# Secrets Compared to ConfigMaps

This lesson briefly compares Kubernets Secrets and ConfigMaps.

WE'LL COVER THE FOLLOWING



## The Similarities

So far, Kubernetes Secrets do not seem to differ from ConfigMaps. From a **functional perspective**, they are, indeed, the same.

- Both allow us to inject some content. Both can use files, literal values, and files with environment variables as data sources.
- Both can output data into containers as files or as environment variables.
- Even the syntax for using Secrets is almost the same as the one used for ConfigMaps.

## The Differences

> The only significant difference between ConfigMaps and Secrets is that the latter creates files in a *tmpfs* (temporary file storage).

Secrets are constructed as in-memory files, thus leaving no trace on the host's files system. That, in itself, is not enough to call Secrets secure, but it is a step towards it. We'd need to combine them with *Authorization Policies* to make the passwords, keys, tokens, and other never-to-be-seen-by-publicly types of data secure. Even then, we might want to turn our attention towards third-party Secret managers like HashiCorp Vault.

## Conclusion #

> Secrets are almost the same as ConfigMaps. The main difference is that the secret files are created in *tmpfs*. Kubernetes secrets do not make your system secure. They are only a step towards such a system.

---

In the next lesson, we will explore the shortcomings of using Secrets and the way we can overcome these.