


# Kubernetes Operator for API Management

---

Lakmal Warusawithana

 @lakwarus

# Kubernetes

- Automating deployment
- Automating scaling
- Management of containerized applications.

# API Management

- **Access Control and security** — Control who can access and how can access
- **Rate limiting** — How many requests can user or application do?
- **Monitoring** — How API is doing?
- **Self-service portal** — Help developers to discover and use the APIs
- **Monetization** — Bill and charged for API usage



# Kubernetes Operators

- Kubernetes is designed for automation
- Custom Resource Definitions
- Custom Controllers
- Extend the Kubernetes cluster's behavior without modifying the code of Kubernetes itself.

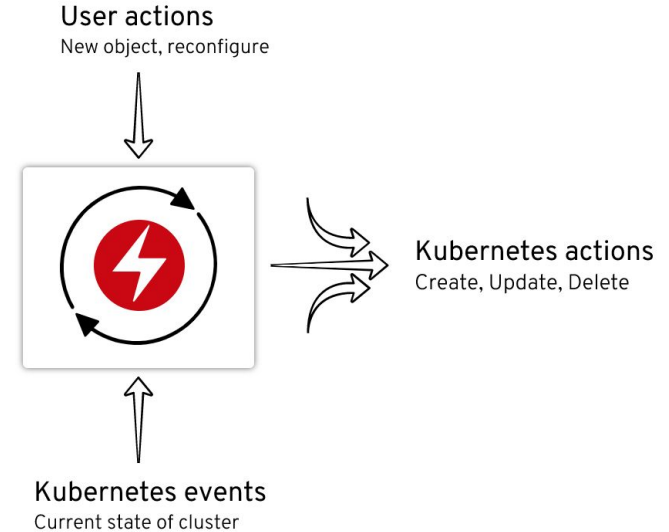
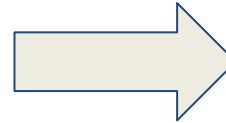
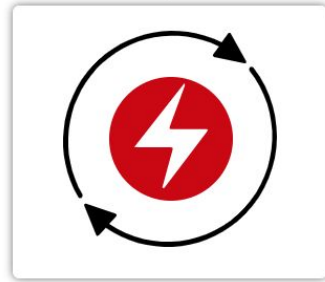


