Kubernetes RBAC and Security Best Practices

Day 28 – Devops 90 Days Challenge

What is RBAC?

- RBAC (Role-Based Access Control) controls who can perform what actions on which resources.
- It ensures fine-grained permissions and integrates with the Kubernetes API server.
- Used for multi-tenancy, security, and compliance.

Core RBAC Components

1. Role – Namespace-scoped permissions.
2. ClusterRole – Cluster-wide or cross-namespace permissions.
3. RoleBinding – Assigns a Role to users within a namespace.
4. ClusterRoleBinding – Assigns a ClusterRole to users cluster-wide.

RBAC Best Practices

- Follow Principle of Least Privilege (PoLP).
- Use namespaces for isolation.
- Avoid ClusterRole unless necessary.
- Use ServiceAccounts for workloads.
- Regularly audit permissions.
- Avoid using wildcards.

Kubernetes Security Best Practices

- Pod Security: Use securityContext to restrict privilege.
- Network Policies: Restrict traffic flow between pods.
- Secrets: Use encrypted secrets, Sealed Secrets, or Vault.
- Audit Logging: Track user actions.
- Admission Controllers: Enforce custom security policies.

Summary: RBAC Components

Component	Scope	Purpose
Role	Namespaced	Grant namespace permissions
ClusterRole	Cluster-wide	Grant cluster-level permissions
RoleBinding	Namespaced	Bind Role to user/SA
ClusterRoleBinding	Cluster-wide	Bind ClusterRole to user/SA