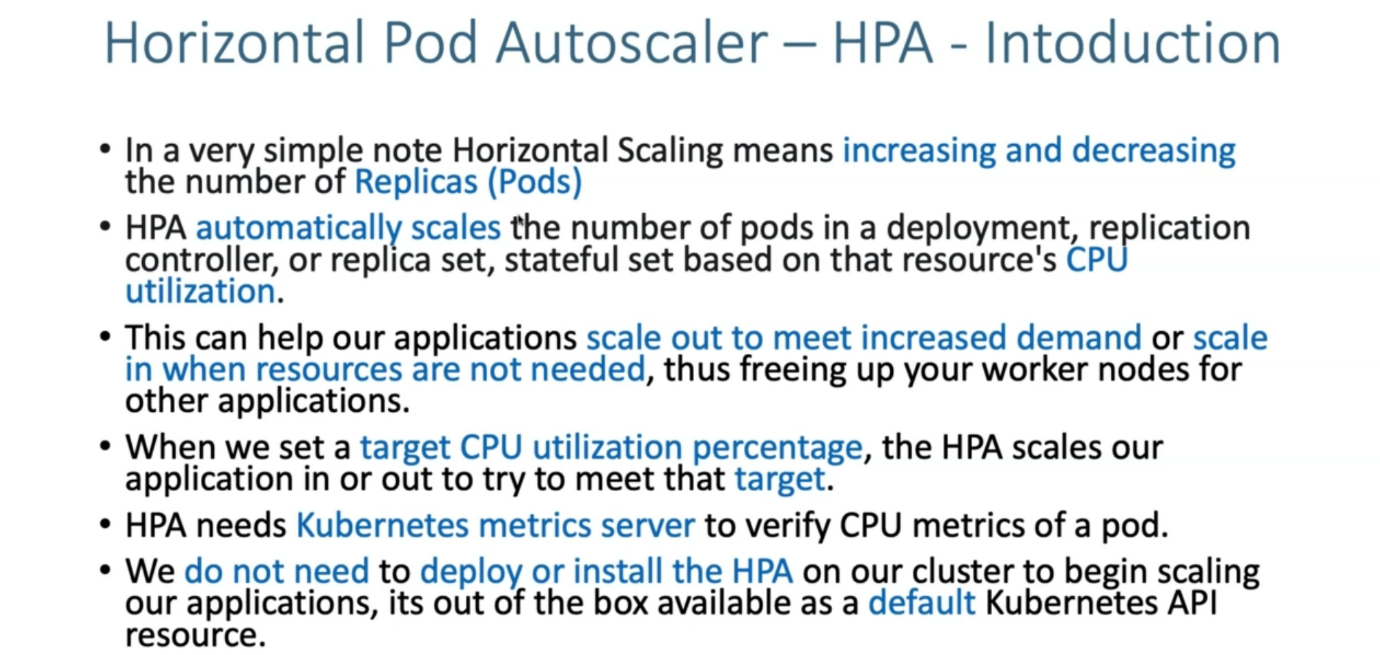
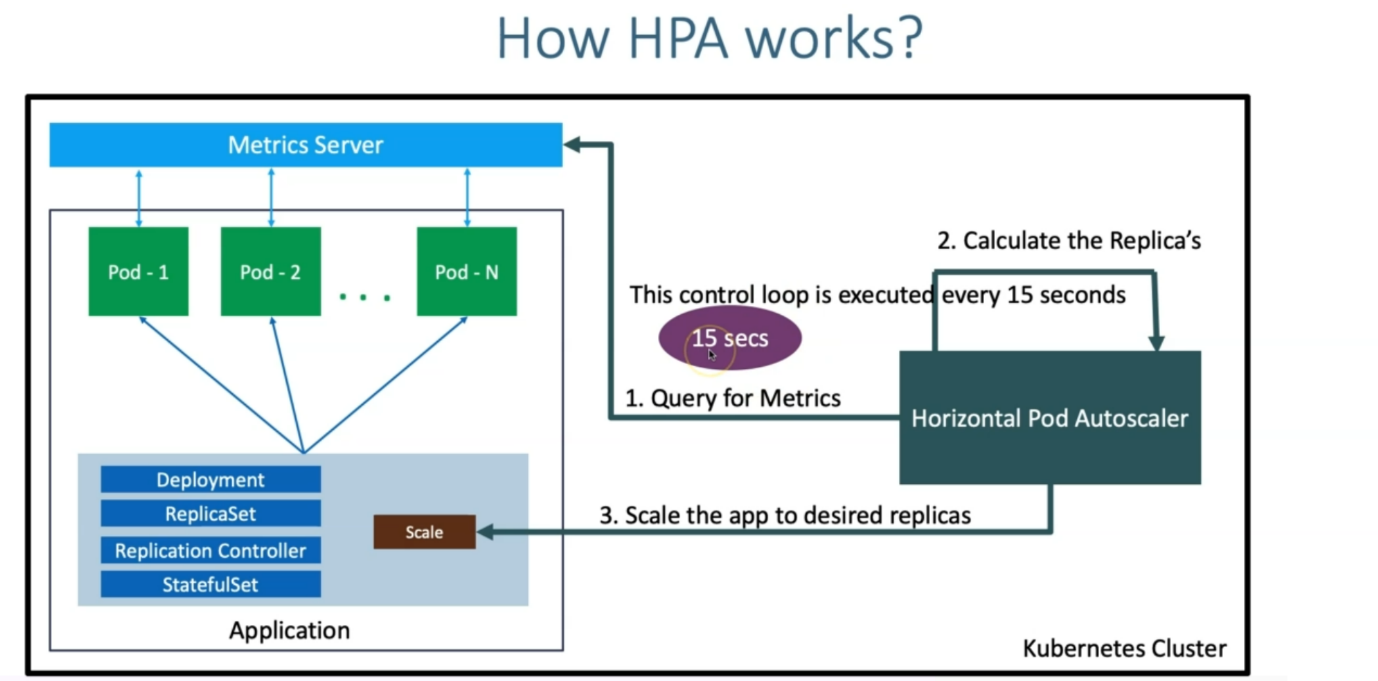
**01. Horizontal Pod Autoscaler - Introduction**

--- In this lecture, we are going to understand about horizontal Pod auto scaler.

