KubeCon

North America 2017

Webhooks for automated updates

Rajashree Mandaogane, Software Engineer, Rancher Labs

whoami

Rajashree M

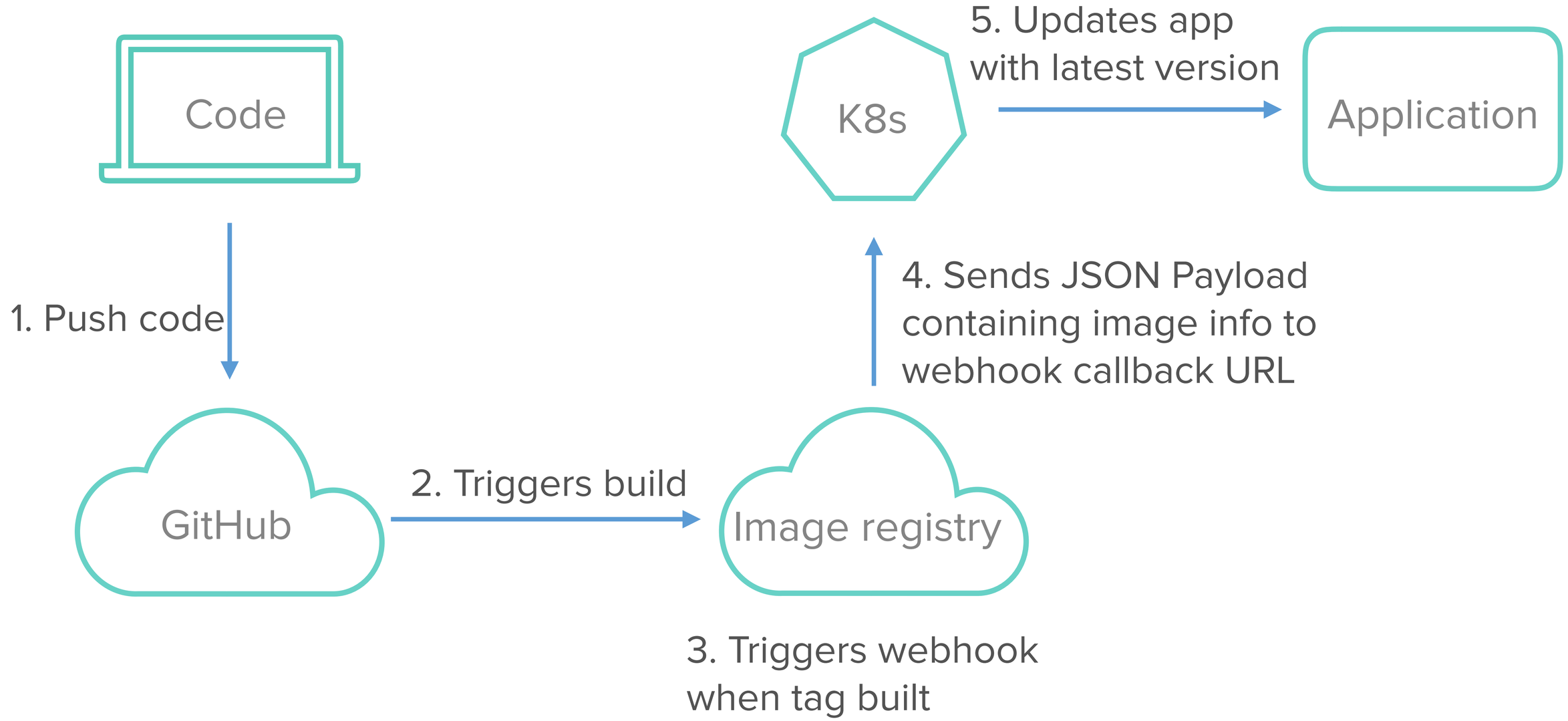
 @rajashree_28

 mrajashree

RANCHER

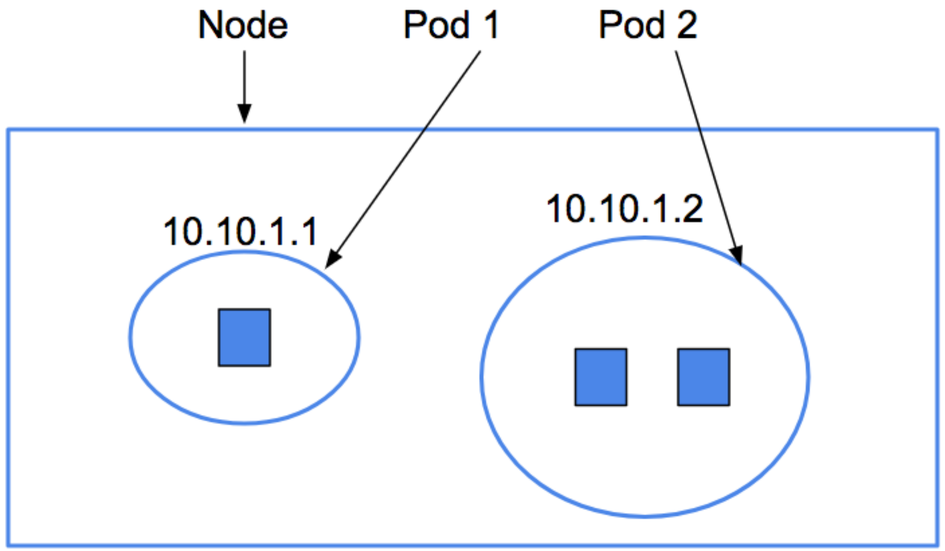
@Rancher_Labs

github.com/rancher


Code
1. Push code
GitHub
2. Triggers build