Image source: <https://robszumski.com/building-an-operator/>

# API Operator

API Developer / API Publisher  
API Definition

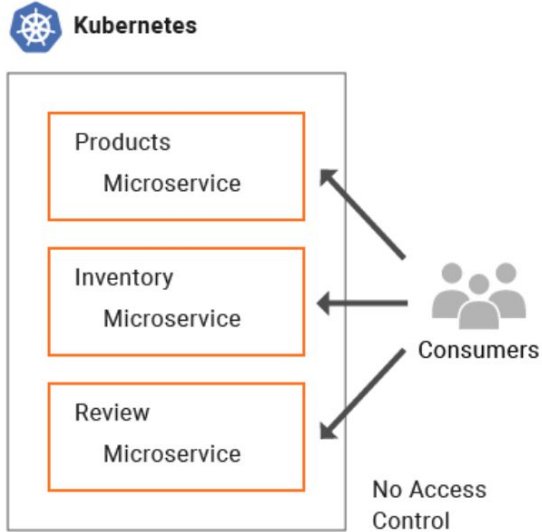


API Gateway  
Managed Services  
Access Control

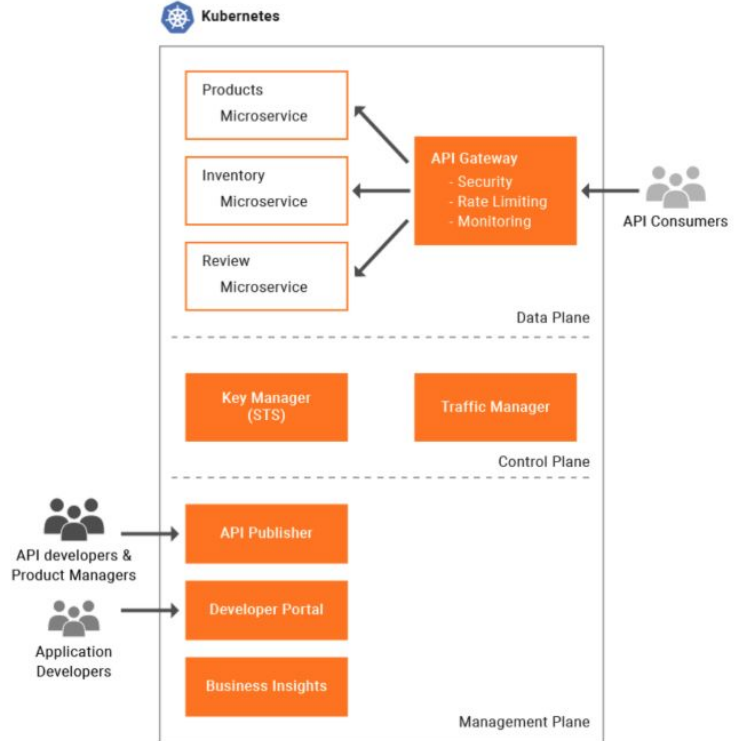
K8s Events + Existing Objects  
Current state

