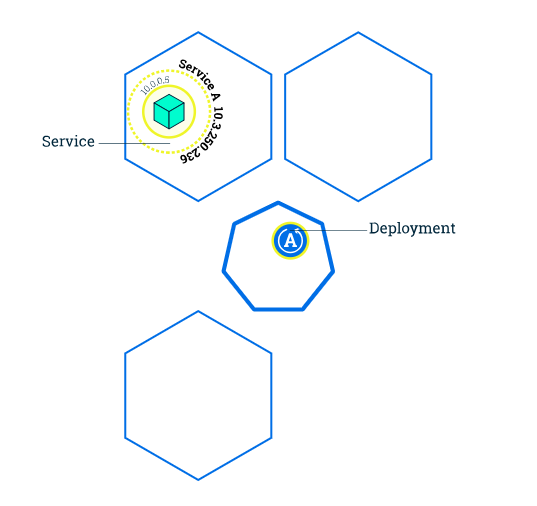
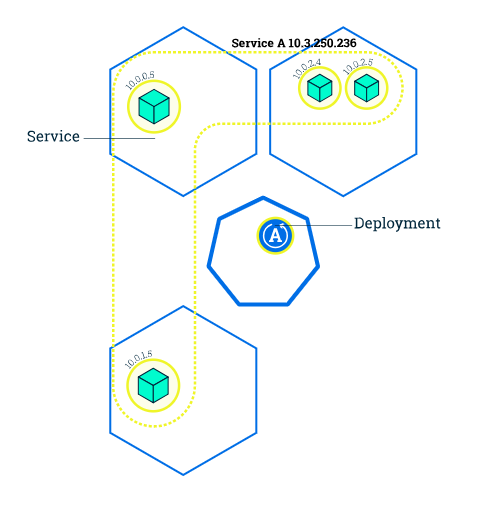
Earlier we have created a [Deployment](http://kubernetes.io/docs/user-guide/deployments/), and then exposed it publicly via a [Service](http://kubernetes.io/docs/user-guide/services/).

**EARLIER:** The Deployment created only one Pod for running our application. When traffic increases, we will need to scale the application to keep up with user demand.





**NOW:** New Pods are created and scheduled to Nodes.

Scaling Up - new Pods are created and scheduled to Nodes with available resources. Scaling Down - Reduce the number of Pods to the new desired state.

Kubernetes also supports Scaling to zero and [autoscaling](http://kubernetes.io/docs/user-guide/horizontal-pod-autoscaling/)of Pods, but it is outside of the scope of this tutorial.

Running multiple instances of an application will require a way to distribute the traffic to all of them. Services have an integrated load-balancer that will distribute network traffic to all Pods of an exposed Deployment. Services will monitor continuously the running Pods using endpoints, to ensure the traffic is sent only to available Pods.

*Scaling is accomplished by changing the number of replicas in a Deployment.*

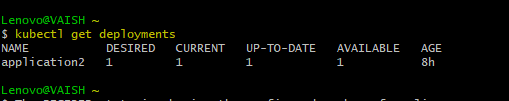
**PRACTICAL:**

The **DESIRED** state is showing the configured number of replicas

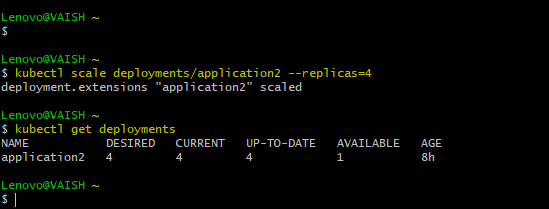
The **CURRENT** state show how many replicas are running now

The **UP-TO-DATE** is the number of replicas that were updated to match the desired (configured) state

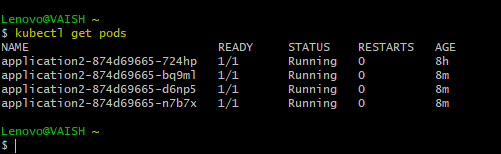
The **AVAILABLE** state shows how many replicas are actually AVAILABLE to the users



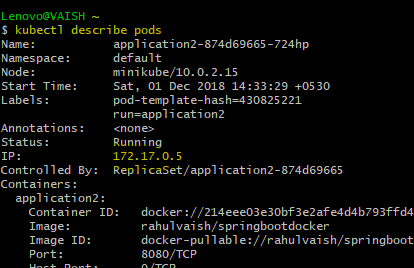
Now we are scaling. Post that checking the deployment status.



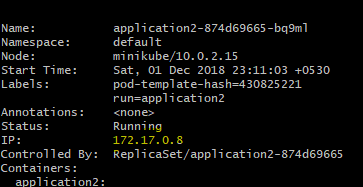
As mentioned above that Scaling Up means new Pods are created, let’s check that…



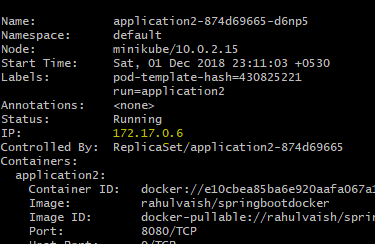
POD#1



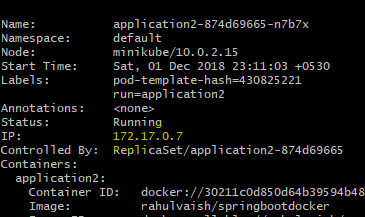
POD#2

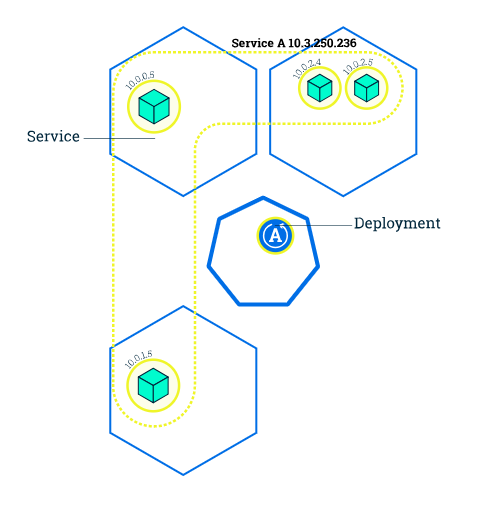


POD#3



POD#4





This diagram is present in the beginning of this document. Now, we can see those 4 pods in the CLI and in the diagram.