Kubernetes Threat Model

Persistence

Persistence in Kubernetes refers to the capability to store data beyond the lifecycle of individual pods. This is essential for stateful applications that need to retain data across restarts and failures. Kubernetes provides mechanisms for managing persistent storage using PersistentVolumes (PVs) and PersistentVolumeClaims (PVCs).

Kubernetes abstracts the underlying storage infrastructure through PersistentVolumes and PersistentVolumeClaims. This allows users to request and use storage resources without being tightly coupled to the specific storage implementation. Kubernetes supports various storage backends, including NFS, iSCSI, cloud storage (e.g., AWS EBS, GCE PD), and more.

RealLife Example:

Imagine a prosperous kingdom with a network of granaries and storehouses for long-term storage of essential goods. Just like these storage facilities are crucial for the kingdom's sustainability, persistence in Kubernetes ensures your applications retain data beyond the lifecycle of pods. However, persistence introduces security considerations that require careful attention.

Key Concepts

1. **PersistentVolume (PV)**

◦ A PersistentVolume is a piece of storage in the cluster that has been provisioned by an administrator or dynamically provisioned using StorageClasses.

◦ PVs are cluster resources, similar to nodes, and have a lifecycle independent of any individual pod.

2. **PersistentVolumeClaim (PVC)**

◦ A PersistentVolumeClaim is a request for storage by a user.

◦ PVCs consume PV resources and provide storage to pods.

3. **StorageClass**

◦ A StorageClass defines the types of storage available in the cluster.

◦ It allows for dynamic provisioning of PVs based on defined parameters.

4. **Access Modes**
   ◦ Access modes determine how a volume can be mounted.
   ◦ Common access modes include ReadWriteOnce (RWO), ReadOnlyMany (ROX), and ReadWriteMany (RWX).

Security Best Practices

1. **Use Secure Storage Backends**
   ◦ Ensure the underlying storage solutions are secure and comply with organizational security policies.
   ◦ Use encryption at rest to protect sensitive data.
2. **Implement Access Controls**
   ◦ Use Role-Based Access Control (RBAC) to restrict access to PVs and PVCs.
   ◦ Limit the ability to create, modify, or delete storage resources to authorized users.
3. **Backup and Recovery**
   ◦ Implement regular backups of critical data stored in PVs.
   ◦ Test recovery procedures to ensure data can be restored in case of failures.
4. **Monitor Storage Usage**
   ◦ Continuously monitor storage usage to detect anomalies and prevent resource exhaustion.
   ◦ Use tools like Prometheus and Grafana to track storage metrics.