Image registry
3. Triggers webhook
when tag built
4. Sends JSON Payload
containing image info to
webhook callback URL
K8s
5. Updates app
with latest version
Application

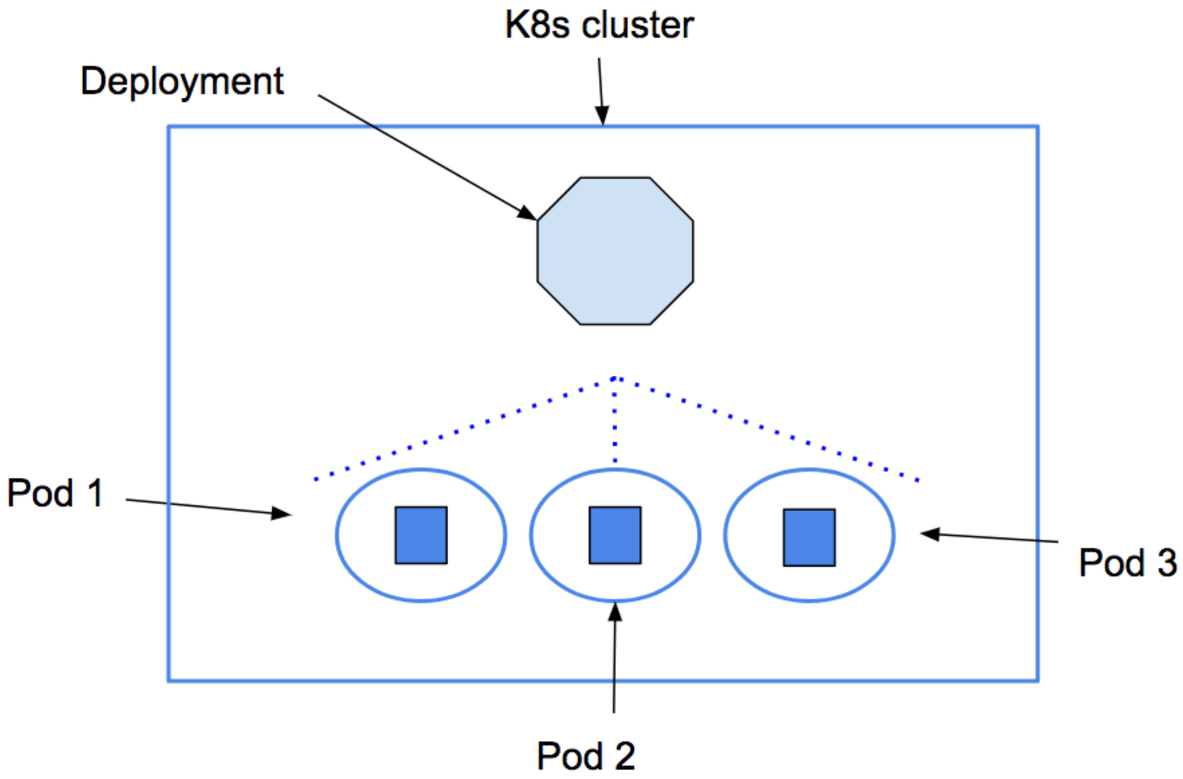
Pod

- Smallest deployable unit
- Runs one or more containers
- Containers tightly coupled
- Unique IP address



