**Kubernetes: Why?**

* Prerequisite to learn kubernetes - Knowledge on Docker
* Why is it important? People started moving to microservices.

**Diff between Docker and Kubernetes:**

* Docker is a container platform.
* Kubernetes is a container orchestration (text def 🙂)

**Docker Problems:**

1. Relies on Single HOST
2. No Auto Healing feature
3. No Auto Scaling feature
4. No Load Balancing feature
5. It's a simple platform meaning it does not support enterprise level Solutions/Applications. (because of NO Firewall, load balancers,auto scale, auto healing, api gateways etc)

* **Kubernetes (or) Dockerswam will Solve all of the above problems.**
* **Note, Docker independently never used in production. Use either dockerswam/ K8s**

Simple Explanation on each of above problems:

**Relies on Single HOST:**

* Containers are **Ephemeral** in Nature. Meaning they have SHORT LIFE in nature and can die anytime.
* Let us say I have a host (VM or EC2 instance) on top that I have installed docker. I have 100 containers running, let us name them C1 to C100.
* if any of the containers (let's say C1) is taking a lot of resources like memory or CPU all of sudden. Because of this C1, other containers may get impacted and die.
* This is all because we have a single host.

**No Auto Healing Feature:**

* There are many many reasons the container may go down. Which results, Application running inside container may not be accessible.
* Some One should act on the container manually to bring the container or application up.
* There is no Mechanism/process in docker to bring the containers up automatically.
* Devops Engineer/Anyone cannot monitor 100’s of containers continuously (by issuing docker ps command)

**No Auto Scaling Feature:**

* Let us say I have a host (VM or EC2 instance) on top that I have installed docker.
* I have a few containers running for an Application let us say Netflix, I expected Audience or load if 10 million. All of sudden because of the new movie release (or) good talk of the movie, audience or load increased to 12 million.
* To act on increased load (or) to provide the high availability containers should have a feature of Auto Scaling of resources automatically. Which docker doesn't have.

**No Load Balancing Feature:**

* Docker doesn't have a load balancing feature, in above example, even though we increase the containers manually.
* We cannot tell users like first 10k users access the application url using C1 container ip address, and next 10-20k users access C2 container ip address etc as such.
* As an end user, we can try to access netflix.com and select movies.
* Once the requests come in, the docker/container should be able to distribute the load to the respective container. Which docker doesn’t have.

**No Support for Enterprise Level Support:**

* It's a simple platform meaning it does not support enterprise level Solutions/Applications.
* To provide enterprise level support it should have all - Firewall Support, load balancers, auto scale, auto healing, api gateways features etc)

**Solution Notes in Kubernetes:**

By default, kubernetes is a cluster.

Cluster is a group of nodes (One master node and multiple worker nodes).

**Resolution to docker problem-1:**

* We saw, one of the containers (c1) is impacting other containers because of a lack of resources in docker.
* In kubernetes, As it is clustered in nature and has multi-node architecture, if any of the applications in one node is affected, kubernetes immediately move them to different nodes and make them highly available.

**Resolution to Auto Healing Problem:**

* There are many many reasons the container (or) pod may go down.
* Whenever there is damage, kubernetes should control and fix the damage.
* In kubernetes, there is support for auto healing - meaning, whenever the container is going down (or) ever before it goes down. Kubernetes API server will rollout (or) start a new container which will not impact the Applications availability.

**Resolution to Auto Scaling Problem:**

* Whenever there is a high load on the Application kubernetes should be able to handle it without any impact.
* This can be achieved by increasing Replicas Set value. We just need to change the Yaml file and increase the replica set value.
* In general, replicaset.yaml or deployment.yaml can have this.We can change this to solve the purpose.