



Optimizing the Software Development Process with IDP and Kusion

Hoang Dinh Nguyen / Cloud Engineer



Hoang Dinh Nguyen

Hoang Dinh Nguyen works as a Cloud Solution Engineer at Viettel Group, where he is responsible for developing solutions for cloud platform services (Viettel Kubernetes Engine, Cloud Observability, and Internal Developer Platform) on Viettel Public Cloud.

Also working on Kusion (Platform Orchestrator) a first-ever open-source project that included in both the CNCF Sandbox and the Platform Tooling Landscape.

email: hoangndst@gmail.com
linkedin: <https://www.linkedin.com/in/hoangndst>
telegram: <https://t.me/hoangndst>
website: <https://hoangndst.com>

TOC

01. Challenges

Challenges Encountered in
Software Development

02. Kusion

Open-source Platform
Orchestrator for Platform
Engineering

03. Kusion and IDP

Integrate Kusion with your
IDP

Challenges

About the Software Development Process

- DevOps was supposed to be a culture, not a job title.
- Better collaboration between developers and Ops.
- **Reality: shift as much load as possible onto developers.**

NOT SURE IF DEVOPS IS ABOUT COLLABORATION OR JUST

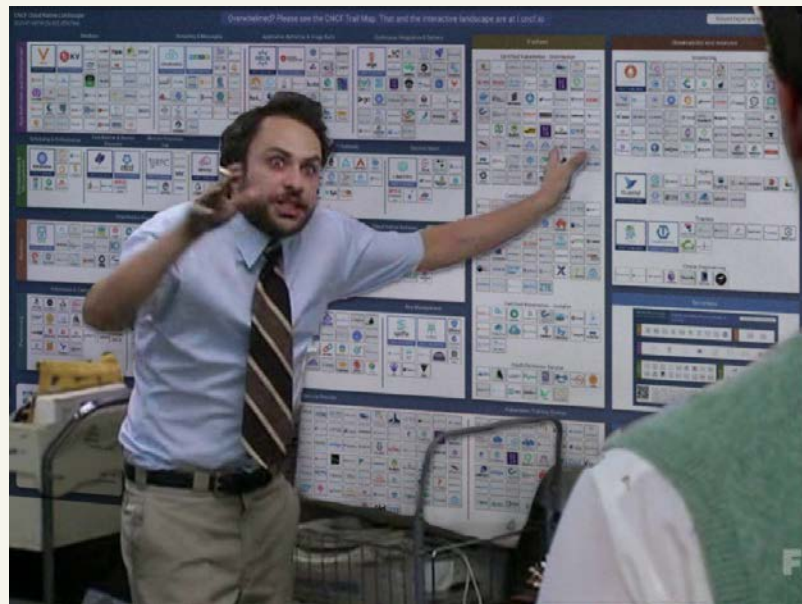


SHIFT AS MUCH LOAD AS POSSIBLE ONTO DEV

Challenges

About the Software Development Process

- Sprawling tooling landscape.
- Increasing cloud adoption.
- Popularization of complex tools.
- Developers needed to understand complex cloud native toolchains.
- Trends that made the developers' day-to-day more painful.



Challenges

Developers:

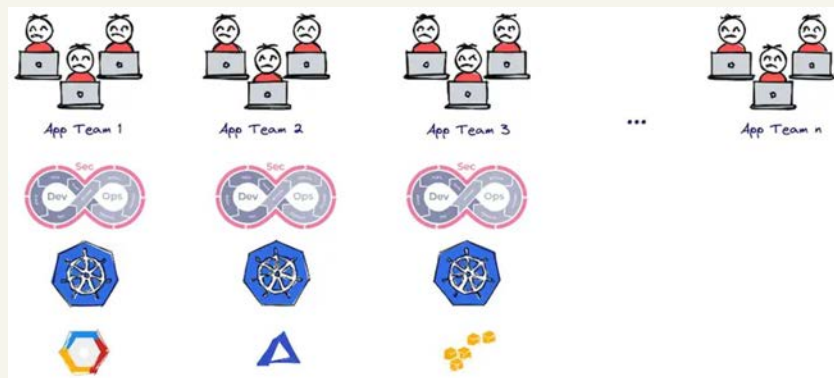
- Overload with unnecessary infrastructure/tools knowledge

Platform:

- Became a bottleneck
- Coordination is harder

Security:

