



# kubernetes

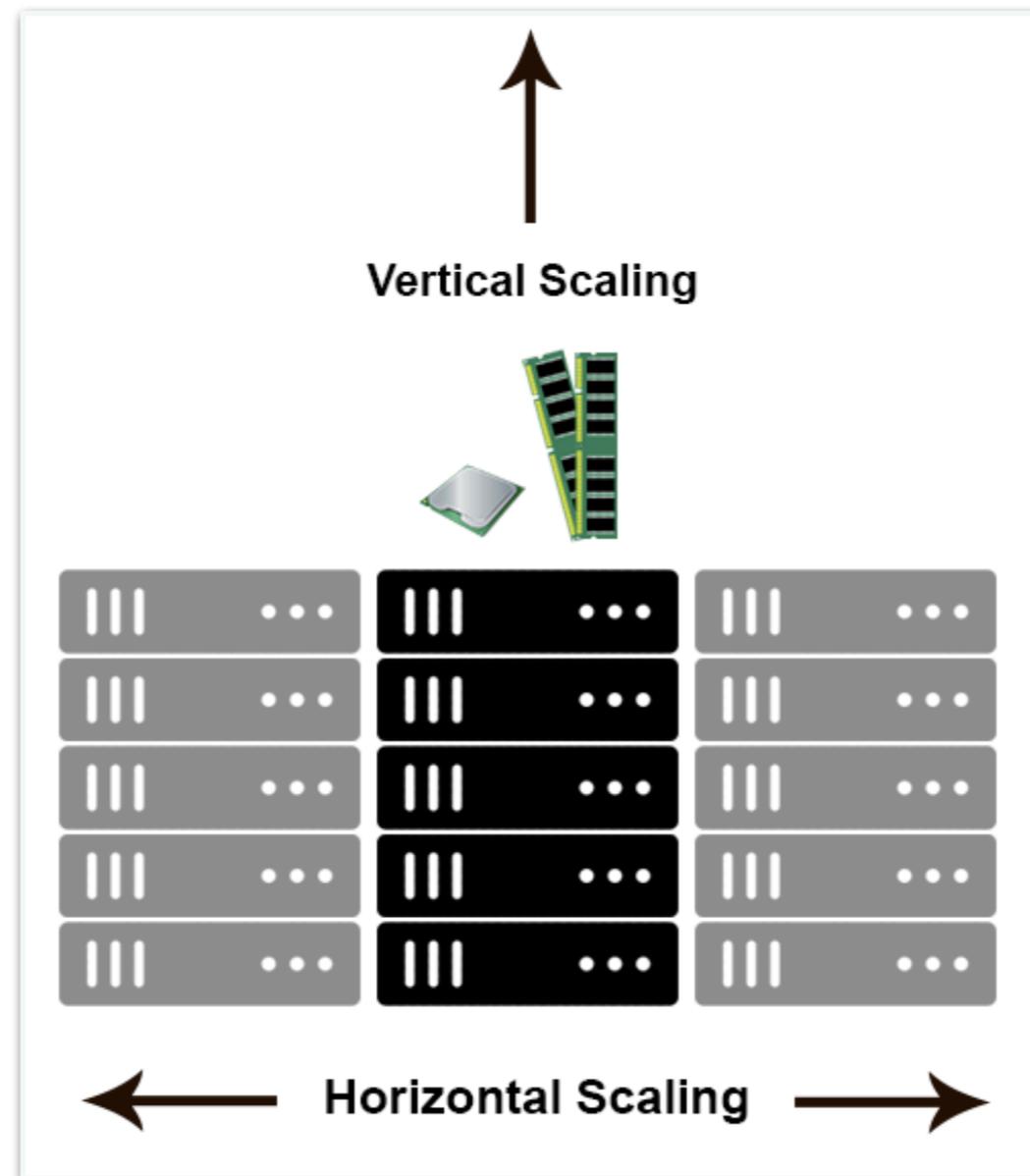
---

*Kubernetes: Scale Pods*

# *KUBERNETES : Basics of Kuebernetes*

---

- You can scale the Pods **Vertically** or **Horizontally**.



