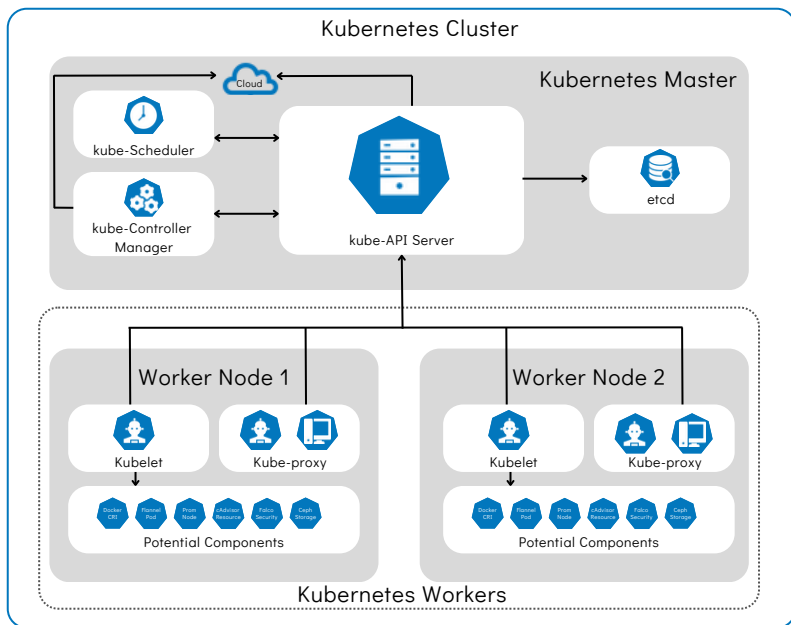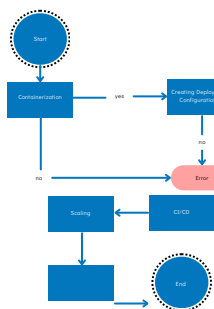# Kubernetes

Kubernetes is an open-source container orchestration platform used for automating deployment, scaling, and management of containerized applications.

It simplifies the process of deploying and managing applications in a dynamic and scalable environment.

Kubernetes Architecture

Deployment Process

Key Components

Pods: Containers grouped together for deployment.
Services: Abstraction to access Pod functionality.
Deployments: Manages lifecycle of replicated Pods.
Control Plane: Oversees cluster management operations.

kubectl command List

Pod Management

Commands	Example	Commands	Example
kubectl create pod	`kubectl create -f pod.yaml`	kubectl delete pod	`kubectl delete pod my-pod`
kubectl get pods	`kubectl get pods`	kubectl logs	`kubectl logs my-pod`
kubectl describe pod	`kubectl describe pod my-pod`		

Deployment Management

Commands	Example	Commands	Example
kubectl rollout history	`kubectl rollout history deployment/nginx`	kubectl scale deployment	`kubectl scale deployment nginx --replicas=3`
kubectl rollout status	`kubectl rollout status deployment/nginx`	kubectl create deployment	`kubectl create deployment nginx --image=nginx`
kubectl get deployments	`kubectl get deployments`		

Service Management

Commands	Example	Commands	Example
kubectl delete service	`kubectl delete service my-service`	kubectl describe service	`kubectl describe service my-service`
		kubectl get services	`kubectl get services`
kubectl expose	`kubectl expose deployment my-deployment --port=80 --target-port=8080`	kubectl create service	`kubectl create service nodeport my-service --tcp=80:8080`

Namespace Management

Commands	Example	Commands	Example
kubectl apply -f	`kubectl apply –f pod.yaml --namespace=my-namespace`	kubectl create namespace	`kubectl create namespace my-namespace`
kubectl describe namespace	`kubectl describe namespace my-namespace`	kubectl delete namespace	`kubectl delete namespace my-namespace`
kubectl get namespaces	`kubectl get namespaces`		

Node Management

Commands	Example	Commands	Example
kubectl get nodes	`kubectl get nodes`	kubectl describe node	`kubectl describe node my-node`
kubectl cordon	`kubectl cordon my-node`	kubectl drain	`kubectl drain my-node`
kubectl uncordon	`kubectl uncordon my-node`		

Kubernetes Logging
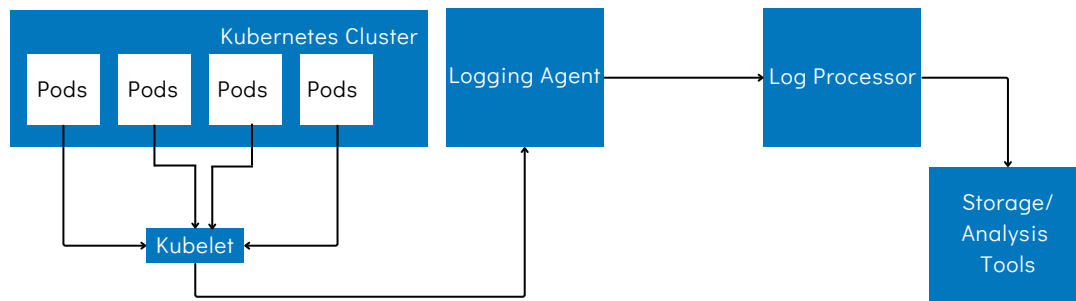
Kubernetes Logging - Security Logging Analysis

Pod Security Events: Log events related to pod security violations, such as unauthorized access attempts or privilege escalation.

