



KubeCon



CloudNativeCon



China 2024

KCL: Simplifying Kubernetes Manifests Management

Zhe Zong

What does KCL do ?




China 2024

Mutation, Validation, Abstraction, Automation Production-Ready

KCL is an open-source constraint-based record & functional language mainly used in configuration and policy scenarios.

A domain-specific language for infrastructure as code (IaC)

- Manifest Explosion and Cognitive Load :
- Dynamic Configuration Management
- Configuration Reliability
- Easily Automation and Integration



KCL **CNCF** **SANDBOX** **TAG RUNTIME** 📅 2023

Cloud Native Computing Foundation (CNCF)

Provisioning Automation & Configuration

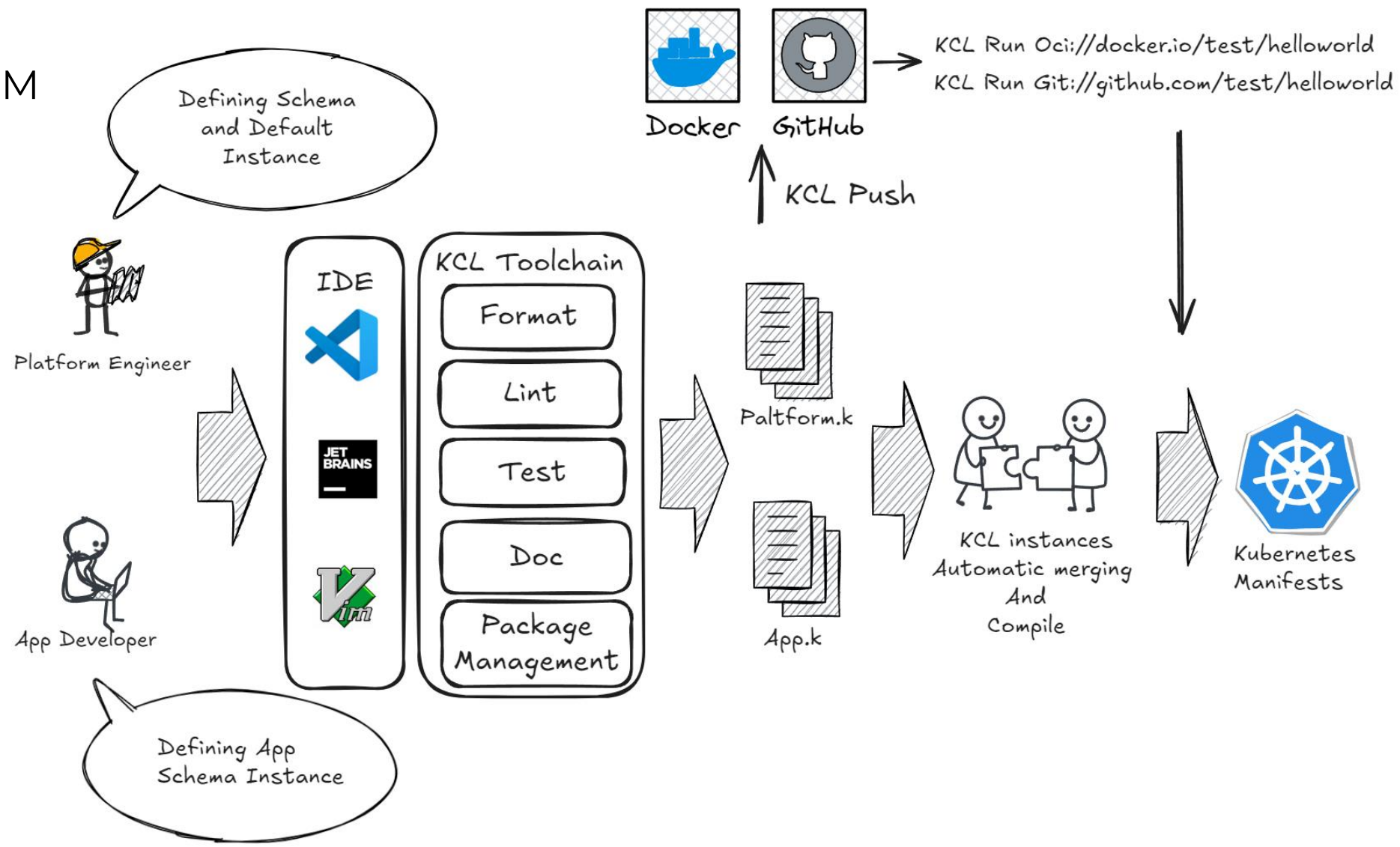
Manifest Explosion and Cognitive Load

Good IDE supports
Vscode, JetBrains and VIM

Use **Schema** to abstract
the repeated manifest

**Automatic
configuration merging**
will reduce the
cognitive load.

Package management
further reuses the
manifests and reduces
the size.

