

OpenAPI

Rest APIs are the most common and effective method of communication in an Internet environment. In addition, there are various ways to enhance security, such as SSL and JWT. Therefore, most applications are designed and operated on an API basis. OpenAPI allows you to standardize this communication, and using these standardized API specifications makes it easy and fast to develop services.

What is OpenAPI?

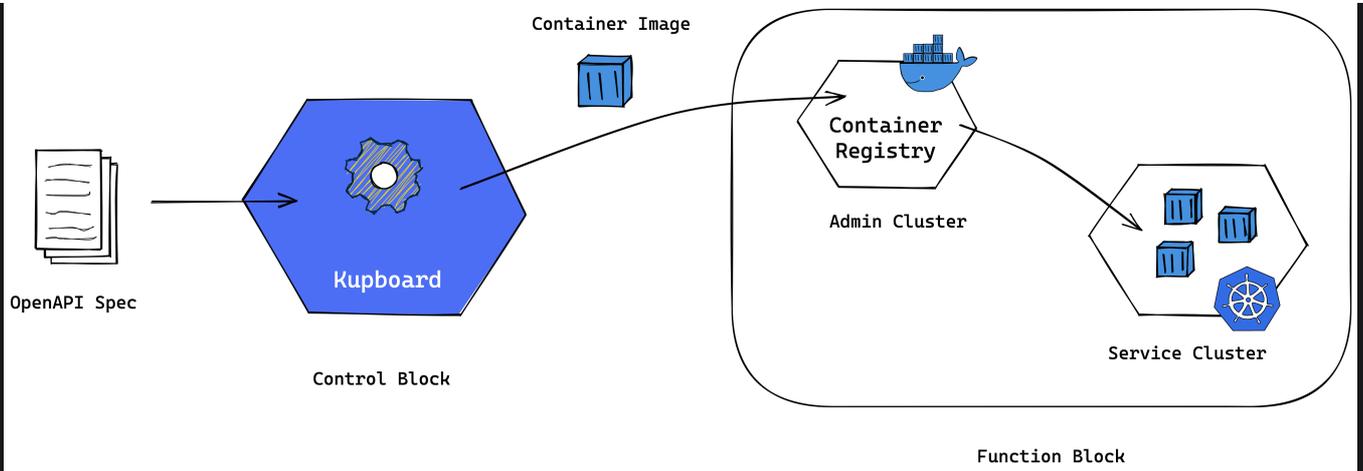
The OpenAPI Specification (OAS) defines a standard, language-agnostic interface to RESTful APIs which allows both humans and computers to discover and understand the capabilities of the service without access to source code, documentation, or through network traffic inspection. When properly defined, a consumer can understand and interact with the remote service with a minimal amount of implementation logic.

An OpenAPI definition can then be used by documentation generation tools to display the API, code generation tools to generate servers and clients in various programming languages, testing tools, and many other use cases.

[Open API 3.0 Specification](#)

OpenAPI with Kupboard

If you have a specification written for the OpenAPI format, Kupboard can build and deploy service containers without any development. It is even possible to deploy a test server that generates and returns random data.



Demo

[kupboard-data-sample.tar.gz](https://github.com/kupboard/kupboard-data-sample) contains `data/api/demo.yaml` and this is an API specification based on OpenAPI 3.0. `/v1/demo/list` is defined in the file.

```
data
├── api
│   └── demo.yaml
```

Build

Use the `build` command to deploy your application using the OpenAPI specification located in the `data/api` directory. And to deploy a mockup application that provides random data, use the `--mock` option together. Container images for these are automatically pushed into the Harbor registry and used for deployment.

```
$ kupboard.sh build -s demo [--mock]
```

In order to expose applications deployed in the above ways, a gateway must also be deployed. To build a gateway, use the service name `gateway`.

```
$ kupboard.sh build -s gateway [--mock]
```

NOTE

Built and deployed with the `build` command, `gateway` is an application deployed in Kubernetes. This is different from a gateway cluster.

Deploy

When you complete building your OpenAPI application and gateway, you can deploy them using the `deploy` command. As with the `build` command, you must deploy applications and gateways for mockup using the `-mock` option.

```
$ kubernetes.sh deploy -s demo [--mock]
```

```
$ kubernetes.sh deploy -s gateway [--mock]
```

NOTE

After the OpenAPI application and gateway are deployed, the Nginx package must be installed in the gateway cluster for external access. Once Nginx is installed, the applications can be accessed by `app.mycompany.com` and the mock application can be accessed by `mock.mycompany.com`.

```
$ kubernetes.sh collection package -n nginx
```

 [Edit this page](#)