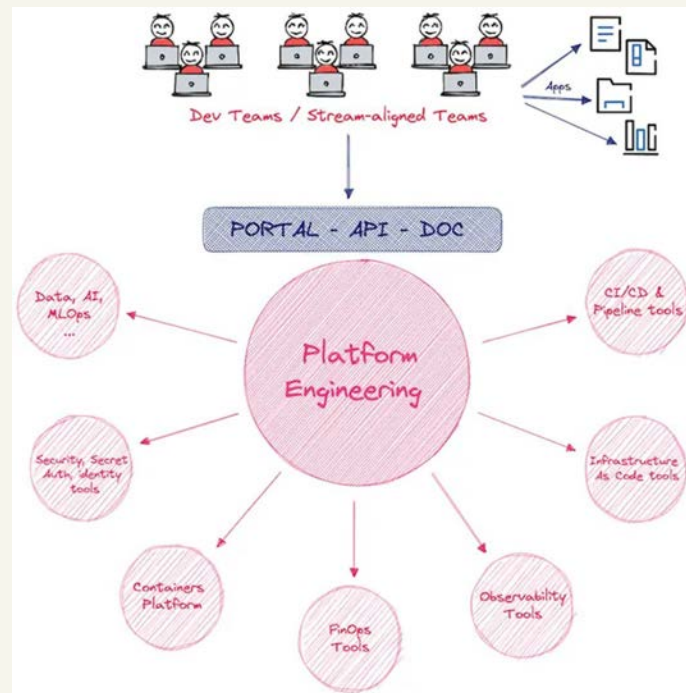
- Perimeter is harder to define and establish
- Blast radius is potentially huge



