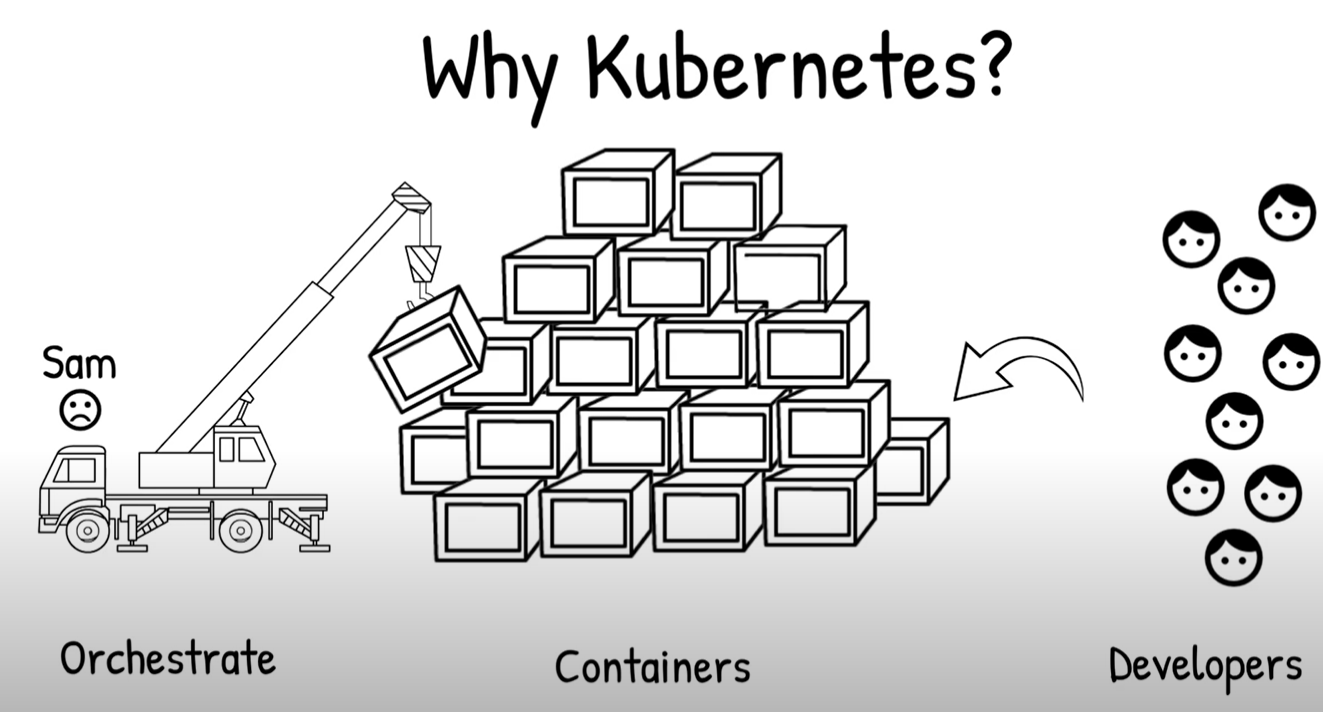


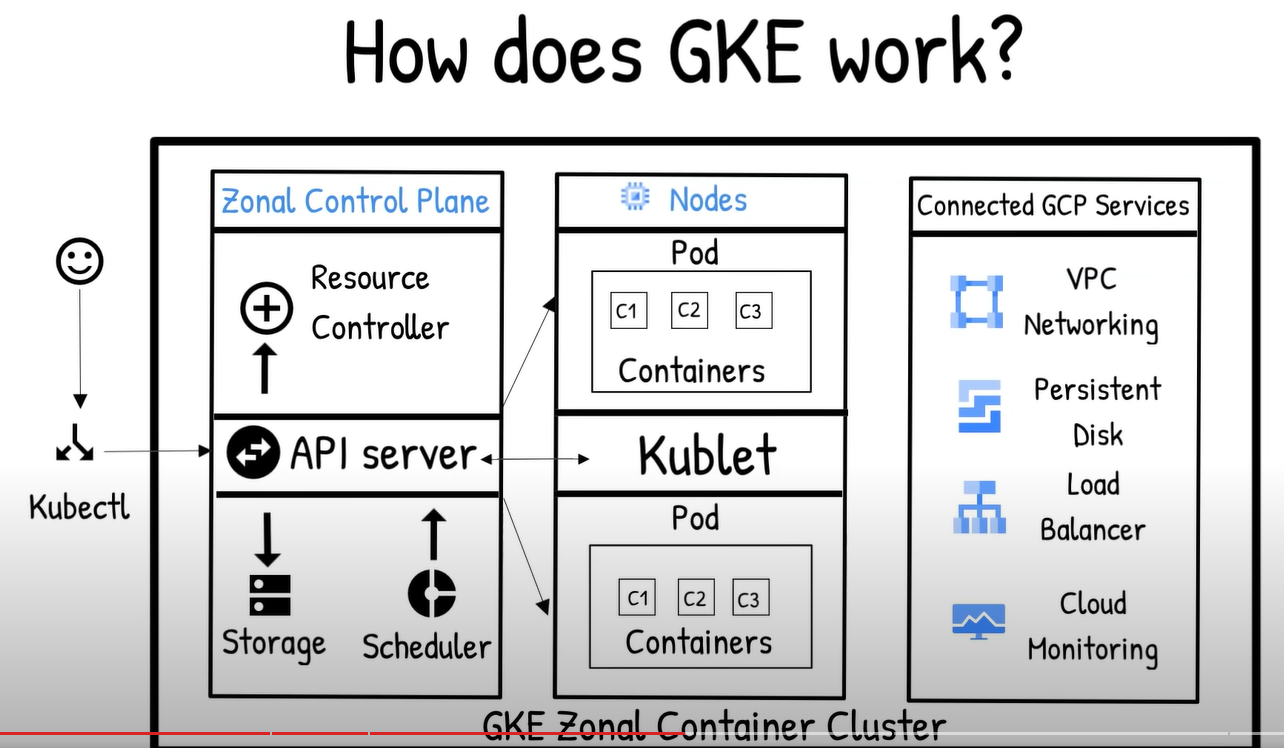
Containers **decouple** the OS from the application dependencies and the code .