## *KUBERNETES : Basics of Kuebernetes*

---

- Stateless Applications can be Scaled up **Horizontally**.
- **Stateless**, no State of application. Doesn't have local files and local sessions.
- A **stateless** system can be seen as a Black box, where at any point in time the value of the output(s) depends only on the value of the input(s).
- A **stateful** system instead can be seen as a box where at any point in time the value of the output(s) depends on the value of the input(s) and of an internal state, so basically a stateful system is like a state machine with "*memory*" as the same set of input(s) value can generate different output(s) depending on the previous input(s) received by the system.

## *KUBERNETES : Basics of Kuebernetes*

---

- All traditional database (MySQL, Postgres) are stateful services, they have file systems that can't be split over the multiple Instances.
- A **stateful** system will requires that multiple threads of execution access and update the internal state of the system in an exclusive way, hence there will be a need for a synchronization point.

## *KUBERNETES : Basics of Kuebernetes*

---

- Modern Web-Applications are stateless.
- **Session Management** needs to be put outside the container.
- Files can't be saved locally in **Containers**.
- Stateful applications can't be Horizontally Scale but user can Scale these Applications **Vertically**.

## *KUBERNETES : Basics of Kuebernetes*

---

- Scaling in Kubernetes can be done using **Replication Controller**.
- *Replication Controller* ensures that a specified number of pod replicas are running at any one time.
- Pods created by Replication Controller will be **automatically replaced**, if pod failed, deleted or terminated.
- If there are too many pods, the ReplicationController terminates the extra pods. If there are too few, the ReplicationController starts more pods.

## *KUBERNETES : Basics of Kuebernetes*

---

- Scaling in Kubernetes can be done using **Replication Controller**.
- *Replication Controller* ensures that a specified number of pod replicas are running at any one time.
- Pods created by Replication Controller will be **automatically replaced**, if pod failed, deleted or terminated.
- If there are too many pods, the ReplicationController terminates the extra pods. If there are too few, the ReplicationController starts more pods.

## *KUBERNETES : Basics of Kuebernetes*

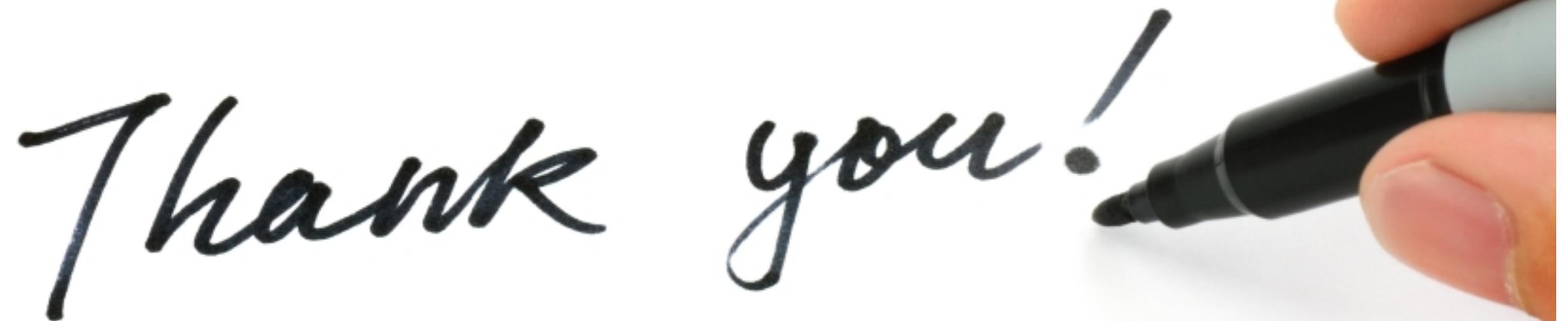
---

- **Replication Controller** manage multiple Pods on Single Node or Multiple Pods on multiple Nodes.
- ReplicationController is often abbreviated to “rc” or “rcs” in discussion.

*Will see you in Next Lecture...*

---

*Thank you!*

A close-up photograph of a hand holding a black marker, writing the words "Thank you!" in a cursive script on a white surface. The hand is positioned on the right side of the frame, with the fingers gripping the marker. The text is written in a fluid, handwritten style.

*See you in next lecture ...*