# API Operator

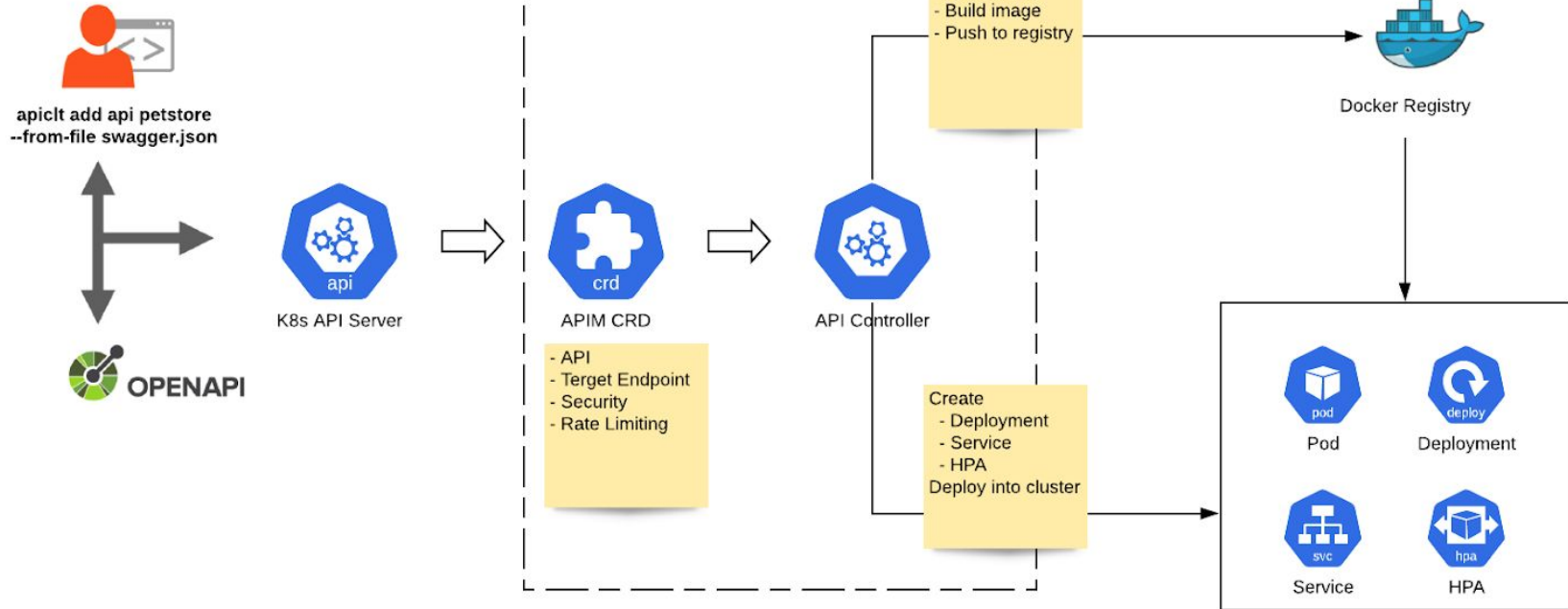
## Without API Operator



## With API Operator



# API Operator

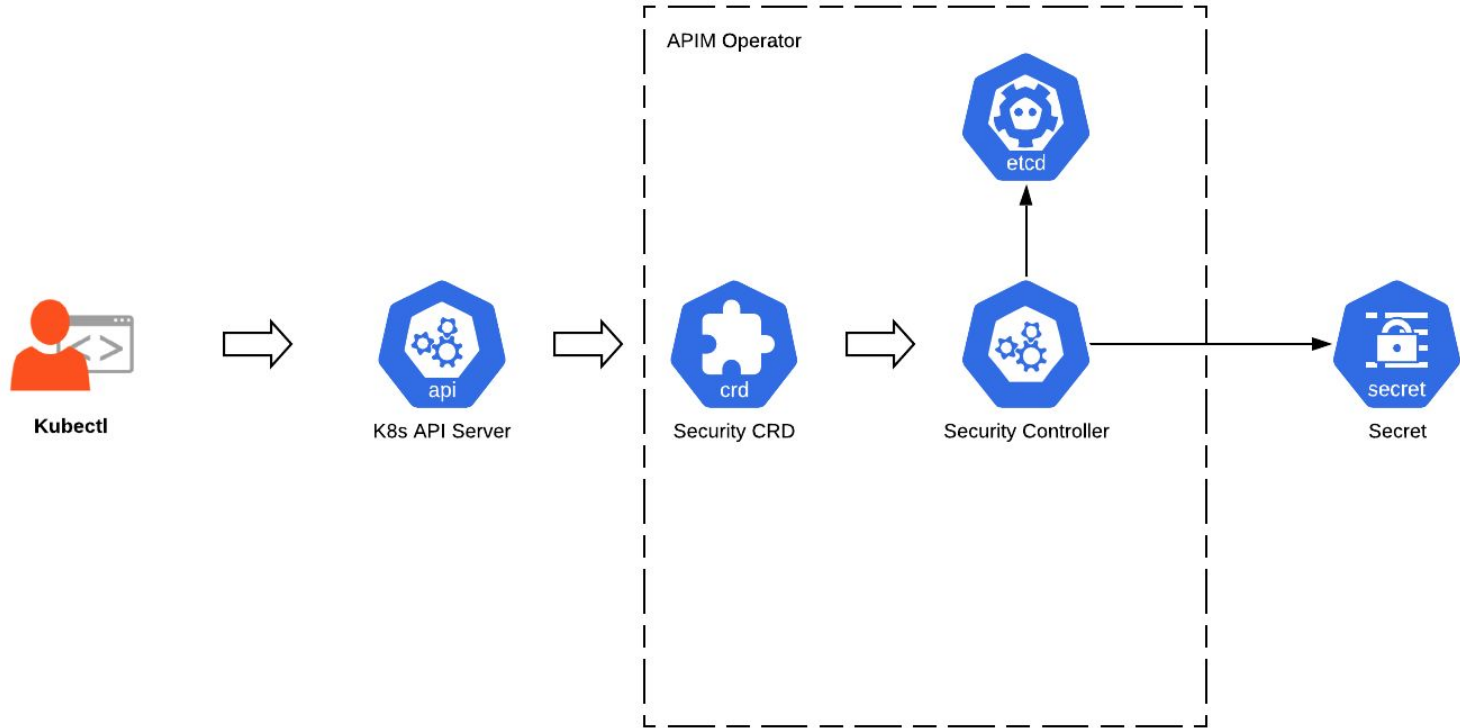


# Custom Resource Definition - Security

```
1  apiVersion: wso2.com/v1alpha1
2  kind: Security
3  metadata:
4    name: petstorejwt
5  spec:
6    # Security - JWT
7    type: JWT
8    issuer: https://localhost:9443/oauth2/token
9    audience: http://org.wso2.apimgt/gateway
10   # Create secret with certificate and add secret name
11   certificate: wso2am-secret
```



