Topic: Kubernetes Architecture and Components
Category: Cloud Infrastructure
Date: 2024-03-05
Author: Priya Sharma, Platform Engineer

CONTENT:

KUBERNETES OVERVIEW:
Kubernetes (K8s) is an open-source container orchestration platform that automates deployment, scaling, and management of containerized applications. Originally developed by Google, it is now maintained by the Cloud Native Computing Foundation (CNCF).

ARCHITECTURE COMPONENTS:

1. CONTROL PLANE (MASTER NODE):
   - kube-apiserver: Frontend to the control plane
   - etcd: Consistent and highly-available key-value store
   - kube-scheduler: Assigns pods to nodes
   - kube-controller-manager: Runs controller processes
   - cloud-controller-manager: Cloud-specific control logic

2. WORKER NODES:
   - kubelet: Agent that ensures containers are running
   - kube-proxy: Network proxy maintaining network rules
   - Container Runtime: Docker, containerd, or CRI-O

3. PODS:
   - Smallest deployable units in Kubernetes
   - Can contain one or multiple containers
   - Share network namespace and storage
   - Ephemeral by design

4. SERVICES:
   - Abstract way to expose applications
   - Types: ClusterIP, NodePort, LoadBalancer, ExternalName
   - Provides stable IP address and DNS name

5. DEPLOYMENTS:
   - Declarative updates for Pods and ReplicaSets
   - Supports rolling updates and rollbacks
   - Maintains desired state of application

6. CONFIGMAPS AND SECRETS:
   - ConfigMap: Configuration data in key-value pairs
   - Secrets: Sensitive data stored encoded
   - Decoupled from application code

NETWORKING MODEL:
- Each Pod gets its own IP address
- Pods can communicate across nodes without NAT
- Network plugins: Calico, Flannel, Cilium, Weave Net
- Ingress Controllers for HTTP routing

STORAGE IN KUBERNETES:
1. Volumes: Directory accessible to containers in a Pod
2. PersistentVolumes (PV): Cluster-wide storage resource
3. PersistentVolumeClaims (PVC): User's request for storage
4. StorageClasses: Dynamic provisioning of storage

AUTO-SCALING MECHANISMS:
1. Horizontal Pod Autoscaler (HPA): Scales pods based on CPU/memory
2. Vertical Pod Autoscaler (VPA): Adjusts resource requests/limits
3. Cluster Autoscaler: Adds/removes nodes based on demand

SECURITY FEATURES:
1. RBAC (Role-Based Access Control)
2. Network Policies for pod-to-pod communication
3. Pod Security Standards (PSS)
4. Secrets management and encryption

MONITORING AND OBSERVABILITY:
1. Metrics Server: Collects resource usage data
2. Prometheus: Time-series database for metrics
3. Grafana: Visualization dashboard
4. Jaeger: Distributed tracing
5. EFK Stack: Elasticsearch, Fluentd, Kibana for logging

HELM CHARTS:
- Package manager for Kubernetes
- Charts: Collection of files describing related K8s resources
- Templating engine for configuration
- Repository for sharing applications

OPERATOR PATTERN:
- Method of packaging, deploying, and managing applications
- Uses custom resources and controllers
- Extends Kubernetes API
- Examples: Prometheus Operator, Elasticsearch Operator

CLOUD-NATIVE ECOSYSTEM:
- Service Mesh: Istio, Linkerd
- GitOps: ArgoCD, Flux
- Serverless: Knative
- CI/CD Integration: Jenkins X, Tekton

BEST PRACTICES:
1. Use namespaces for logical separation
2. Implement resource requests and limits
3. Use readiness and liveness probes
4. Version control all manifests
5. Regular cluster upgrades and maintenance

TOOLS IN KUBERNETES ECOSYSTEM:
- kubectl: Command-line interface
- k9s: Terminal UI for K8s
- Lens: IDE for Kubernetes
- Kustomize: Native configuration management

- Skaffold: Local development tool

DISASTER RECOVERY STRATEGIES:
1. Regular etcd backups
2. Cluster state backup using Velero
3. Multi-cluster and multi-region deployments
4. Blue-green deployments for zero-downtime updates

PERFORMANCE OPTIMIZATION:
1. Right-sizing resource requests
2. Efficient scheduling with node affinity/anti-affinity
3. Horizontal scaling over vertical scaling
4. Optimized container images

CERTIFICATION PATH:
- Certified Kubernetes Administrator (CKA)
- Certified Kubernetes Application Developer (CKAD)
- Certified Kubernetes Security Specialist (CKS)

This comprehensive knowledge base is essential for anyone working with container orchestration at scale in modern cloud environments.