Improving the K8S API Operator First Hand Experience

Problem :

As of now, API Operator requires a lot of configurations to be included if someone needs to try it out. This breaks the user experience and will lose the interest of the users towards the API Operator.
Following are some of the noticed issues.

1. *DockerHub* account configuration.
    a. Users have to go through multiple editing in configuration file
    b. Users have to base64 encode the values
2. Download multiple distributions (*API Operator* distribution and *apictl* distribution)

Solution :

To enhance the user experience for the PoC/Quick start/demo users so that they can deploy an API with minimal configurations and minimal downloads of distributions.
For this, we can integrate the configuration of API Operator with *apictl* tool.

Approach :

This is mainly focused on the users who are trying out API operators for non-production scenarios.

According to the current implementation, using Kaniko is the most suitable and ideal approach. Hence we have to provide a docker registry access(registry(optional), repository (optional), username, password)

We can do the API Operator configuration part in the ***apictl*** itself so that the users would not have to download multiple tools and they would not have to worry(when they try out) about the files that they need to change.

Proposed commands:

Using the OperatorHub.io users can make the API Operator up and running.

Introducing command for Initializing API Operator

```
```

apictl init k8s
````

On this command, we can automatically execute the operatorhub.io commands and make the API Operator up and running in the Kubernetes cluster.

After executing the above command, it would prompt user to enter the following

1. Docker registry url (if you are using a private docker registry, the user has to provide the value. Otherwise type just press ENTER)
2. Docker username
3. Docker password

Once these things are provided, apictl will create a secret and deployed on kubernetes to be used in the Kaniko job.

By this time,  everything is configured and the operator is up and running.

**TBD**

1. Check if we can inject namespace,configurations related configmaps and secrets to the Operatorhub.io
2. If not, the possibility to deploy the artifacts after downloading them or pointing to a downloaded directory.
3. Or, ship artifacts with the apictl itself as a single yaml file. Upon the init, it would deploy this yaml file in the kubernetes cluster.
````