**Kubernetes Security Configuration - Assessment Guide**

### **1. Kubernetes Security Features**

#### **Role-Based Access Control (RBAC)**

* **What it does:** Controls access to Kubernetes resources based on assigned roles.
* **How it works:** Uses Roles and RoleBindings (or ClusterRoles for cluster-wide permissions) to grant specific permissions.
* **Why it's important:** Ensures only authorized users can manage Kubernetes resources.

#### **Network Policies**

* **What it does:** Controls how pods communicate with each other and external services.
* **How it works:** Defines rules that allow or deny traffic based on labels, namespaces, and other criteria.
* **Why it's important:** Prevents unauthorized communication between services, enhancing security.

#### **Pod Security Policies (PSPs)**

* **What it does:** Defines security constraints for running pods.
* **How it works:** Enforces security best practices like preventing privileged containers or requiring read-only filesystems.
* **Why it's important:** Reduces the attack surface and enforces compliance with security policies.

### **2. Kubernetes YAML Configuration with Security Context Settings**

#### **YAML File: pod-security.yaml**

apiVersion: v1

kind: Pod

metadata:

name: secure-pod

spec:

containers:

- name: secure-container

image: my-node-app:latest # Replace with your Docker image

securityContext:

runAsUser: 1000 # Runs the container as user ID 1000

runAsGroup: 3000 # Runs the container with group ID 3000

allowPrivilegeEscalation: false # Prevents privilege escalation

readOnlyRootFilesystem: true # Ensures the root filesystem is read-only

capabilities:

drop:

- ALL # Drops all Linux capabilities to restrict privileges

ports:

- containerPort: 80 # Exposes port 80 on the container

### 

### 

### 

### 

### 