**Solution Documentation**

Pre-requisite

* Kubernetes cluster (min 1 Master and 2 worker notes)
* Install docker
* Linux Environment
* Git Hub

Configuration instructions

* Package “Hello World” web application into a Docker image.
* Upload the Docker image to Artifact (Git hub) Registry.
* Deploy the sample app to the Kubernetes cluster.
* Manage autoscaling for the deployment.
* Expose the “Hello World” app to the internet.
* Deploy a new version of the sample app.

1. Deploy the webapp

From your docker container deploy the YML (deployment container)Hello world

***kubectl apply -f hello-world.yaml***

confirm that it worked”

***kubectl get deployments***

1. To expose your application and make it accessible from the outside run:

***kubectl expose deployment/hello-world --type="LoadBalancer" --port 8080***

confirm that it worked:

***kubectl get services***

**Auto Scaling**

With my deployment file, I have described the **auto scaling service**, so the application will scale up or own depending on load or traffic. I have also setup **resources**, which will escribe my resource limits

Monitoring

For monitoring, I have decided to go with two application monitors  
- Kubernetes Dashboard.

* Heartbeat on Elasticsearch

Kubernetes Monitoring

Kubernetes dashboard will give me something like below  

            Kubernetes Dashboard
        

Installation is simple an one can use the below comman

***kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.0.5/aio/deploy/recommended.yaml***

Monitoring with Heartbeat ELK

Graphical user interface, application, table, Excel

Description automatically generated

Configure the ***heartbeat.yml*** an insert the url for the “hello-world”

Assuming that ELK is installed. This will track an monitor your http traffic