Deployment

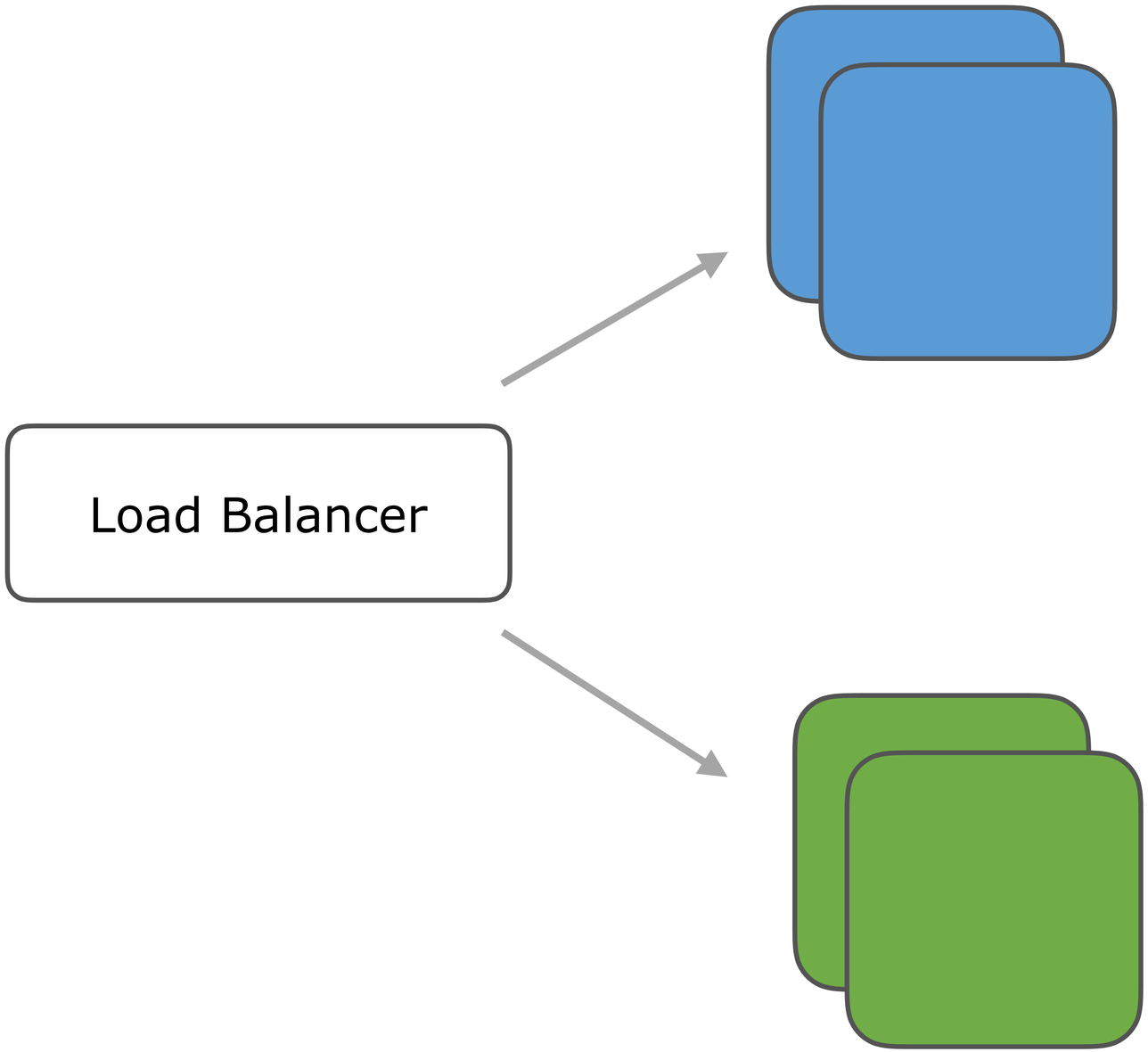
- Pods are ephemeral
- ‘Deployment’ manages life-cycle of pods
- Desired state vs current state