Due to this Abstraction ,we can log into each machine and instruct it to run containers ,iti just pull down just the files that have changed since the last containers and run the new code .

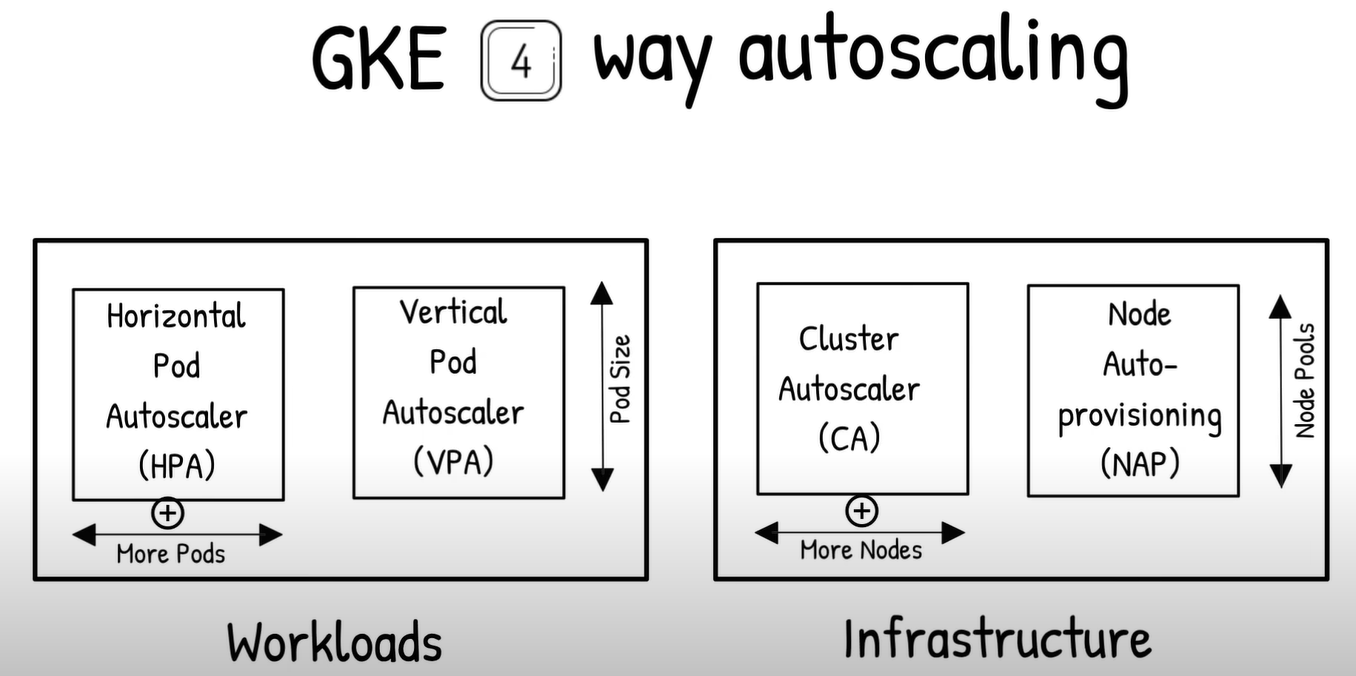
If you need to rollback ,all the old files are still there , as containers images are immutable.











* **Horizontal Pod Autoscaler (workloads):** increasing or decreasing the number of **Pods**.
* **Vertical Pod Autoscaler** **(workloads)**:automatically adjusts the **CPU** and memory reservations for your pods .
* **cluster autoscaler** (Infra)automatically resizes the number of **nodes** in a given node pool, based on the demands of your workloads
* **Node auto-provisioning (infra):** new **node pools** are created and deleted automatically.

The Kubernetes Vertical Pod Autoscaler **automatically adjusts the CPU and memory reservations for your pods to help "right size" your applications**. This adjustment can improve cluster resource utilization and free up CPU and memory for other pods.

**Horizontal Pod Autoscaler**: changes the shape of your Kubernetes workload by automatically increasing or decreasing the number of Pods in response to the workload's CPU or memory consumption, or in response to custom metrics reported from within Kubernetes or external metrics from sources outside of your cluster.

With Autopilot clusters, you don't need to worry about provisioning nodes or managing node pools because node pools are *automatically* provisioned through node auto-provisioning, and are [automatically scaled](https://cloud.google.com/kubernetes-engine/docs/concepts/cluster-autoscaler) to meet the requirements of your workloads.

**Overview**

*Node auto-provisioning* automatically manages a set of node pools on the user's behalf. Without node auto-provisioning, GKE starts new nodes only from user-created node pools. With node auto-provisioning, new node pools are created and deleted automatically.

**Horizontal** Scaling means modifying the **compute resources** of an existing cluster, for example, by adding new nodes to it or by adding new pods by increasing the replica count of pods (Horizontal Pod Autoscaler).

**Vertical** Scaling means to modify the **attributed resources** (like CPU or RAM) of each node in the cluster.