# Custom Controller - Security



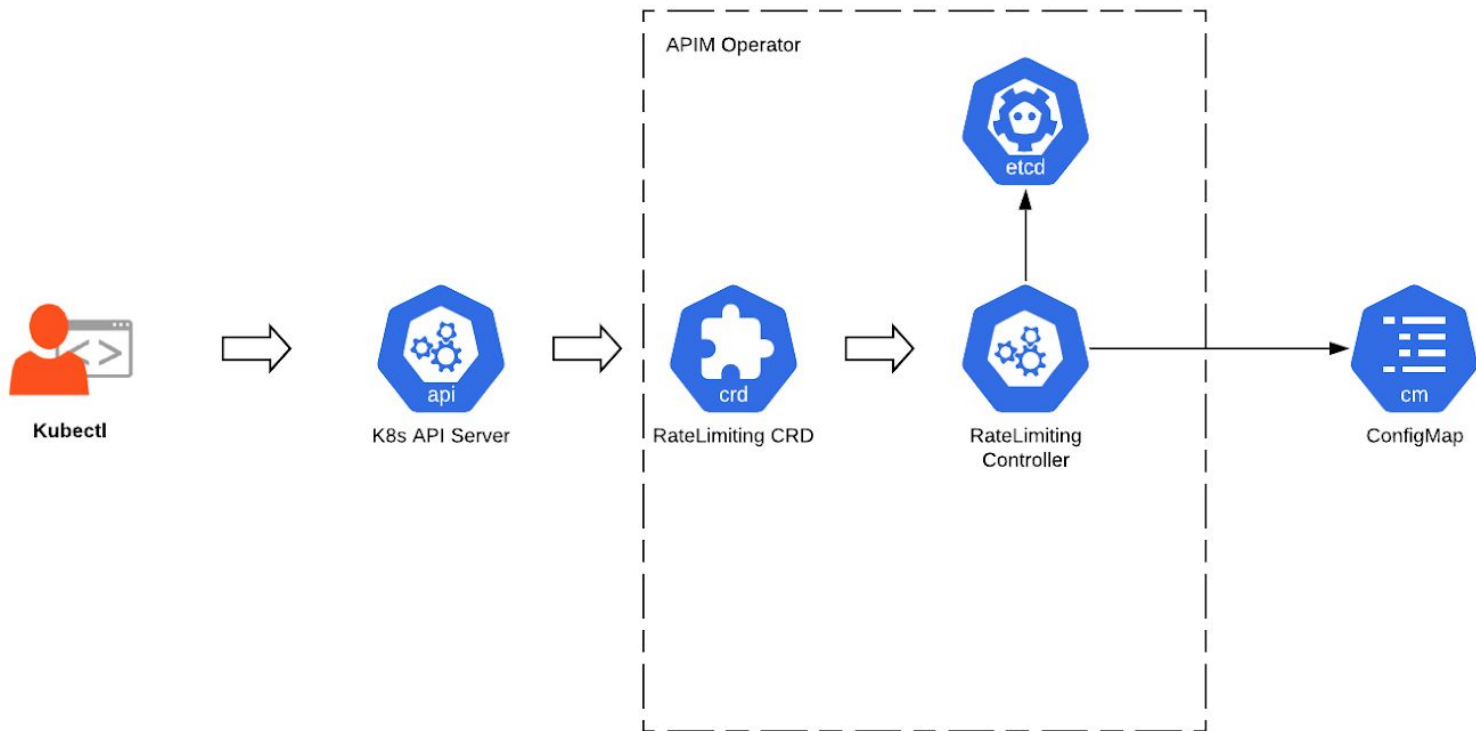


# Custom Resource Definition - Rate Limiting

```
1  apiVersion: wso2.com/v1alpha1
2  kind: RateLimiting
3  metadata:
4    name: app1
5    namespace: app1-ns
6  spec:
7    type: application
8    description: Allow 4 requests per minute # optional
9    timeUnit: min
10   unitTime: 1
11   requestCount:
12     limit: 4
```