DevOps upgrade/deployment strategies

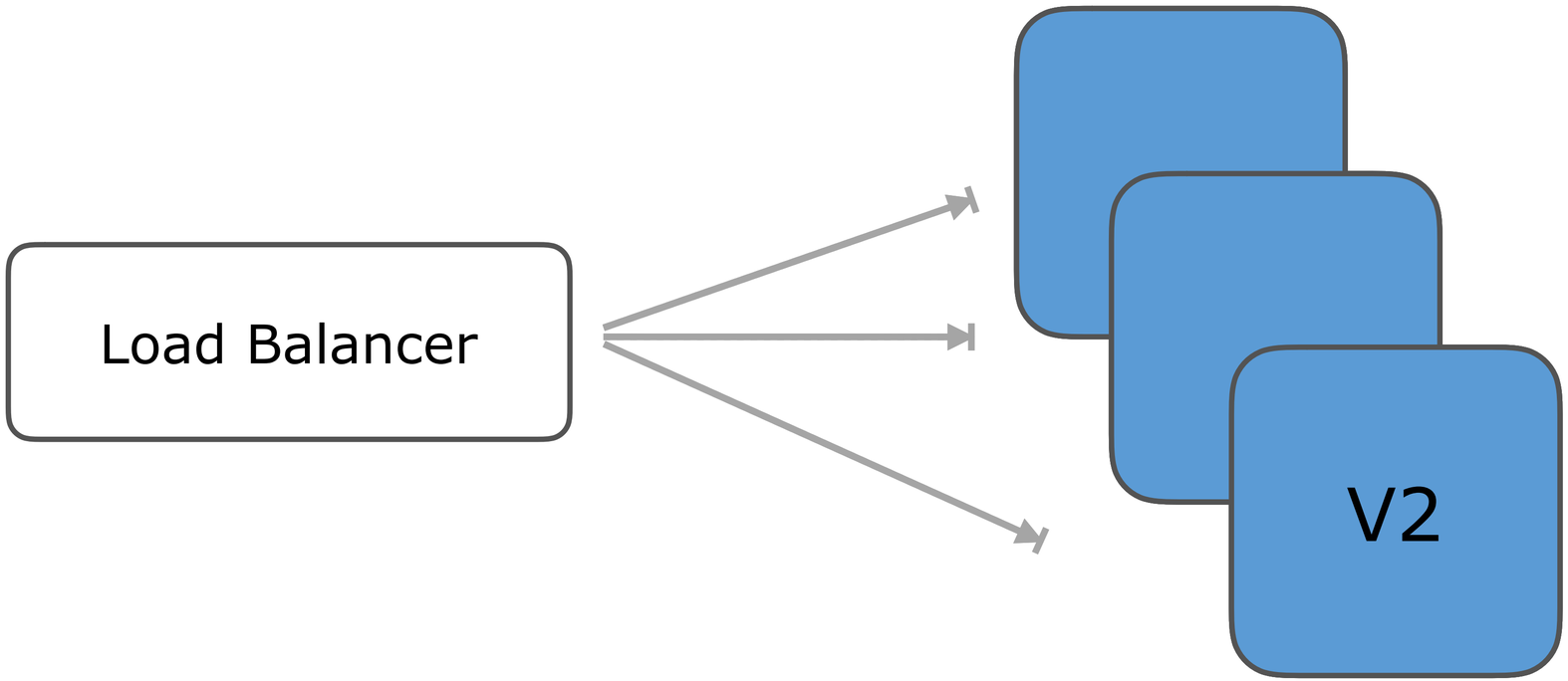
Blue Green Strategy

- Identical production environment for new version
- Zero downtime, option to rollback


