

Continuous GitOps

Arun Ramakani
@imarunrk



ustomize.io



flux



Open Policy Agent
Gatekeeper



Width & Depth of Kubernetes API

Application Dev

Product Description

- Pod
- Volume
- Service
- Deployment
- Ingress

Application Ops

Product Config

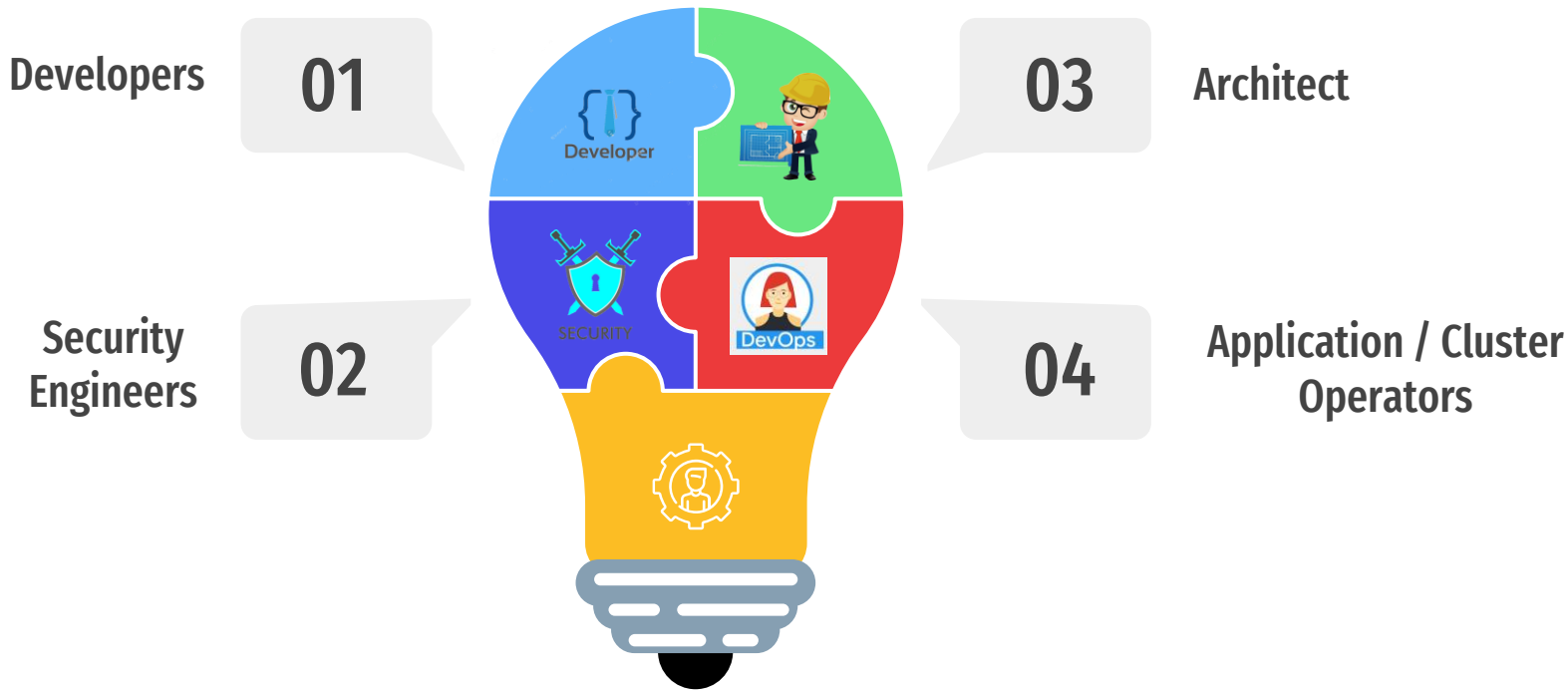
- Config Map
- Secrets
- Labels
- Namespaces