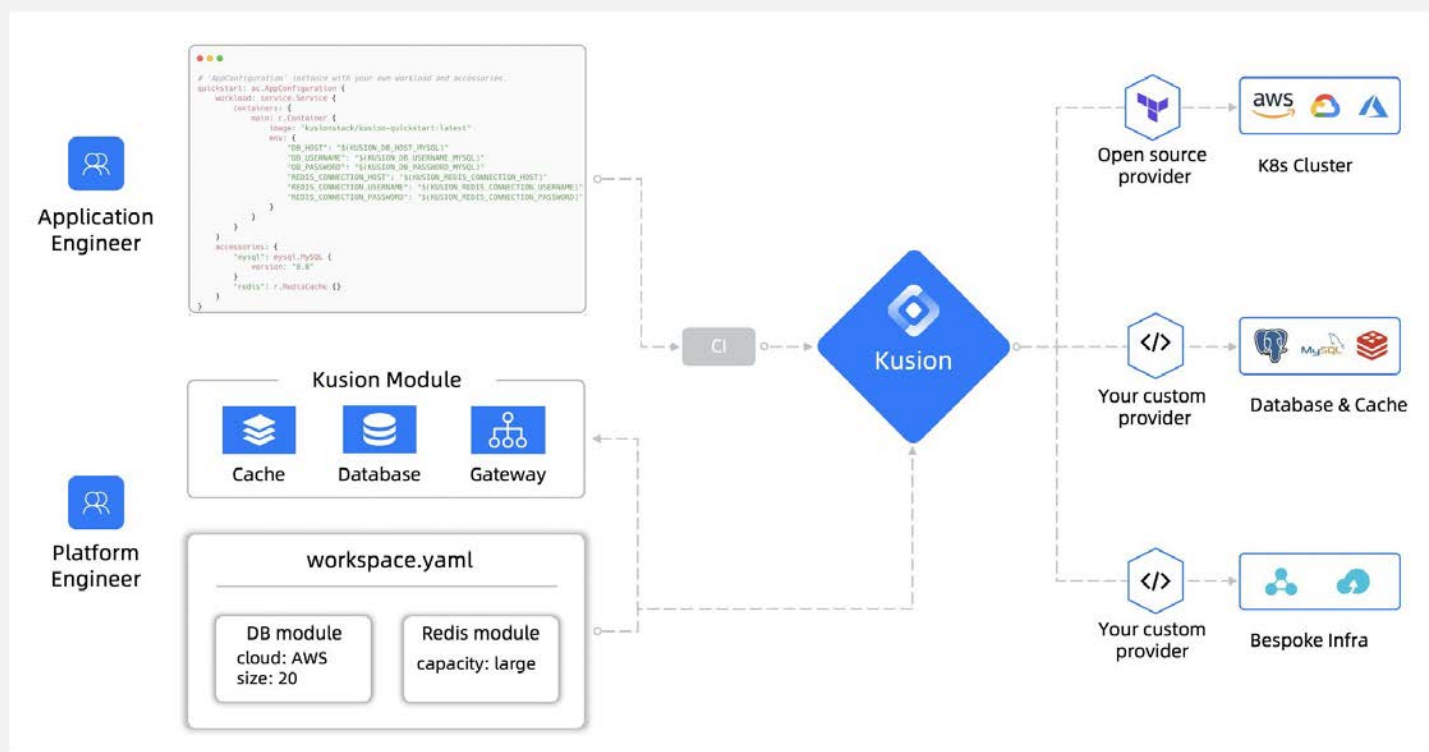
Platform Engineering

Platform Engineering Team.

- Building and maintaining IDPs for their "customers".
- Developer no need to master overly complicated and extensive toolchains → focus on what they do best - developing.
- Improves developer experience and drives increased developer productivity.



Platform Engineering with Kusion



Kusion AppConfiguration

```

1  helloWorld: ac.AppConfiguration {
2    workload: wl.Service {
3      containers: {
4        "main": c.Container {
5          image: "ghcr.io/kusion-stack/samples/helloworld:latest"
6          readinessProbe: p.Probe {
7            probeHandler: p.Http {
8              url: "http://localhost:80"
9            }
10         }
11       }
12     }
13   }
14 }
15
16 # a collection of accessories that will be attached to the workload
17 accessories: {
18   # Built-in module
19   "my-prometheus": m.Prometheus {
20     path: "/metrics"
21   }
22   # Built-in module
23   "my-database": d.MySQL {
24     type: "cloud"
25     version: "8.0"
26   }
27   # Customized module
28   "my-customize": customizedModule {
29     ...
30   }
31 }
32
33 pipeline: {
34   "step": Step {
35     use: "exec"
36     args: ["--test-all"]
37   }
38 }
39
40 dependency: {
41   dependentApps: ["init-kusion"]
42 }

```

Workload

Core part of application, representation of the code that runs in the cluster.

Accessory

Various runtime capabilities and operational requirements that app needs to function properly.

Dependency

Describes cross-application dependency.

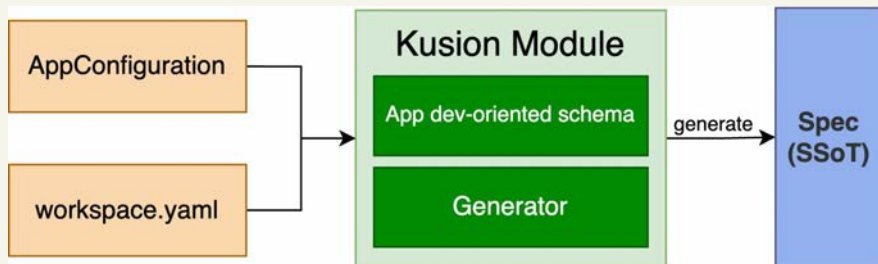
Pipeline

More fine-tuned control and customization over how to delivery app.

Developers Configuration

- AppConfiguration model consolidates workload and their dependent accessories for the application deployment.
- Drives a standardized and efficient application delivery and operation process in a hybrid environment.

Kusion Modules



- Standardized application deployments, reusable building blocks (database, storage, monitoring, etc)

Platform Configurations:

- Environment, vendor-specific should be defined by platform team.
- Application configuration should be decouple with platform so it can be written once and applied anywhere.
- Developers chooses environment as they want.

```
modules:
  mysql:
    default:
      provider: aws
      size: 20
      instanceClass: db.t3.micro
    smallClass:
      size: 50
      instanceClass: db.t3.small
    largeClass:
      instanceClass: db.t3.large
```