Load Balancer

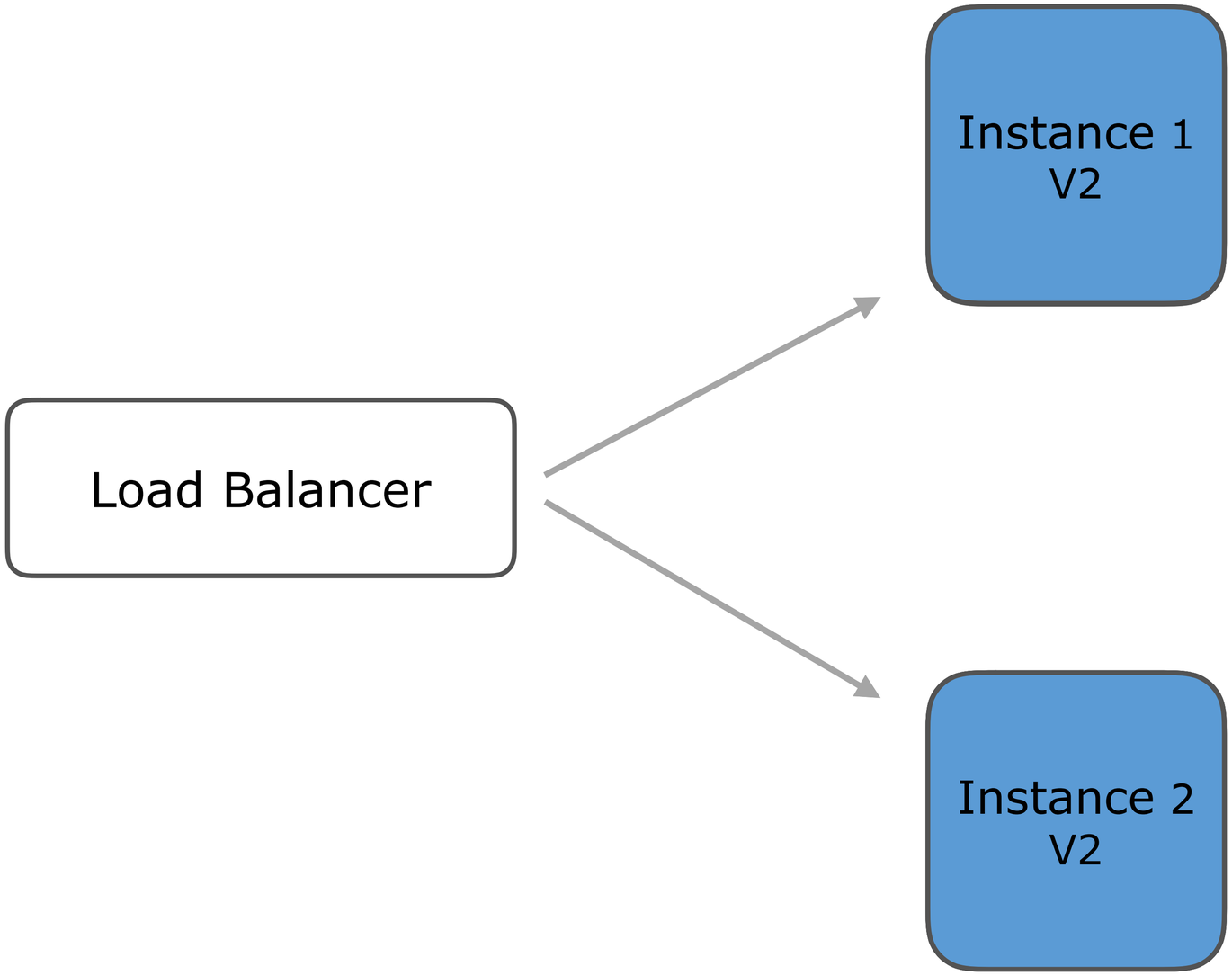
Recreate Strategy

- Remove older instances, then create new ones
- Incurs downtime


Load Balancer
V2

Rolling Update strategy

- Update ‘n’ out of ‘m’ instances at a time
- Always some instances with older code running
- Zero downtime
- Less resource utilization as no need for identical env
- App must support running old and new versions at same time


Instance 1
V2
Load Balancer
Instance 2
V2

How to use these in k8s?

- Specify app update strategy through k8s ‘Deployment’ resource
- Provides recreate and rollingUpdate options
- rollingUpdate zero downtime, recreate incurs downtime

Kubernetes rollingUpdate strategy

- At most pods allowed to be unavailable: 1
- At most pods allowed to be scheduled: 4 (3+1)
- So only one pod updated at a time
- Can also be percentage values

```yaml
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: test-deployment
spec:
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 1
      maxSurge: 1
  replicas: 3
```

How to trigger rollingUpdate?