Cluster Ops

Cluster Config

- Network Rules
- Logging
- Monitoring
- Sealed Secrets
- Policies

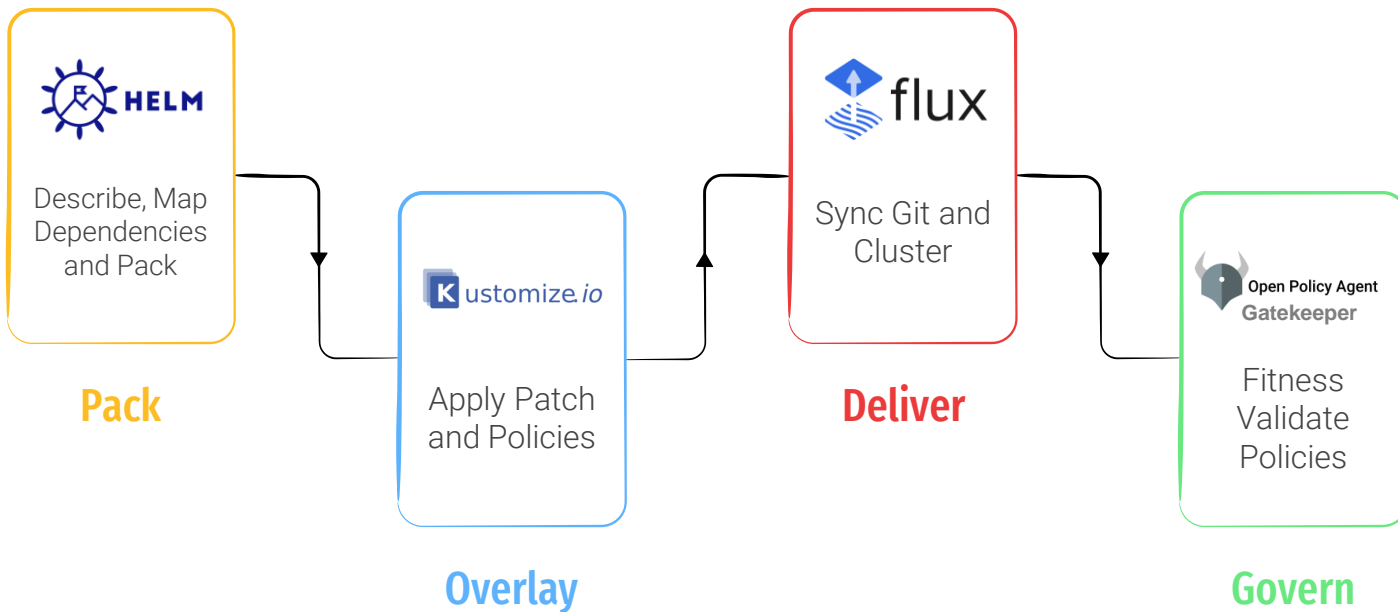
Solving the YAML Puzzle



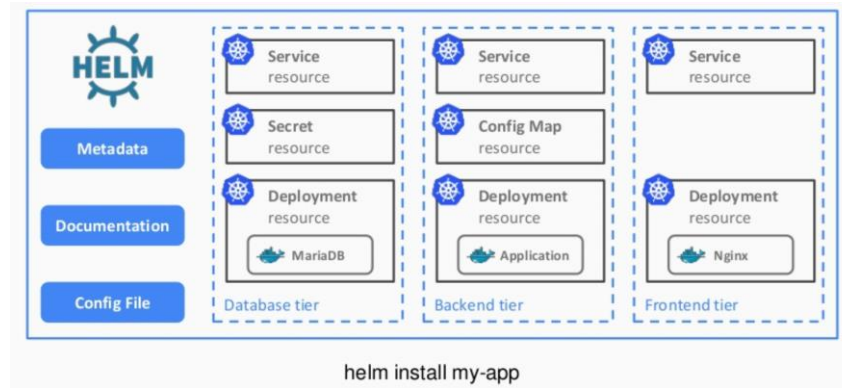
Guiding Principles to Choose Tools

- Support Different Team Construct & Easy to Collaborate
- Support Version Control & Cross Pollination
- Expose Full Power Kubernetes API & Support Customization
- Avoid Leaky Abstraction & Support Segregation of Concerns
- Application & Cluster Security
- Cost Of Workflow

