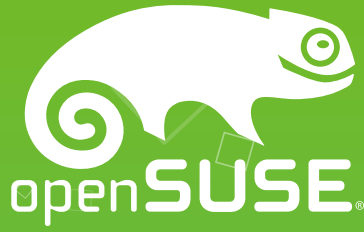


Reliable Deployments with Kubernetes and Istio

Implement blue/green deployments and canary roll outs

Estu Fardani

estu@opensuse.id

tuanpembual.wordpress.com

@tuanpembual

About Me

- Estu Fardani
- @tuanpembual
- DevOps Consultant
- Gecko Lover

About this workshop

- git clone github.com/tuanpembual/bali

What is Istio

- Istio is an open source independent service mesh that provides the fundamentals you need to successfully run a distributed microservice architecture.
- Istio lets you connect, secure, control, and observe services.

What is a service mesh?

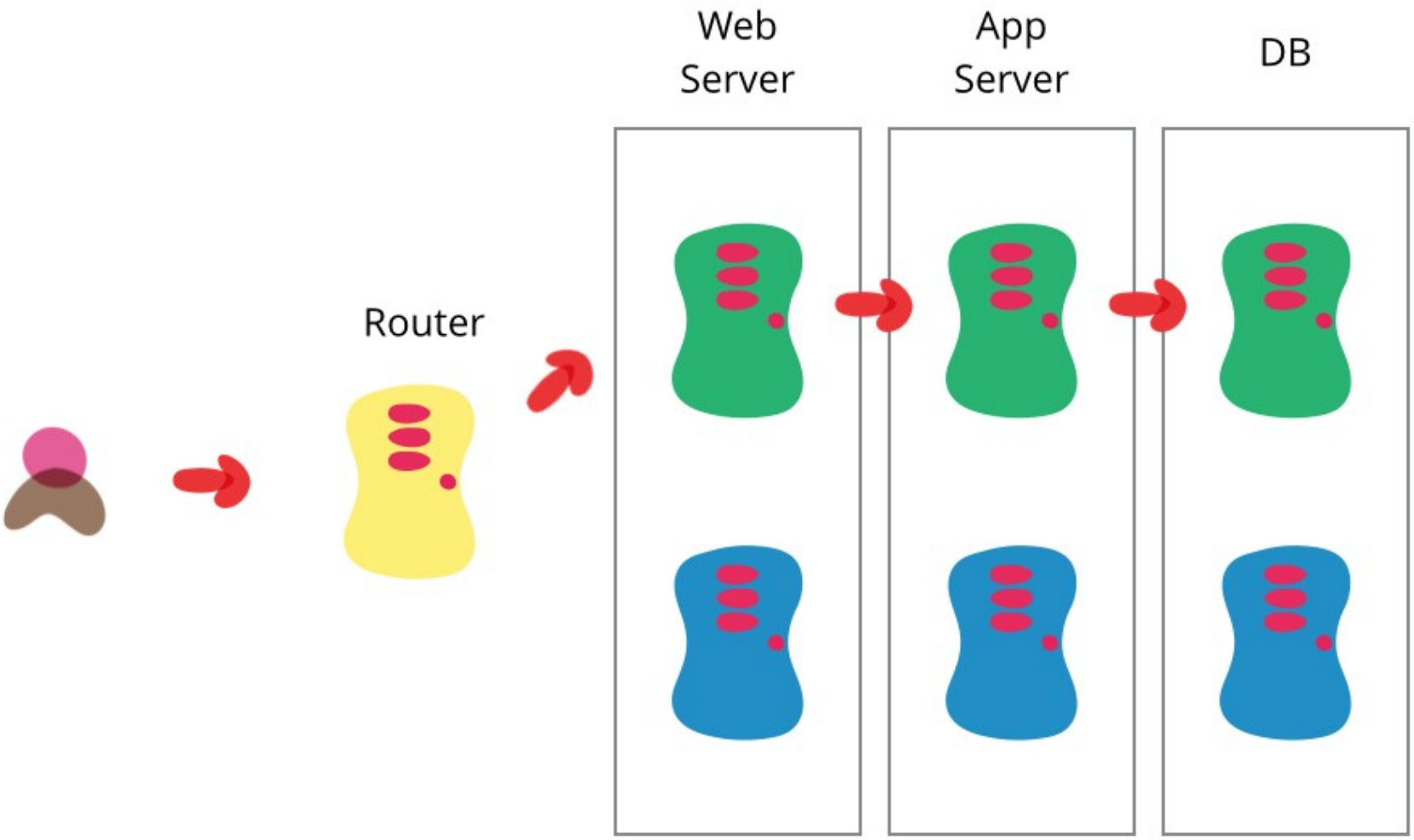
- The term service mesh is used to describe the network of microservices that make up such applications and the interactions between them

Why use Istio?