Start RollingUpdate manually

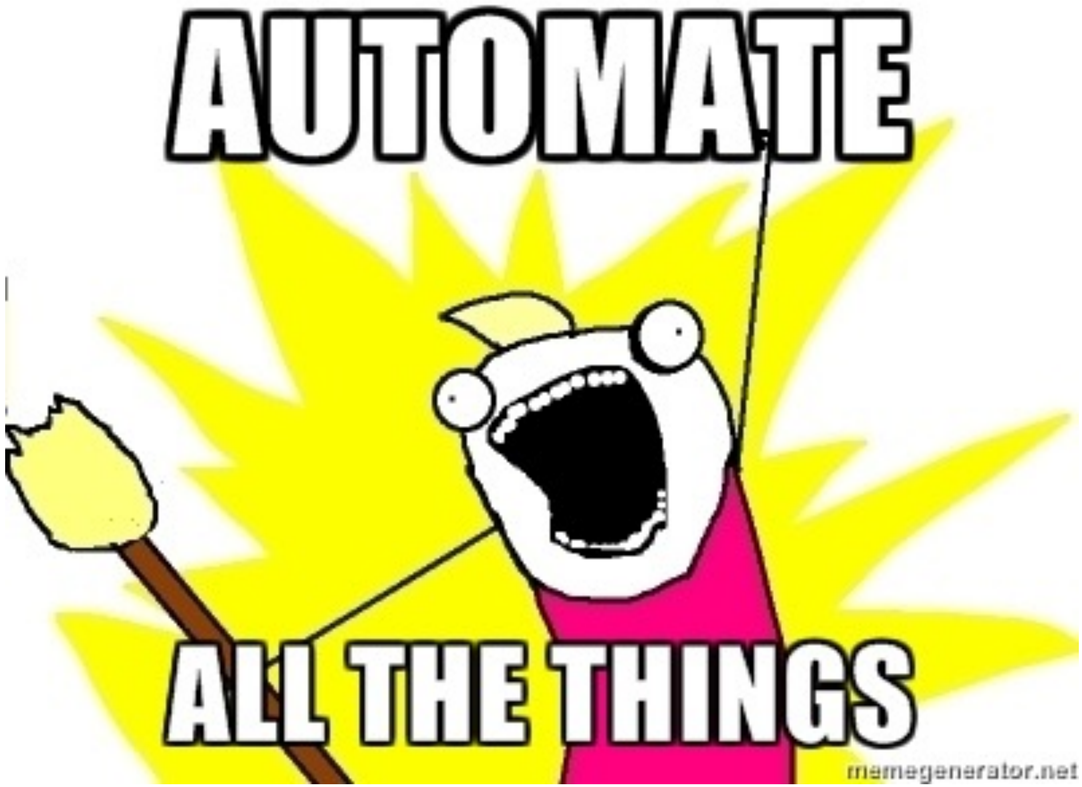
- set image

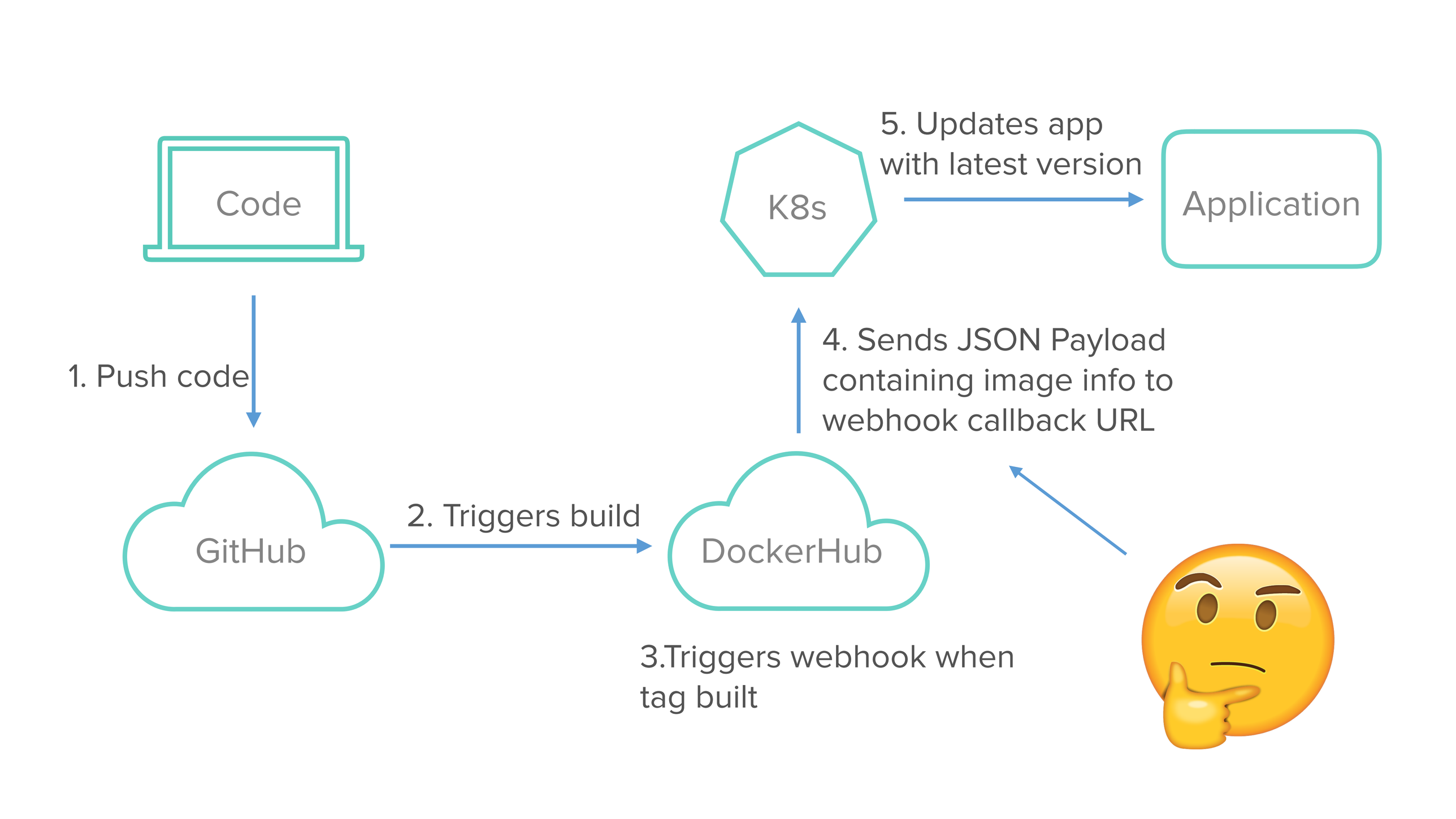
```
$ kubectl set image deployment/nginx-deployment nginx=nginx:1.9.1
deployment "nginx-deployment" image updated
```

- edit

```
$ kubectl edit deployment/nginx-deployment
deployment "nginx-deployment" edited
```

Any other way?




Code
1. Push code
GitHub
2. Triggers build
DockerHub
3.Triggers webhook when tag built
4. Sends JSON Payload containing image info to webhook callback URL
K8s
5. Updates app with latest version
Application

Webhook receiver

- Webhook receiver will consume Docker Hub webhook
- Find the image tag pushed
- Patch k8s deployment resource via API call

Where does this webhook receiver go?

