

# Application Configuration Management in Kubernetes

Application configuration management in Kubernetes is required due to various reasons such as:

- to provide the same configuration, but modify it for different environments
- to update configuration without reworking every individually affected item
- to ensure that best practices are followed every time

There are four different approaches to handling application configuration management in Kubernetes.

## 1. Replicate and Customize:

This is the simplest approach and is also the least flexible. In this, existing definition is replicated, and then it is customized as per needs. The replicated configuration needs to be reworked for changes.

## 2. Parameterized Templating

In this, resource definitions are templated as per requirements. Parameters inserted into a template can either make use of sane default values or apply a user provided value. This method allows flexibility

## 3. Overlay Configuration

In this, there is a base configuration which acts as a general configuration. But the base configuration can be customized by adding layer of different configurations for requirements. The base is then merged with the overlay.

## 4. Programmatic Configuration

In this, Domain Specific Language (DSL) or a more generic programming language is used. This enables the creation of a default definition for an application's configuration, but also provides for replacing default configuration items with specific values as per needs. This also enables reusability of configuration definitions.

### ConfigMaps:

It stores non confidential data in key-value pairs. It allows for setting configuration data separately from application code. It can be passed to pod as environment variables, command-line arguments, or as configuration files. ConfigMaps can be mounted as data volumes. ConfigMaps can also be used by other parts of the system, without being directly exposed to the Pod.

### Secrets:

It contains sensitive data such as password, or key. Secrets can be created independently of the Pods that use them. It is similar to ConfigMaps but it hold only confidential data.