**Debug and Set up commands**

**Changes in Frontend web application:**Make sure to run: flutter build web --release   
This command ensures the Web release mode is created in the build folder with the newest changes you made in your code. Then push the code to the repo so CI/CD takes care of the rest and deployment.

**Image pull failure by Kubernetes from GHCR:**When the Image pull fails make sure to run the following command to check if your Personal Access Token you created in the Github and added it in the secret with the name GHCR\_TOKEN is containing the right information.   
Command: kubectl get secret ghcr-secret -n default -o jsonpath="{.data.\.dockerconfigjson}" | % { [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String($\_)) }

Payload should Contain the auth credentials and the PAT. See below:  
{"auths":{"ghcr.io":{"username":"erfanalizada","password":"ghp\_ybTMZCtJjU9U3GFLUibgIDoucGR5h82OVaH4","auth":"ZXJmYW5hbGl6YWRhOmdocF95YlRNWkN0SmpVOVUzR0ZMVWliZ0lEb3VjR1I1aDgyT1ZhSDQ="}}}

**Application changes on web or backend not visible yet:**After passing CI/CD it takes a while until the pods restart with the new changes and terminate the older pods. Therefore after CI and CD passes you need too wait from 30 seconds up to 60 until you see the changes.   
In order to see the pods running and their status run the following command:  
kubectl get pods -n default -w

When willing to see the changes on web make sure to hard refresh the page using Ctrl+Shift+R. Otherwise the cache will keep using the older downloaded version of you app and you won’t be able to see the changes. Other way is to open an Incognito Page/Private web page and open your web link to see the changes.

**Getting your application web link:**Run the following command to find the public IP where your application is accessible:  
kubectl get svc -n default

**When your Kubernetes Secret is outdate do the following to resolve issue:**Make sure to delete the current old secret by running the following command:  
kubectl delete secret ghcr-secret -n default

Then run the following command to regenerate the new Secret. **Keep in mind** that you need to add your new PAT and your github username and keep the namespace (-n) on default.  
See command below:

kubectl create secret docker-registry ghcr-secret `

--docker-server=ghcr.io `

--docker-username=YOUR\_GITHUB\_USERNAME `

--docker-password=YOUR\_PERSONAL\_ACCESS\_TOKEN `

--docker-email=YOUR\_EMAIL `

-n default

*Keep in mind to Update the old PAT with new PAT Token in your git Secret with the name GHCR\_TOKEN*

**See the pods in default namesapce:**In order to see all your pods in the default name space run the following command:  
kubectl get pods -n default

**After updating your manifests or created new manifest files in k8s folder**

You need to run the following command to apply the updates and new files so Kubernetes can use it: kubectl apply -f k8s/

*What it does:*

* If a resource (Deployment, Service, Ingress, etc.) doesn’t exist, it creates it.
* If it exists, it updates it to match your YAML.  
  *When to use:*
* Every CD run → this is how you “deploy” your new image, configs, or replicas.

**To get detailed info about a pod:**Run the following command following with the name of pod:  
kubectl describe pod <pod-name>

*What it does:*

* Shows pod spec, conditions, and events (like why pulling image failed).  
  *When to use:*
* When a pod is stuck (Pending, ImagePullBackOff).
* To debug issues like missing secrets, wrong image name, insufficient CPU/memory, etc.