ConfigMap

ConfigMap is **an API object that lets you store configuration for other objects to use**. Unlike most Kubernetes objects that have a spec , a ConfigMap has data and binaryData fields. These fields accept key-value pairs as their values

**keep your application code separate from your configuration**.

A ConfigMap is a dictionary of configuration settings. This dictionary consists of key-value pairs of strings. Kubernetes provides these values to your containers. Like with other dictionaries (maps, hashes, ...) the key lets you get and set the configuration value.

Replicat Se

Replicat

**Namespaces**

namespaces provides a mechanism for isolating groups of resources within a single cluster. Names of resources need to be unique within a namespace, but not across namespaces

Namespace-based scoping is applicable only for namespaced objects (e.g. Deployments, Services, etc) and not for cluster-wide objects (e.g. StorageClass, Nodes, PersistentVolumes, etc)

Kubernetes Namespaces can be used to divide a cluster into logical partitions allowing a single large Kubernetes cluster to be used by multiple users and teams, or a single user with multiple applications

Each user, team, or application running in a Namespace, is isolated from every other user, team, or application in other Namespaces and they operate as if they are the sole user of the cluster (note that Namespaces do not provide network segmentation

**Security Measures**

defining resource quotas,

support for auditing,

restriction of etcd access, regular security updates to the environment,

network segmentation,

definition of strict resource policies,

continuous scanning for security vulnerabilities,

using images from authorized repositories

* RBAC (Role-based access control) to narrow down the permissions.
* Use namespaces to establish security boundaries.
* Set the admission control policies to avoid running the privileged containers.
* Turn on audit logging.

RBAC

Role, ClusterRole,

rules that represent a set of permissions. Permissions are purely additive (there are no "deny" rules).

A Role always sets permissions within a particular [namespace](https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces);

ClusterRole, by contrast, is a non-namespaced resource

 RoleBinding and ClusterRoleBinding

Kubernetes Security Best Practices

1 .Enable Kubernetes Role-Based Access Control (RBAC)

### 2. Use Third-Party Authentication for API Server

### 3. Protect etcd with TLS, Firewall and Encryption

### 4. Isolate Kubernetes Nodes

### 5. Monitor Network Traffic to Limit Communications

### 6. Use Process Whitelisting

### 7. Turn on Audit Logging

### 8. Keep Kubernetes Version Up to Date

### 9. Lock Down Kubelet