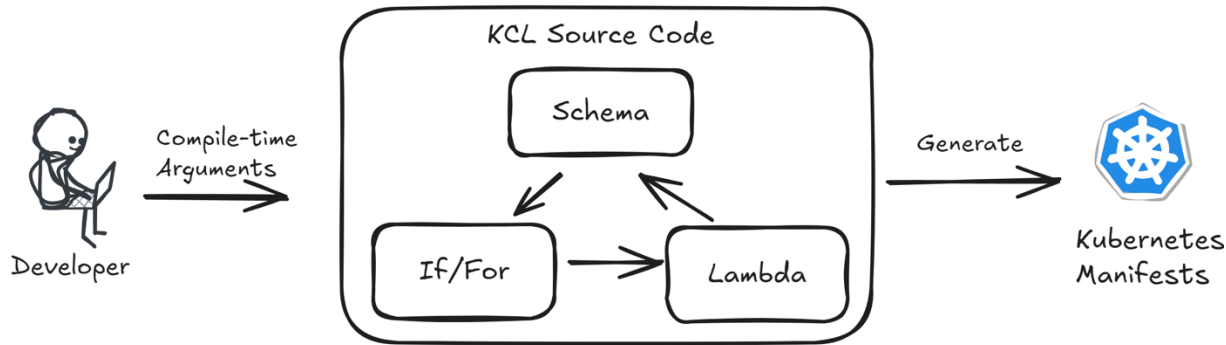


Dynamic Configuration Management

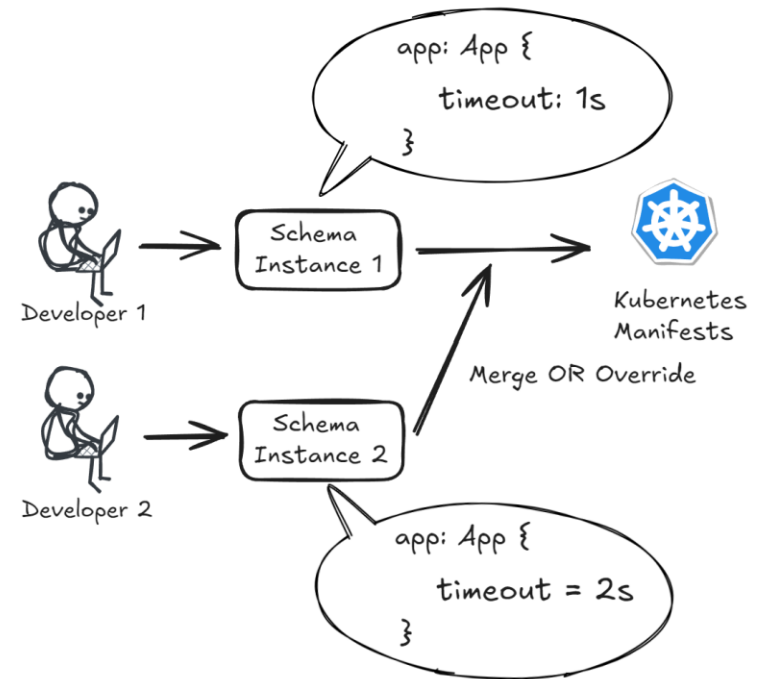


China 2024

Generate configuration based on **arguments** and **if/for/lambda** code logic



Merge or overwrite fields in the configuration through **automatic merging**

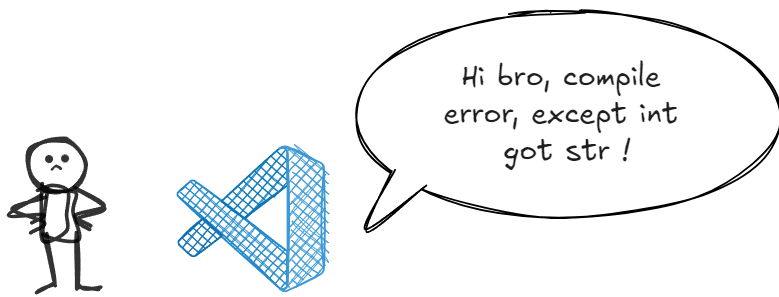


Configuration Reliability



China 2024

The **strong type system** will prompt all type errors in the IDE.

