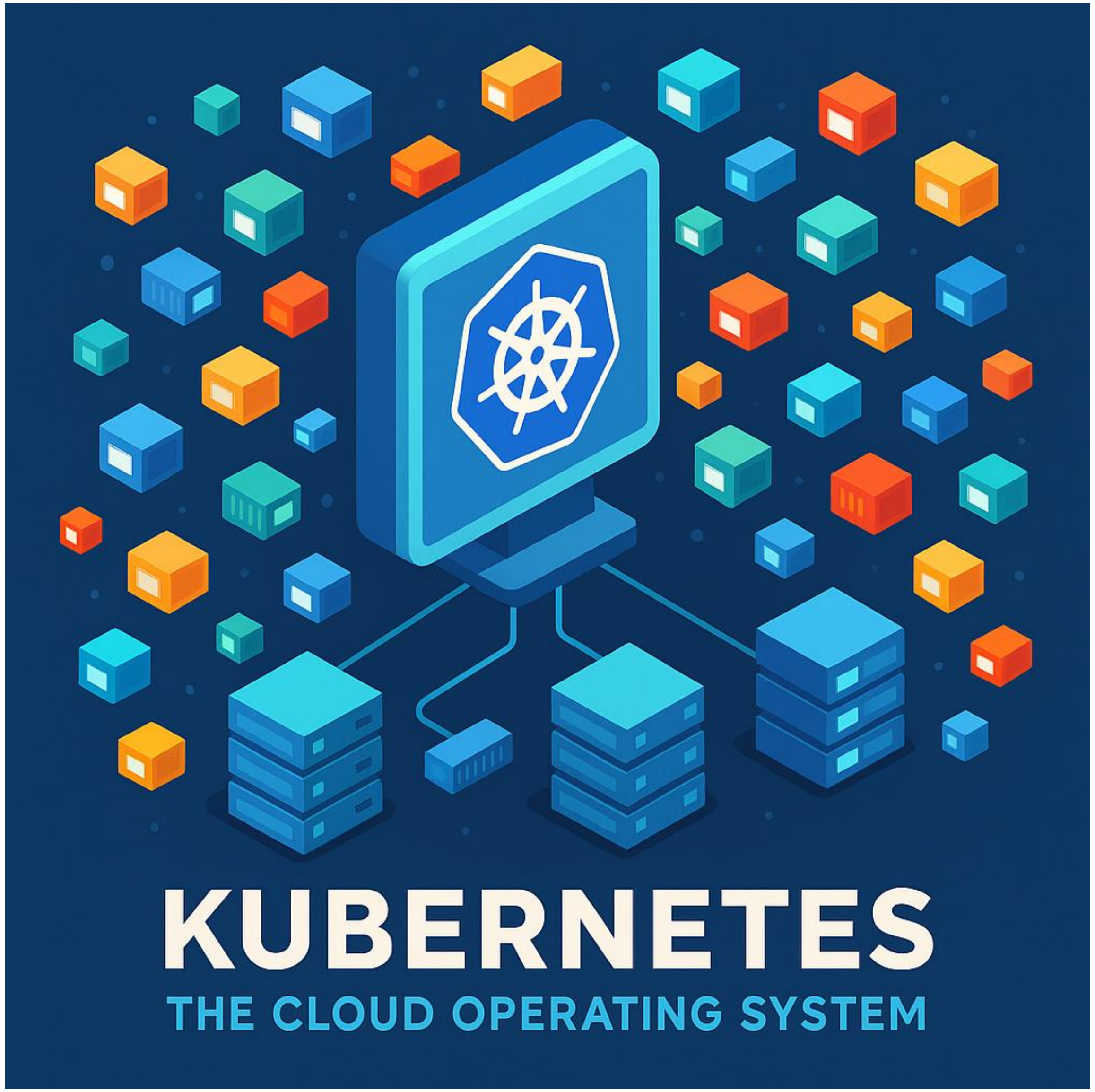
Network Policies

- Control traffic flow between Pods
- Default: all traffic is allowed (if no policy is defined)
- Define rules based on Pod selectors, IP blocks, namespaces
- Enforced by CNI plugin (e.g., Calico)

DNS and CoreDNS

- CoreDNS provides cluster-wide name resolution
- Every Service gets a DNS name *(e.g., my-svc.my-namespace.svc.cluster.local)*
- Pods resolve service names automatically using DNS
- DNS is tightly integrated with Kubernetes Services

Demo



Lab