**Controller**

A **controller** in Kubernetes is a control loop that watches the shared state of the cluster through the apiserver and makes changes in an attempt to move the current state towards the desired state.

Built-in controllers such as the ReplicaSet and StatefulSet controllers ensure that the number of Pods matches the desired state defined in the controller's spec.

While **controllers** are ideal for stateless operations, Kubernetes **operators** shine when managing stateful applications.

Operators extend Kubernetes' functionality by combining custom resources and custom controllers.

They encode operational knowledge into software, automating complex tasks such as software upgrades, backup/restore, and failure recovery.

**Kubernetes native resources** are the building blocks that make your containerized applications come to life. They provide essential functionalities for orchestrating and managing containers.

**Native Resources Use Cases:**

* **Use native resources for straightforward containerized applications.**
* **They are well-suited for stateless, horizontally scalable microservices.**

**🔧 Custom Resources Use Cases:**

* **Custom resources are beneficial for complex, stateful applications.**
* **Applications with unique configuration needs often benefit from CRD**