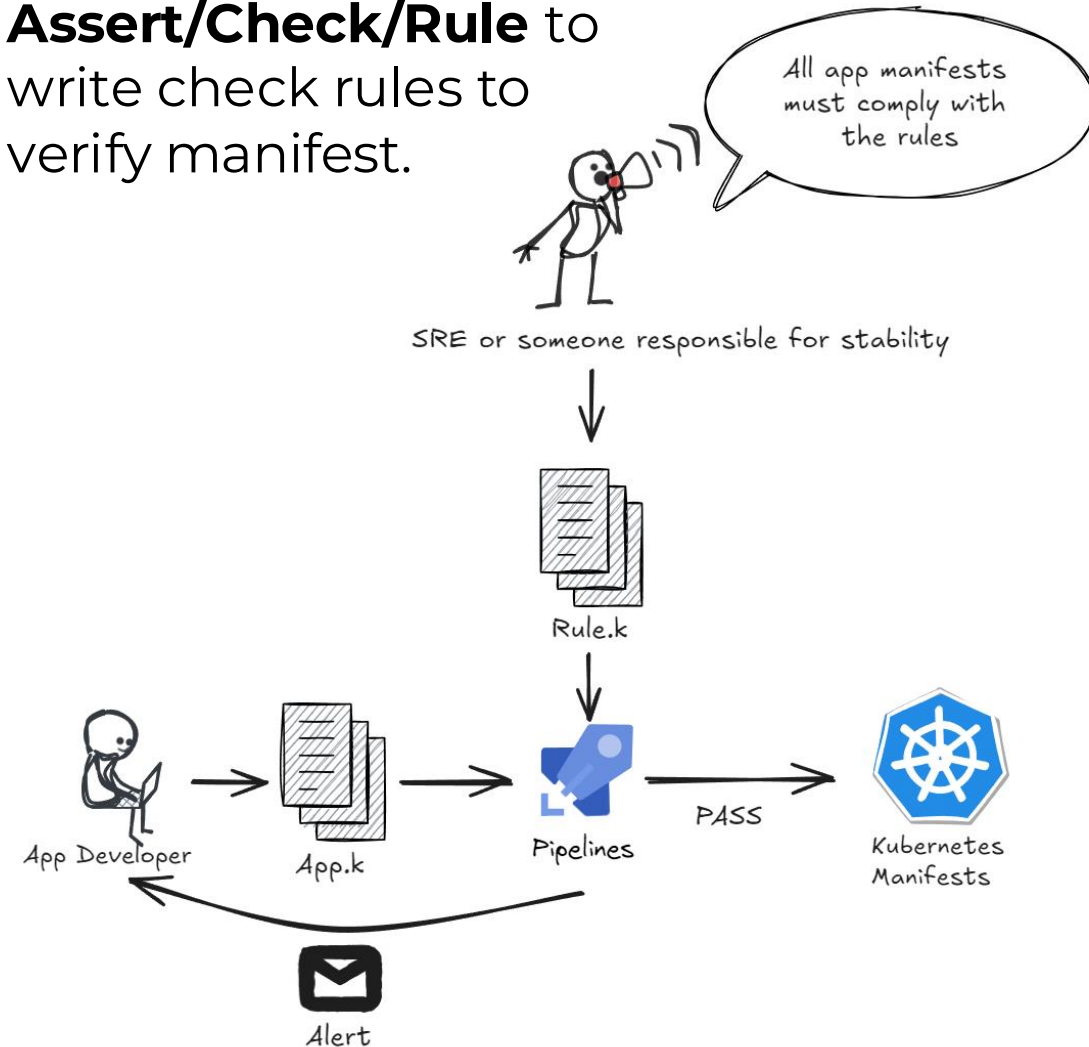


The most effective and simplest way to ensure reliability is to **write test cases**.

```
test_person_age: () -> any = lambda {  
  a = Person{  
    assert a.age == 2  
  }  
}
```

```
zongz@U-2X01KX9Q-2004 sample1 % kcl test  
test_person: PASS (5ms)  
test_person_age: FAIL (7ms)  
EvaluationError  
--> /Users/zongz/Workspace/learn/kcl_learn/sample1/main_test.k:7:1  
7 | assert a.age == 2
```

KCL provides **Assert/Check/Rule** to write check rules to verify manifest.

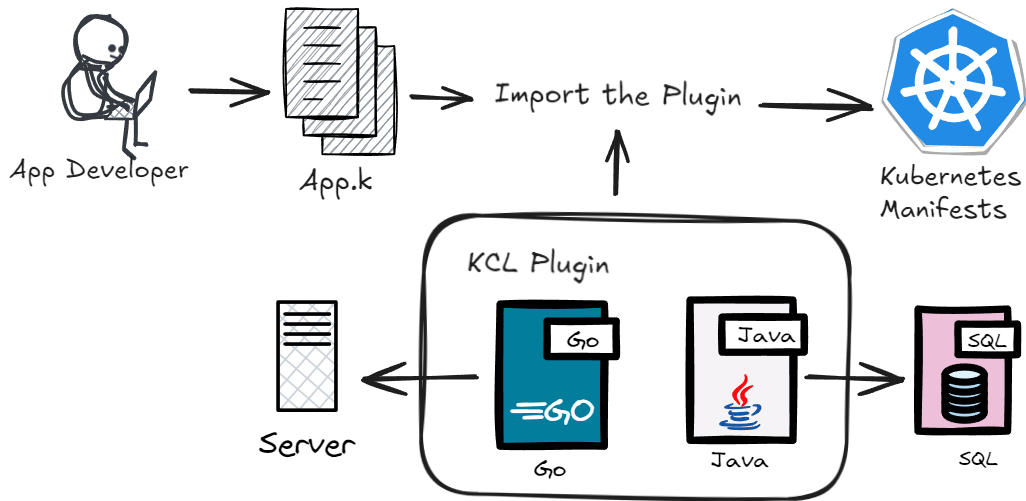


Easily Automation and Integration



China 2024

KCL-Plugin can provide developers more capabilities.



KCL provide **KCL-Operator** to use KCL to verify, mutate and generate k8s resources.

```
apiVersion: krm.kcl.dev/v1alpha1
kind: KCLRun
metadata:
  name: set-annotation
spec:
  params:
    annotations:
      managed-by: kcl-operator
  # Resource modification can be achieved with just one line of KCL code
  source: |
    items = [item | {metadata.annotations: option("params").annotations} for item in option("items")]
```

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  annotations:
    app: nginx
spec:
  containers:
  - name: nginx
    image: nginx:1.14.2
    ports:
    - containerPort: 80
```

```
kubectl get po nginx -o yaml | grep kcl-operator
managed-by: kcl-operator
```

```
1 import kcl_plugin.http
2
3 res: any = http.get("https://kcl-lang.io")
```

Multiple-lang SDK for integration.





- HomePage : <https://www.kcl-lang.io/>
- Github : <https://github.com/kcl-lang>
- Slack Channel: <https://cloud-native.slack.com/archives/C05TC96NWN8>
- Email: kcl-lang.io@domainsbyproxy.com