--- let's see what is horizontal pod auto scaler. in a very simple, horizontal scaling means increasing and decreasing the number of replicas.



**HOW HPA works…?**



--- in kubernetes cluster, you have deployed application. it might be off type deployment or replica set. Replication controller or stateful set. it is going to span the pods.

--- it is going to create the equilent pods where it is going to deploy our application container. we are also going to deploy default kubernetes metric server available for us.

--- this pod metrics information is going to be sent to our respective Metrics server.

--- when we enable the horizontal Pod auto scalar for our respective application. what happens is like, it is going to query for metrics and once it gets the metrics, it is going to calculate the number of replicas, this horizontal Pod auto scalar need to increase or decrease.

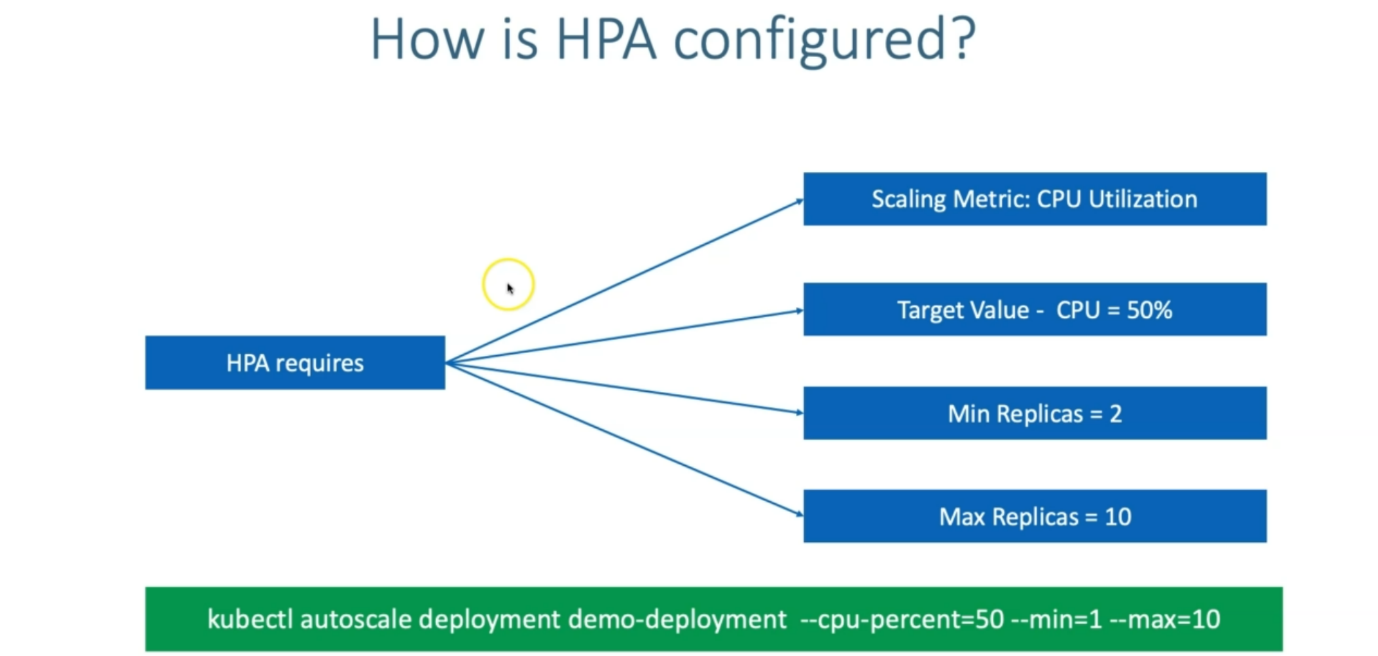
--- then it will scale the app to desired replicas.

--- there will be a kubectl scale option Internally, it is going to be use the scale operation to scale in or scale out our number of ports.

--- once it's scaled, so it is going to either increase or decrease the number of pods in our respective application.

--- this whole thing, whatever happens here, this horizontal pod auto scale in, collecting the metrics, calculating that replaces and then if required, it is going to trigger the scaling request. all this happens is called a control loop and this control loop is executed every 15 seconds.

**How HPA configured**



--- how HPA is configured in our cluster.

--- another important thing, which we need to know is HPA requires scaling metrics. Example you can take our scaling metrics as C.P.U utilization and target value. In your metric, what should be your target value? When it reaches the target value then it can do a scaling in or scaling out, whatever is required based on that.

--- if it reaches 50 percent, then scale out moments like, a greater number of pods need to be added.

--- if the target value C.P.U is 50 percent and then the minimum replicas in my application can be 2 and the maximum replicas can be 10.

--- whenever it triggers the scale out request, it can go up to 10 replicas and whenever it is in the cooling period and then it wants to cool down, it can come back to 2 replicas and then leave the application with two replicas.

--- In our example, we are starting with minimum s one replica and then we'll go to 10 replicas.

--- the command looks like above.