Kubernetes Configuration:

Google Cloud Account:

Login to google account then go to this link:

<https://console.cloud.google.com/>

Cluster: Master Node(s) : Worker Node(s)

Kubernetes Engine Dashboard:

Create a cluster. Take standard setting choice.

Connect to the cluster with cloud shell.

// Delete the cluster whenever not needed.

Launch the cloud shell:

This is the most important deployment command to me, it will replace the old deployment with this new one and replicaset will do the rest to move from old deployment to the new deployment.

1. kubectl set image deployment hello-world-rest-api hello-world-rest-api=in28min/hello-world-rest-api:0.0.2.RELEASE

Local Machine:

1. from eclipse maven to build docker image.
2. Docker push project-tag to upload image to dockerhub.
3. docker push netwave8888/mmv2-springboot-web-angular:0.0.1-SNAPSHOT ( for some reason, can only push when the first part is username )
4. Deploy the image to google cloud using Kubernetes.
5. Online commands:

kubectl create deployment currency-conversion --image=in28min/mmv2-currency-conversion-service:0.0.11-SNAPSHOT

kubectl expose deployment currency-conversion --type=LoadBalancer --port=8100

1. Install Gcloud and kubectl to the local machine.
2. From google cloud console, connect and copy the gcloud command. gcloud container clusters get-credentials netwave-cluster --zone us-central1-c --project myproject-sept14-2022
3. From local console, use copy command on the cluster page to connect to the cluster.
4. <https://github.com/in28minutes/spring-microservices-v2/tree/main/05.kubernetes>
5. Login to google cloud account command line: gcloud auth login
6. Env variables for service automatically created:

Service\_name : Service\_name\_service\_host

Kubernetes Declarative Configuration: ( To set up yaml file on the local console)

1. Use this command to get initial deployment yaml file

Kubectl get deployment currency-exchange -o yaml >> deployment.yaml

1. Kubectl get service currency-exchange -o yaml >> service.yaml
2. Merge deployment.yaml file with service.yaml file, clean up all the timestamp and status stuff.
3. kubectl apply -f deployment.yaml
4. kubectl autoscale deployment currency-exchange --min=1 --max=3 --cpu-percent=5
5. all the Kubenetes commands from the above link to in28mins.

Angular 14 Bootstrap Configuration:

Example: Angular14projects/bootstrap-demo

Link:

<https://efficientcoder.net/add-bootstrap-angular-14/>

1. ng new bootstrap-demo --routing
2. $ npm install --save bootstrap jquery ( installing bootstrap 5, this is wrong with jquery will corrupt bootstrap )
3. "architect": { "build": { [...], "styles": [ "src/styles.css", "node\_modules/bootstrap/dist/css/bootstrap.min.css" ], "scripts": [ "node\_modules/jquery/dist/jquery.min.js", "node\_modules/bootstrap/dist/js/bootstrap.min.js" ] },
4. Continue with the project.

Install Bootstrap 5 to Angular 14:

1. Ng new bootstrap-items –routing
2. Cd to project folder and npm install bootstrap –save
3. @import "~bootstrap/dist/css/bootstrap.css"; src/style.css