# Custom Controller - Rate Limiting

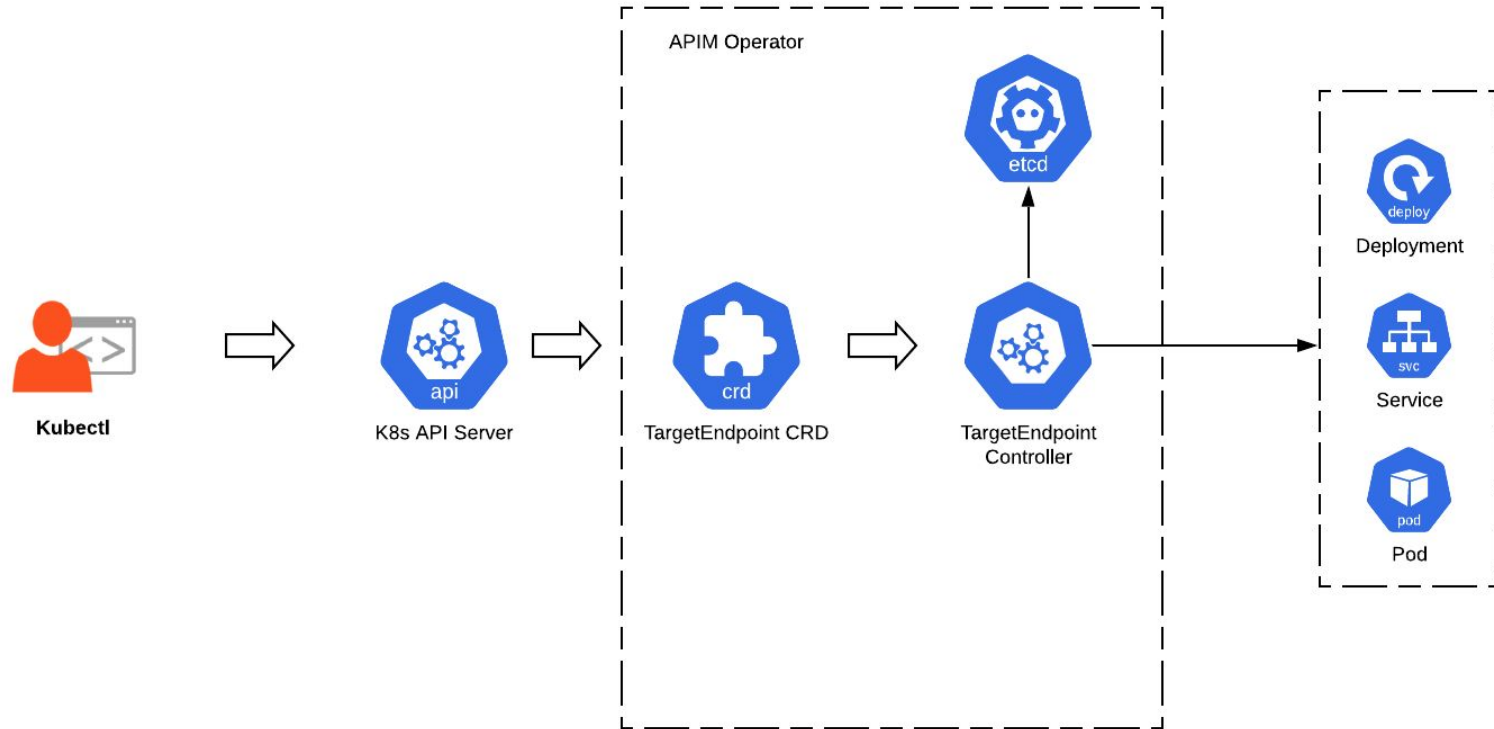


# Custom Resource Definition - Target Endpoint

```
1  apiVersion: wso2.com/v1alpha1
2  kind: TargetEndpoint
3  metadata:
4    name: helloworld-sidecar
5    labels:
6      app: app2
7  spec:
8    protocol: http
9    port: 8080
10   deploy:
11     name: helloworldservice
12     dockerImage: lakwarus/helloworld:v1
13     count: 2
14   mode : sidecar
```



