16-serviceaccount-rbac

ServiceAccount + RBAC (Role, RoleBinding, ClusterRole, ClusterRoleBinding) What it is ServiceAccount: identity for Pods to call Kubernetes API. Role / ClusterRole: permission definitions. RoleBinding / ClusterRoleBinding: attach permissions to identities. When to use Grant app Pods API permissions Limit developer/operator access scope Enforce least privilege Key fields subjects[]: users, groups, service accounts roleRef: permission object reference rules[]: apiGroups, resources, verbs Common commands bash kubectl get sa,role,rolebinding -n payments kubectl create serviceaccount app-sa -n payments kubectl auth can-i get pods --as=system:serviceaccount:payments:app-sa -n payments kubectl describe rolebinding app-reader-binding -n payments YAML example (namespaced read-only) ```yaml apiVersion: v1 kind: ServiceAccount metadata: name: app-sa namespace: payments apiVersion: rbac.authorization.k8s.io/v1 kind: Role metadata: name: pod-reader namespace: payments rules: - apiGroups: [""] resources: ["pods"] verbs: ["get", "list", "watch"] apiVersion: rbac.authorization.k8s.io/v1 kind: RoleBinding metadata: name: app-reader-binding namespace: payments subjects: - kind: ServiceAccount name: app-sa namespace: payments roleRef: kind: Role name: pod-reader apiGroup: rbac.authorization.k8s.io ``` Pod using ServiceAccount yaml apiVersion: v1 kind: Pod metadata: name: api namespace: payments spec: serviceAccountName: app-sa containers: - name: api image: nginx:1.27 Practical notes Prefer namespace-scoped Role over ClusterRole where possible. Continuously validate access with kubectl auth can-i.