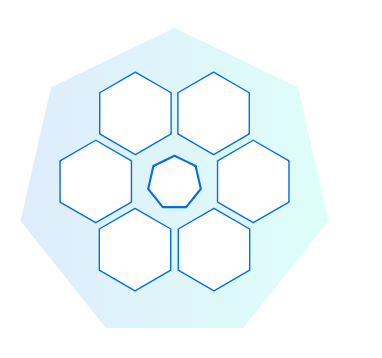
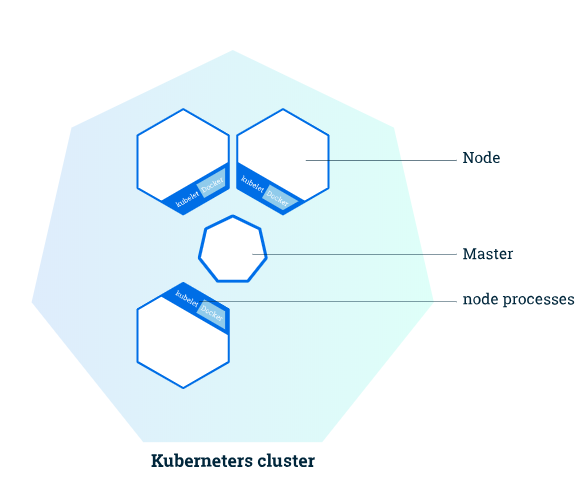
Let’s see and understand everything about Kubernetes.

Kubernetes Platform looks like this from crow-eye view:



**Meaning of the above diagram**: Kubernetes Platform is a highly available cluster of computers that are connected to work as a single unit. So, all the hexagons in the diagram are interconnected computers. In the middle we have a master computer which is controlling and managing the cluster and all others are node computers also known as worker computers (Actual application deployment happens on these node/worker computers).



As we are using **Minikube** – which is a one node cluster therefore within this one node there are different segregations (referred as namespaces) one of them acts as a Master and other one as Worker Node. In next chapter we will see the deployment part, the deployment on minikube happens on the Worker Node segregation.

**In general:**

The **Master** is responsible for managing the cluster. The master nodes will coordinate all the activity happening in your cluster like scheduling applications, maintaining their desired state, scaling applications and rolling new updates.

A **Node** is a VM or a physical computer that is used as a worker machine in a Kubernetes cluster. Every node from the cluster is managed by the master. On a typical node you will have tools for handling container operations (like **Docker**, rkt) and **Kubelet**, an agent for managing the node. A Kubernetes cluster that handles production traffic should have a minimum of three nodes, that’s why in the diagram we see 3 nodes ☺

Communications between the master and the nodes is done via an **API** exposed by the master. The same API is exposed towards the users in order to facilitate interaction with the cluster.

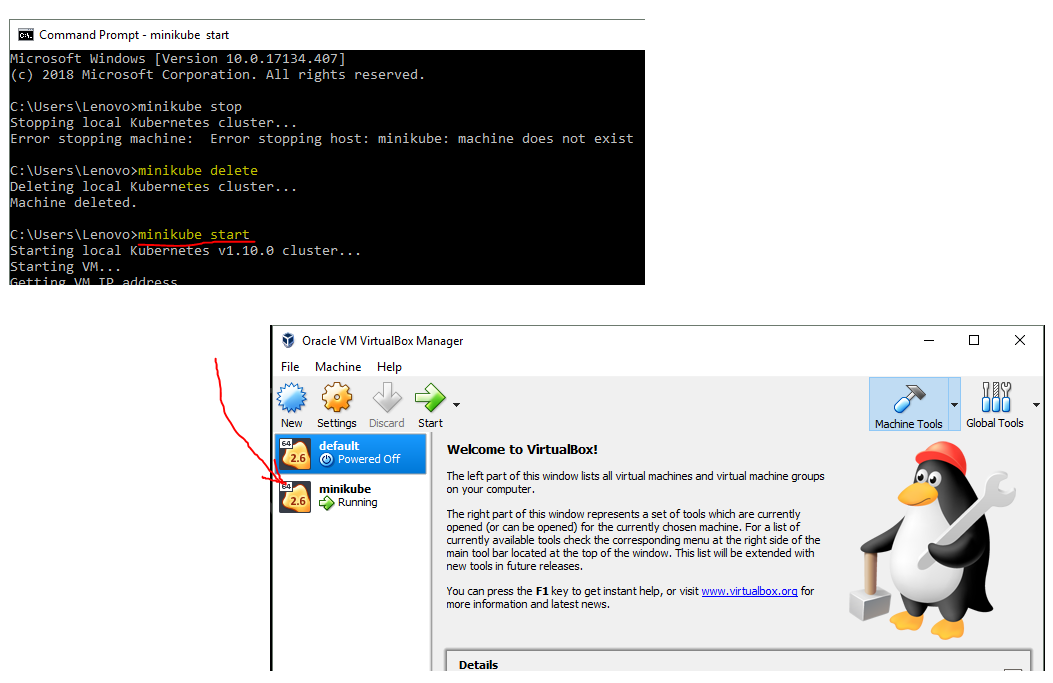
**EXERCISE**

To start the cluster, we use the below command:

minikube start

You are now have a running Kubernetes cluster (one node) in your terminal. **Minikube started a virtual machine for you, (you can see it on Oracle VM) and a Kubernetes cluster is now running in that VM.**

To interact with Kubernetes we’ll use the command line interface, **kubectl**.



minikube stop –To stop the Kubernetes

minikube delete- To delete the Kubernetes cluster (Running on VM)

kubectl cluster-info

We have a running master and a dashboard. The Kubernetes dashboard allows you to view your applications in a UI

kubectl get nodes

This command shows all nodes that can be used to host our applications. Now we have only one node (minikube- a cluster of 1 node), and we can see that it’s status is ready (it is ready to accept applications for deployment).