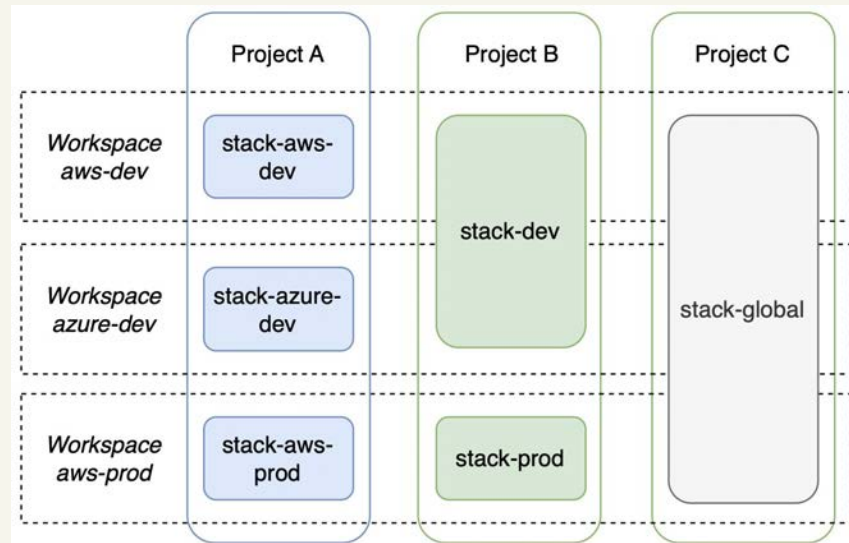
Kusion Workspace

Platform Configurations:

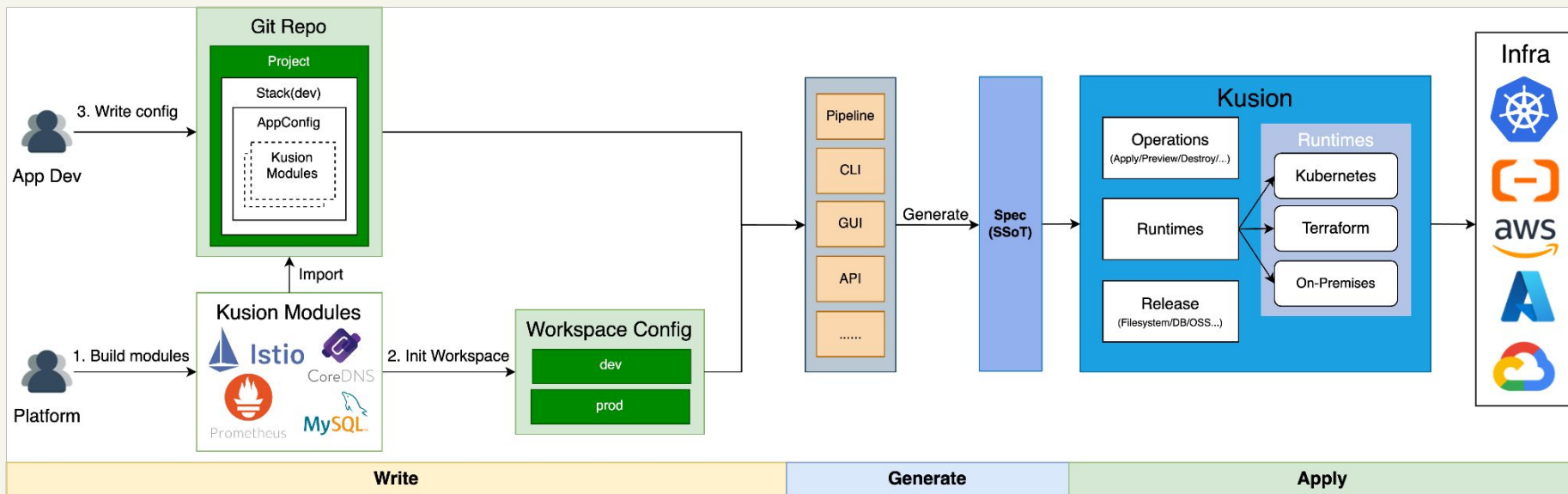
- Logical concept that maps to an actual target environment to deploy
- Associated with a series of platform configurations that are used by developers

```

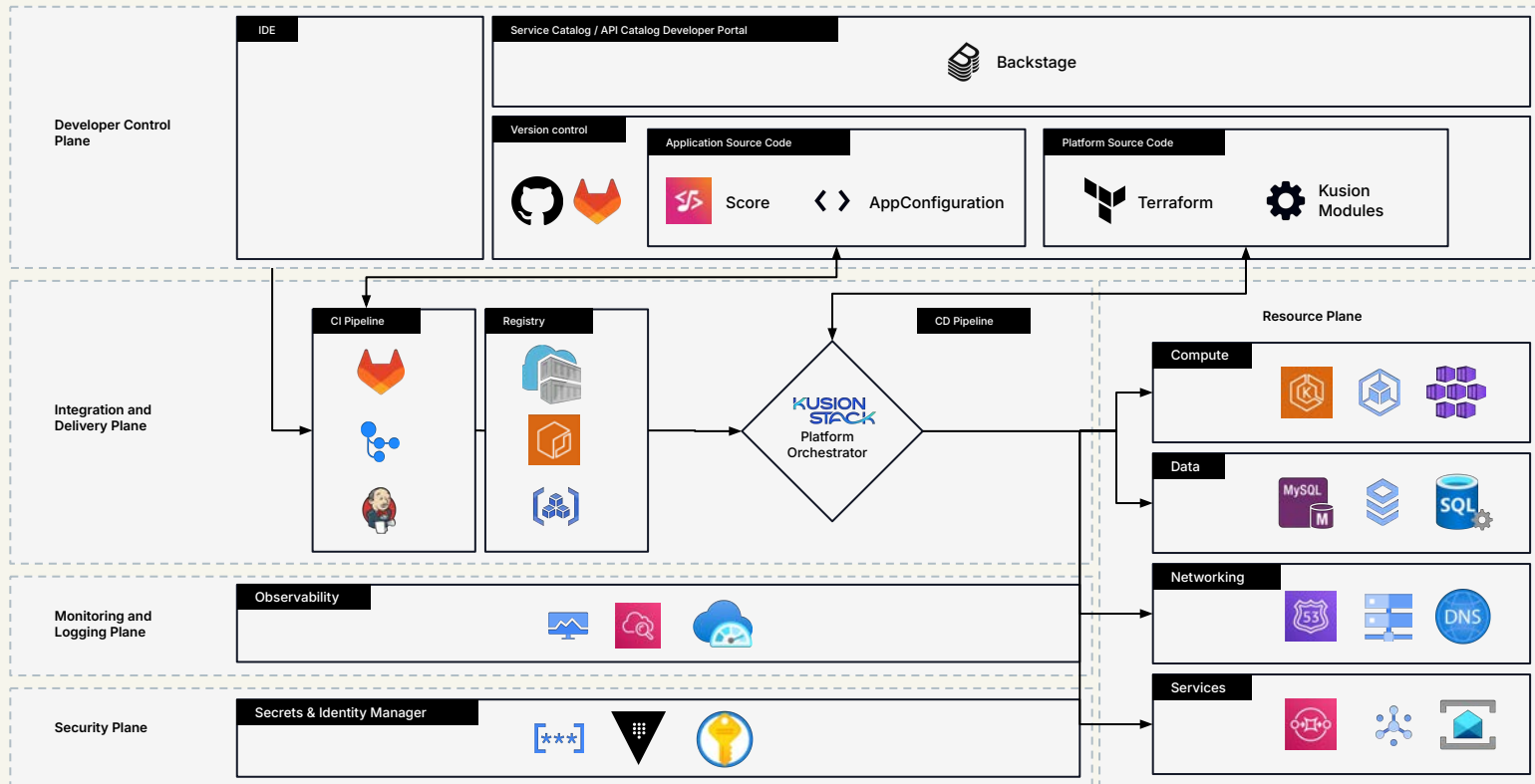
modules:
  mysql:
    default:
      provider: aws
      size: 20
      instanceClass: db.t3.micro
    smallClass:
      size: 50
      instanceClass: db.t3.small
    largeClass:
      instanceClass: db.t3.large
  secretStore:
    provider:
      aws:
        region: us-east-1
  context:
    KUBECONFIG_CONTENT: ref://stag-cluster/kubeconfig
  
```



Overview of workflow



Kusion and IDP



Kusion Server and IDP

- Kusion today serves as a Command Line Interface (CLI).
- Kusion proposes a protocol-driven collaboration paradigm which includes the platform owners and application developers to each do their jobs separately.
- Proposing Kusion 2.0 which is a form of Kusion that runs as a long-running service and exposes REST service endpoints.
- Easy to integrate with Internal Developer Portal

Thank you

Hoang Dinh Nguyen / Cloud Engineer

email: hoangndst@gmail.com

linkedin: <https://www.linkedin.com/in/hoangndst>

telegram: <https://t.me/hoangndst>

website: <https://hoangndst.com>

kusion: <https://www.kusionstack.io>