organizations have to use multiple containers to
Ensure availability
load balancing
scale up and down based on user load

deploying
scheduling
scaling
load balancing
batch execution
roll backs
monitoring

} We have to do that
Here container management tools comes into picture

Docker → creates containers

kubernetes → manages containers

* <u>Features of kubernetes</u>

① Automatic bin packing -
We have 5 servers each having 10 GB of memory (RAM) we have a list of jobs to run on these 5 servers. Every job has different resource (memory) requirement

kubernetes will take care of packaging these jobs (containers) in bins (servers) in the most efficient way

kubernetes automatically packages your application and schedules the containers based on the requirements and resources available.

Automatically places containers based on their resource requirements like CPU & Memory (RAM) while not sacrificing availability saves resources.

② Pods & Nodes -

We do not interact with containers directly. Containers are raped into inside of the functional unit which is called as a "pod"
pod can have single container or multiple containers
pods are housed inside nodes.
A node can have a single pod or multiple pods.

When you specify a pod, you can optionally specify how much CPU and memory (RAM) each container needs.
When containers have resource requests specified the scheduler can make better decisions about which nodes to place pods on

③ Service discovery & load balancing -
How k8s orginizes containers
k8s doesn't run containers directly instead it wrap one or more containers into a higher-level structure called a pod
A pod contains
- An application container (or, in some cases, multiple containers)
- storage resources
- a unique network IP