# Not so Secretive Secrets

In this lesson, we will explore the insecurities associated with using Secrets and discuss the actions we can perform to secure the environment.

WE'LL COVER THE FOLLOWING



## The Insecurities

Almost everything Kubernetes needs is stored in etcd. That includes Secrets. The problem is that they are stored as plain text. Anyone with access to etcd has access to Kubernetes Secrets. We can limit the access to etcd, but that's not the end of our troubles.

*etcd* stores data to disk as plain text. Restricting the access to etcd still leaves the Secrets vulnerable to who has access to the file system. That, in a way, diminishes the advantage of storing Secrets in containers in tmpfs. There's not much benefit of having them in tmpfs used by containers, if those same Secrets are stored on disk by etcd.

Even after securing the access to etcd and making sure that unauthorized users do not have access to the file system partition used by etcd, we are still at risk. When multiple replicas of etcd are running, data is synchronized between them. By default, etcd communication between replicas is not secured. Anyone sniffing that communication could get a hold of our secrets.

Kubernetes Secrets are a step in the right direction. It is, without a doubt, better to use Secrets than to expose confidential information as environment variables or other less secure methods. Still, Secrets can give us a false sense of security.

## How to Secure?

## How to Secure? #

We need to take additional precautions to protect ourselves. That might include, but is not limited to, the following actions:

- Secure the communication between etcd instances with SSL/TLS.
- Limit the access to etcd and wipe the disk or partitions that were used by it.
- Do not define Secrets in YAML files stored in a repository. Create Secrets through ad-hoc `kubectl create secret` commands. If possible, delete commands history afterward.
- Make sure that the applications using Secrets do not accidentally output them to logs or transmit them to other applications.
- Create policies that allow only trusted users to retrieve secrets. However, you should be aware that even with proper policies in place, any user with permissions to run a Pod could mount a Secret and read it.

We did not yet explore etcd configuration, nor did we learn how to set up authorization policies. For now, just remember that Secrets are not as secured as one might think. At least, not those provided by Kubernetes community. We do encourage you to use them, as long as you're aware of their shortcomings.