The Workflow



Stage 1 : Packing With Helm



```
Chart.yaml      # A YAML file containing information about the chart
LICENSE         # OPTIONAL: A plain text file containing the license for the chart
README.md       # OPTIONAL: A human-readable README file
requirements.yaml # OPTIONAL: A YAML file listing dependencies for the chart
values.yaml     # The default configuration values for this chart
charts/         # A directory containing any charts upon which this chart depends.
templates/      # A directory of templates that, when combined with values,
                  # will generate valid Kubernetes manifest files.
templates/NOTES.txt # OPTIONAL: A plain text file containing short usage notes
```

Stage 2 : Overlay with Kustomize



***Enable Your Governance Teams To
Customize Your YAML***

```
1 apiVersion: v1beta2
2 kind: Deployment
3 metadata:
4   name: ldap
5   labels:
6     app: ldap
7 spec:
8   replicas: 1
9   selector:
10     matchLabels:
11       app: ldap
12   template:
13     metadata:
14       labels:
```

patch.yaml - Staging

```
1 # Staging Deployment
2
3 apiVersion: apps/v1beta2
4 kind: Deployment
5 metadata:
6   name: ldap
7 spec:
8   replicas: 2
```

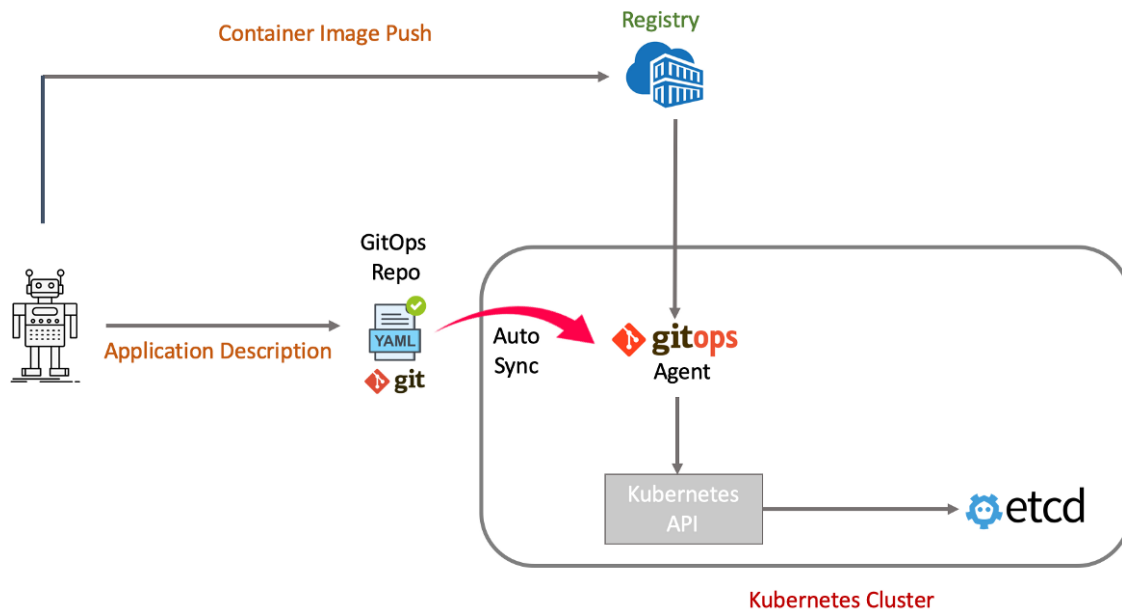
```
1 bases:
2   - ldap
3 patches
4   - patch.yaml
5 ...
```

```
16
17
11 - name: ldap-data
12   mountPath: /var/l
13 - name: ldap-config
14   mountPath: /etc/l
15 - name: ldap-certs
16   mountPath: /conta
17 - name: configmap-v
18   mountPath: /conta
19 - name: container-r
20   mountPath: /conta
21
22 ports:
23   - containerPort: 389
24     name: openldap
25
26 volumes:
27   - name: ldap-data
```