# Custom Controller - Target Endpoint

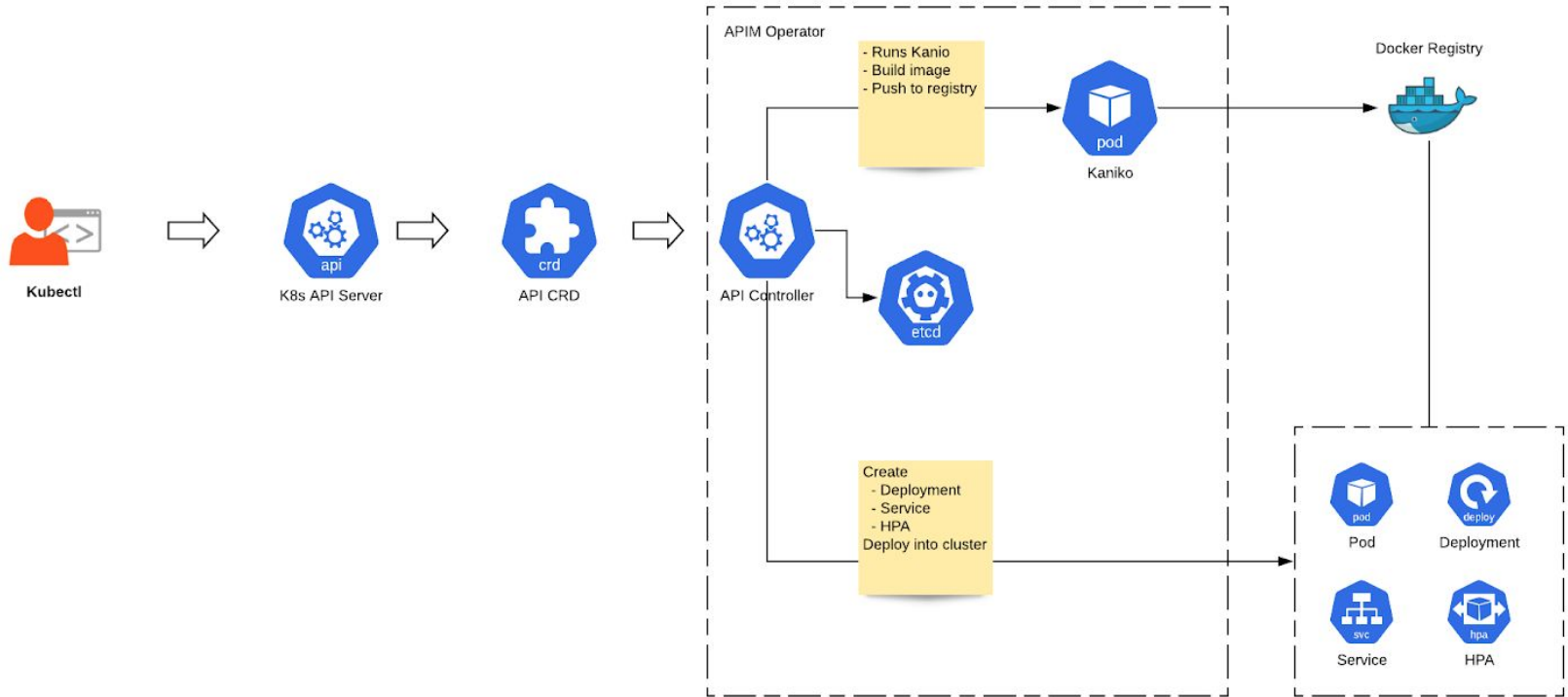


# Custom Resource Definition - API

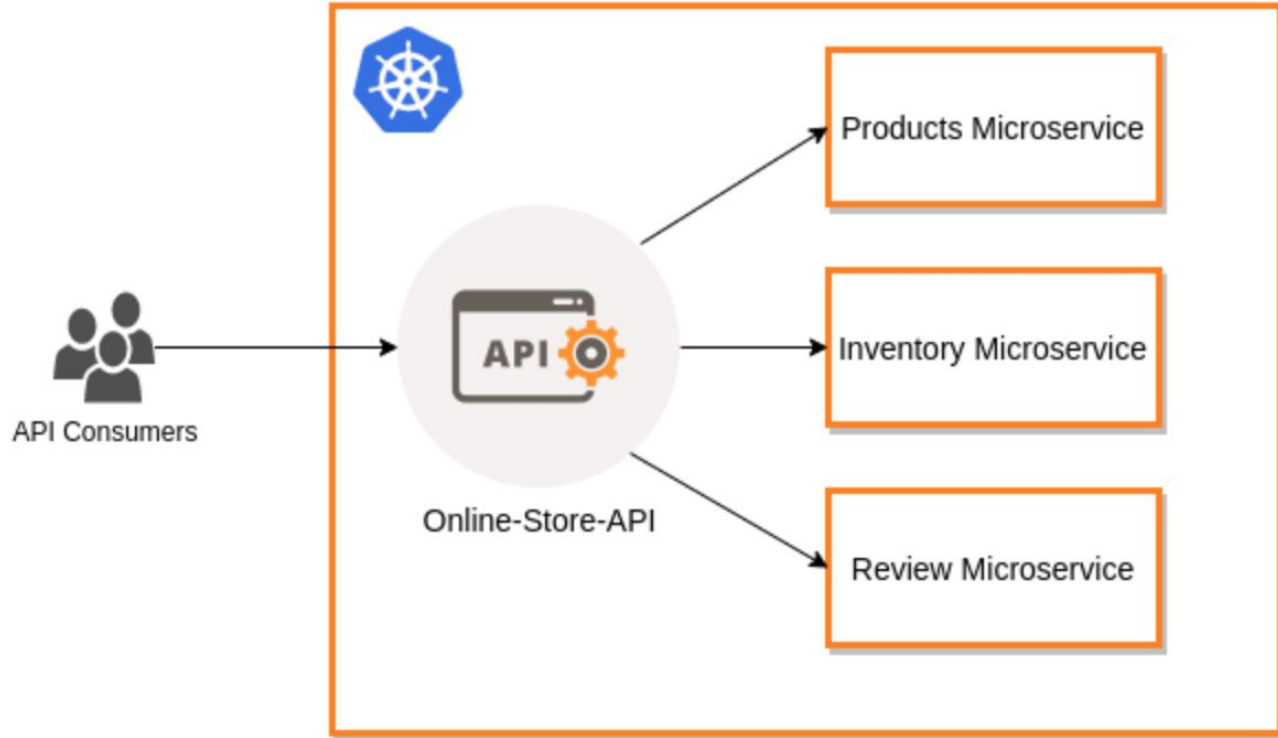
```
1  apiVersion: wso2.com/v1alpha1
2  kind: API
3  metadata:
4    name: "${apiName}"
5  spec:
6    definition:
7      configmapName: "${configmapName}"
8      type: swagger
9      replicas: ${replicas}
10     mode: privateJet
```



# Custom Controller - API



# Demo - multiple microservices endpoints as a single managed API



# References

- <https://github.com/wso2/k8s-apim-operator>
- <https://medium.com/swlh/kubernetes-operator-for-api-management-4f0e56150da6>
- <https://wso2.com/api-management/>





## Key Takeaways

- API Developer, Publisher do not want to learn anything about Kubernetes. They can just focus on API development by using swagger.
- DevOps do not want to worry about automating API gateway deployment for each and every API publish.
- Overall, API Operator will improve API production pipeline by automating and to end process.



# THANK YOU

wso2.com

---