Network Policy Violations: Capture events related to network policy violations, such as unauthorized network access between pods.

Cluster Authentication Failures: Log authentication failures within the Kubernetes cluster, indicating potential unauthorized access attempts.

Container Runtime Anomalies: Monitor container runtime activities for anomalies, such as suspicious process execution or file system modifications.

API Server Authorization Events: Log events related to API server authorization, such as denied requests or policy enforcement.
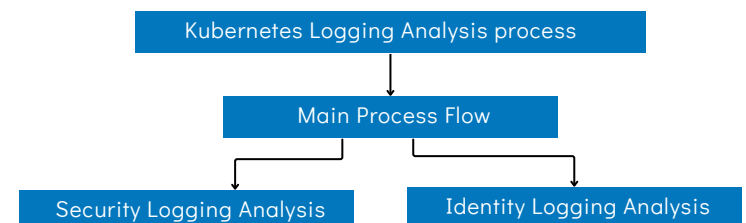
Kubernetes Logging - Identity Logging Analysis

User Activity Logs: Track user activity within the Kubernetes cluster, including authentication events and resource access.

Service Account Activity: Monitor service account usage and activity within the cluster, including creation, deletion, and resource access.

Role-Based Access Control (RBAC) Changes: Log changes to role-based access control (RBAC) configurations, including role assignments and policy updates.

Audit Trail of Kubernetes API Calls: Maintain an audit trail of Kubernetes API calls made by users and service accounts, including requests and responses.

Service Identity Management: Manage and log service identities used by applications and services within the Kubernetes environment.

Kubernetes security controls to the MITRE ATT&CK framework

Initial Access
- Role-Based Access Control (RBAC)
- Kubernetes API Server Configuration
- Secure Kubernetes API Access
- Secure Kubernetes Dashboard Configuration
- Multi-Factor Authentication (MFA) for Kubernetes API Access
- Restrictive Network Access Policies
- Limiting ingress/egress traffic flow.

Discovery
- Kubernetes Audit Logs
- Kubernetes API Server Configuration
- Network Policies
- Kubernetes RBAC Misconfigurations
- Pod Security Context Misuse
- Container Image Vulnerability Scanning
- Centralized Kubernetes Cluster Logging
- Unified logs for better visibility.

Credential Access
- Kubernetes Secrets Management
- Secure Kubernetes API Access
- Pod Security Context
- Rotation and Lifecycle Management for Kubernetes Secrets
- Encryption at Rest for Kubernetes Secrets
- Secure Configuration for Service Account Tokens
- Secure Storage Encryption Mechanisms
- Protecting data at rest.

Privilege Escalation
- Role-Based Access Control (RBAC)
- Service Account Usage
- Pod Security Policies (PSP)
- Least Privilege Principle for Role Bindings
- Regular Review and Audit of Service Account Permissions
- Automated Remediation for PSP Violations
- Role-Based Access Control Auditing
- Monitoring RBAC policy changes.

Defense Evasion
- Pod Security Context
- Image Signing and Verification
- Kubernetes Audit Logs
- Continuous Monitoring for Anomalies in Pod Behavior
- Automated Image Scanning and Verification in CI/CD Pipelines
- Automated Threat Detection Mechanisms
- Early detection of suspicious activities.

Persistence
- Pod Security Policies (PSP)
- Network Policies
- Kubernetes Secrets Management
- Automated Backup and Restore for Kubernetes Secrets
- Role-Based Access Control for etcd Data
- Kubernetes Security Best Practices Documentation
- Immutable Infrastructure Deployment Strategy
- Ensuring consistent and reliable environments.

Lateral Movement
- Network Segmentation
- Service Mesh
- Kubernetes Audit Logs
- Kubernetes Security Best Practices Documentation
- Secure Communication between Microservices in Service Mesh
- Implementing Network Policies for Inter-Pod Traffic
- Secure Configuration of Service Mesh
- Ensuring encrypted and authenticated communication.

Execution
- Container Runtime Security
- Kubernetes Resource Quotas
- Kubernetes Audit Logs
- Secure Configuration of Container Runtimes
- Implementing Resource Quotas for Namespace Isolation
- Real-time Alerting and Monitoring for Unauthorized Pod Creation
- Container Image Signing Enforcement
- Verifying container image authenticity.

Impact
- Disaster Recovery Planning
- Pod Security Policies (PSP)
- Testing and Validating Disaster Recovery Procedures Regularly
- Automated Failover and Redundancy for Critical Cluster Components
- High Availability Cluster Configuration
- Minimizing downtime risks.