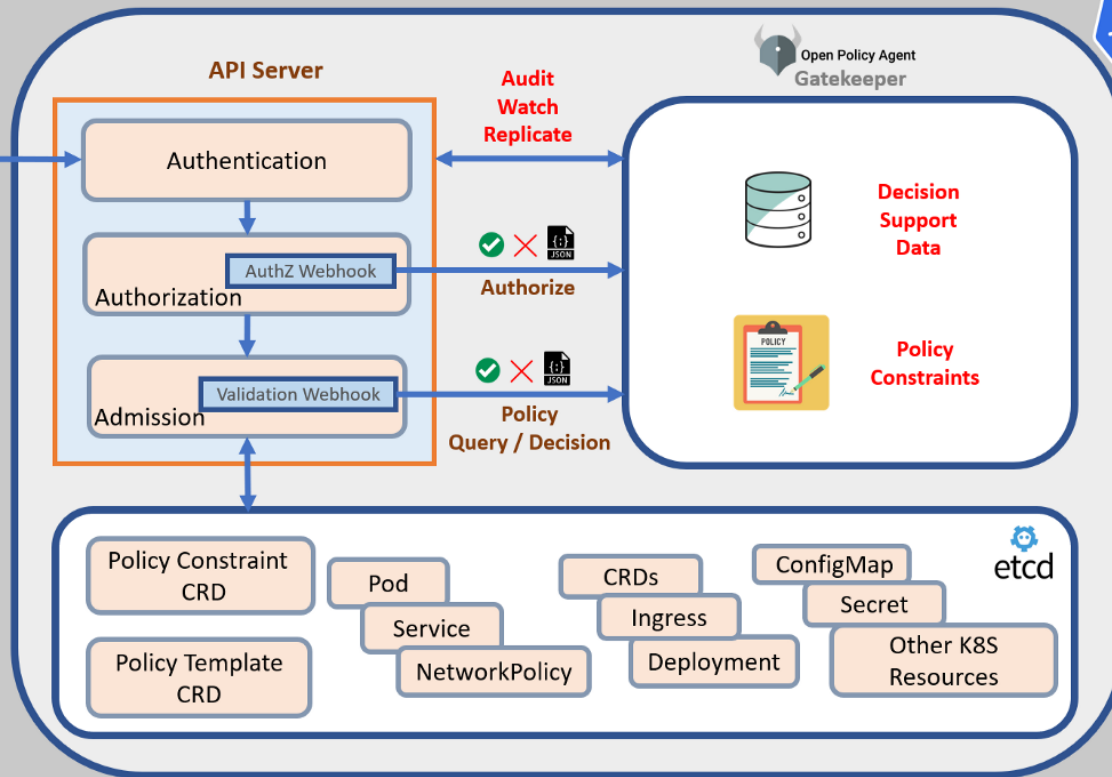
patch.yaml - Prod

```
1 # Production Deployment
2
3 apiVersion: apps/v1beta2
4 kind: Deployment
5 metadata:
6   name: ldap
7 spec:
8   replicas: 6
9   template:
10     spec:
11       volumes:
12         - name: ldap-data
13           emptyDir: null
```