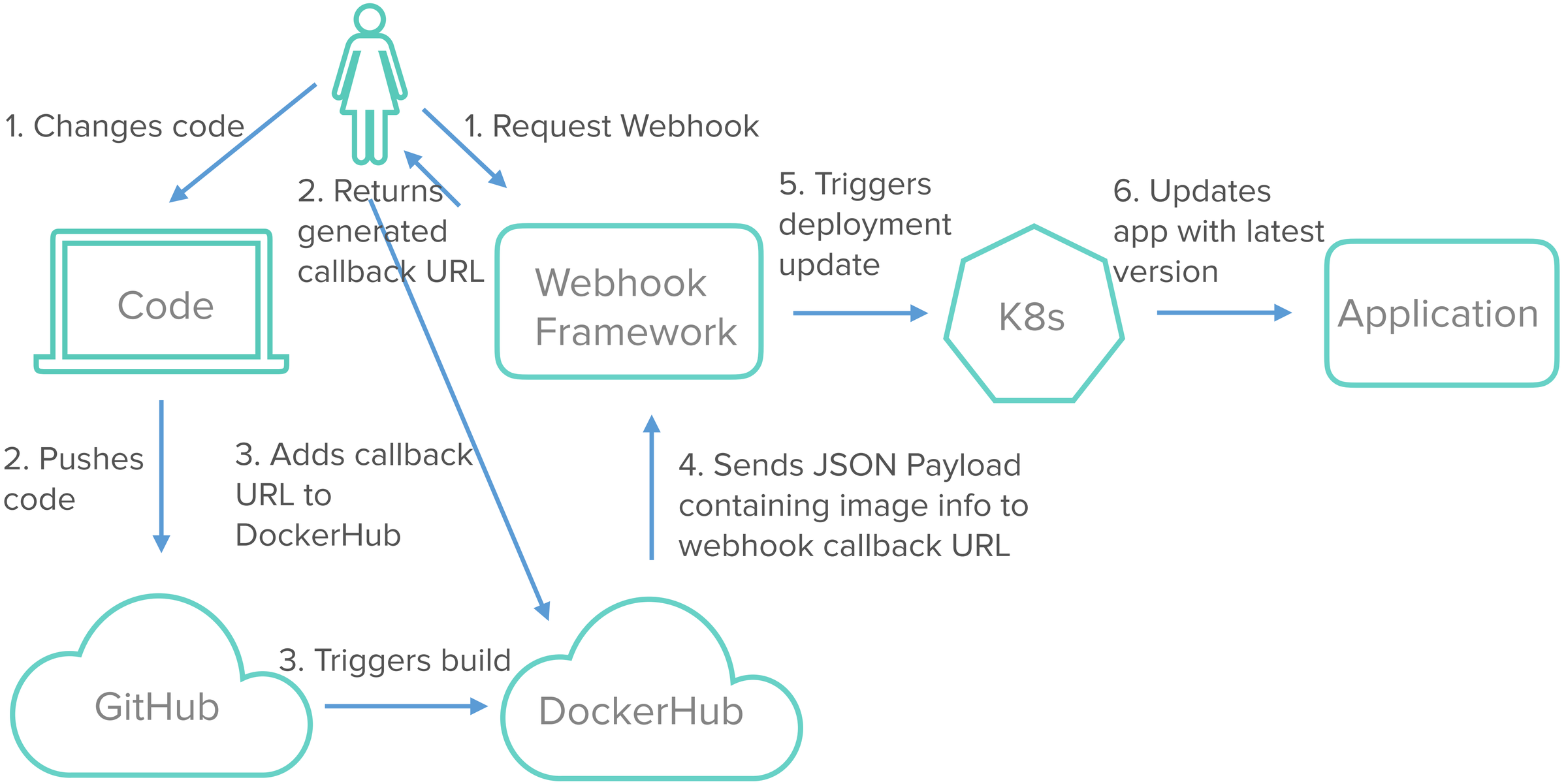
- One option: Rancher's webhook-receiver framework
- Go microservice that triggers actions within Rancher
- On trigger makes API calls
- https://github.com/rancher/webhook-service

Automate with Rancher’s webhook framework

Rancher webhook service framework

```go
//WebhookDriver interface for all drivers
type WebhookDriver interface {
    ValidatePayload(config interface{}, apiClient *client.RancherClient) (int, error)
    Execute(config interface{}, apiClient *client.RancherClient, requestBody interface{}) (int, error)
    GetDriverConfigResource() interface{}
    ConvertToConfigAndSetOnWebhook(conf interface{}, webhook *model.Webhook) error
    CustomizeSchema(schema *v1client.Schema) *v1client.Schema
}

//RegisterDrivers creates object of type driver for every request
func RegisterDrivers() {
    Drivers = map[string]WebhookDriver{}
    Drivers["scaleService"] = &ScaleServiceDriver{}
    Drivers["serviceUpgrade"] = &ServiceUpgradeDriver{}
    Drivers["scaleHost"] = &ScaleHostDriver{}
}
```


1. Changes code
1. Request Webhook
2. Returns generated callback URL
Code
Webhook Framework
5. Triggers deployment update
K8s
6. Updates app with latest version
Application
2. Pushes code
3. Adds callback URL to DockerHub
4. Sends JSON Payload containing image info to webhook callback URL
GitHub
3. Triggers build
DockerHub

DEMO

Thank you!

- Rajashree Mandaogane
  @rajashree_28 
  mrajashree