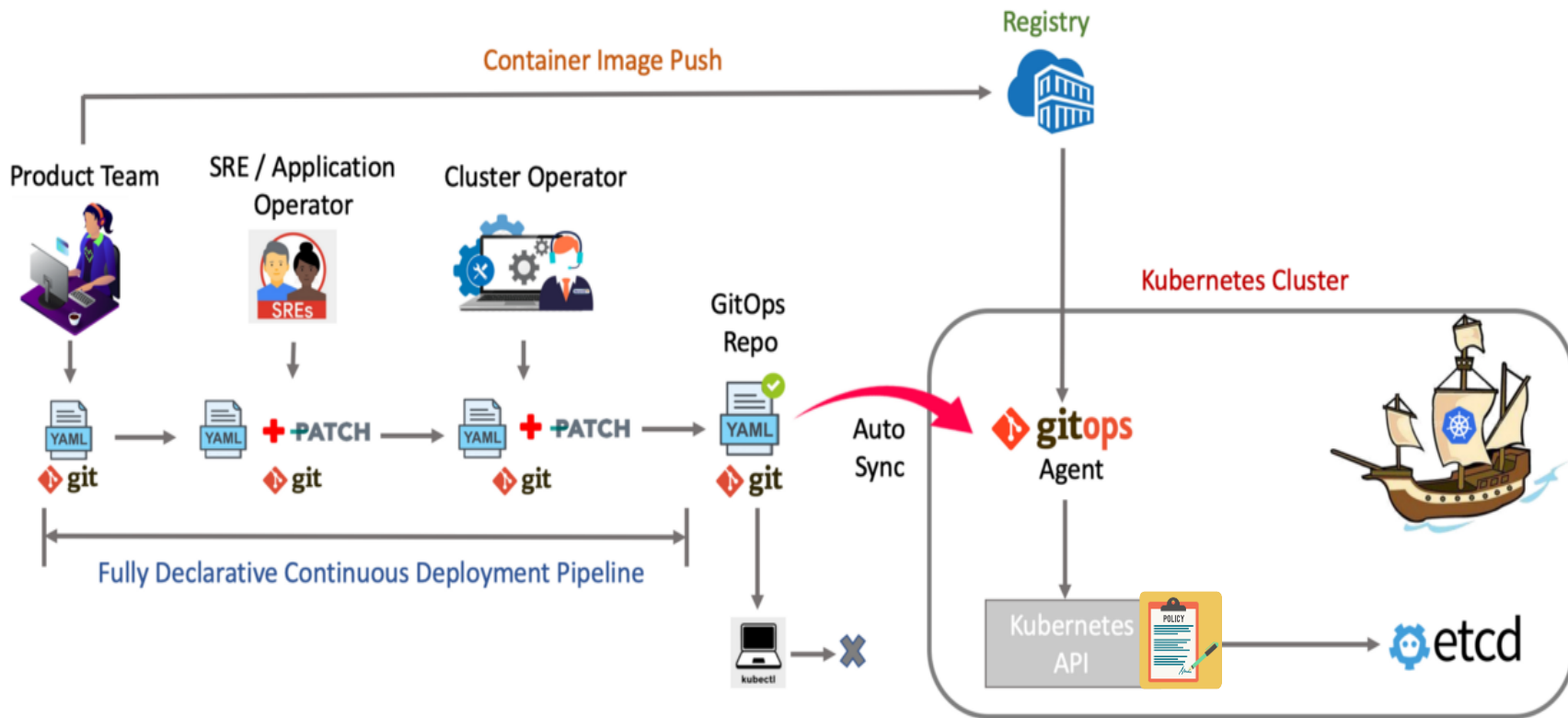
Stage 3 : Deliver with Flux



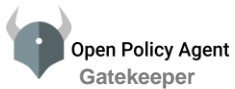
Stage 4 : Govern with Gatekeeper



Tool Agnostic CD Workflow



Demo



Questions

For more question follow me in Medium & Twitter
[@imarunrk](#)