- Istio makes it easy to create a network of deployed services with load balancing, service-to-service authentication, monitoring, and more, with few or no code changes in service code

Blue-Green Deployment

- The blue-green deployment approach does this by ensuring you have two production environments, as identical as possible. At any time one of them, let's say blue for the example, is live.

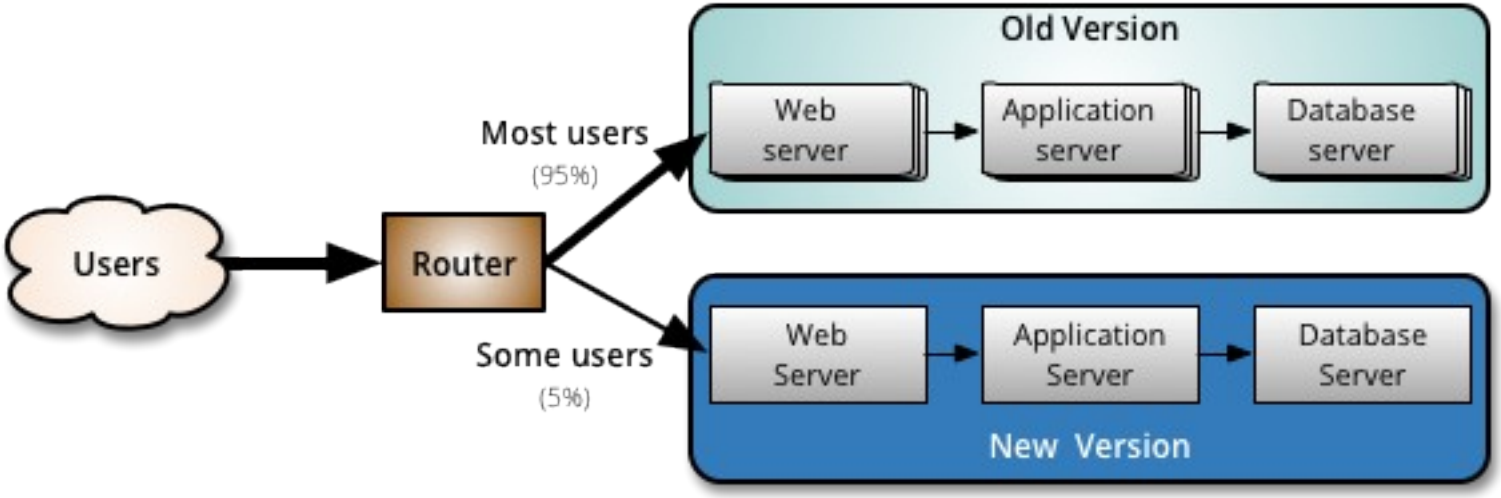


https://martinfowler.com/bliki/BlueGreenDeployment.html

Canary Release

- Canary release is a technique to reduce the risk of introducing a new software version in production by slowly rolling out the change to a small subset of users before rolling it out to the entire infrastructure and making it available to everybody.

Canary Release



https://martinfowler.com/bliki/CanaryRelease.html

Starting Point

Familiar with:

- Docker
- k8s/minikube

Go to Labs

- We will create kubernetes cluser using minikube

Go to Labs

- We will create kubernetes cluser using minikube
- Create vm on local or cloud(vagrant, digital ocean, aws)

Go to Labs

Install tools:

- Docker without sudo
- Kubectl
- Minikube
- Security group
- Gist.

On VM

$ docker version

$ kubectl version

$ minikube version

Go to Labs

Warming up:

$ minikube start –vm-driver=none

$ kubectl run hello-minikube
--image=gcr.io/google_containers/echoserver:1.4 --port=8080

$ kubectl expose deployment hello-minikube --type=NodePort

$ kubectl get services

$ kubectl delete services hello-minikube

$ kubectl delete deployment hello-minikube

Go to Labs (2)

```
$ curl -L https://git.io/getLatestIstio |
ISTIO_VERSION=1.3.0 sh -

$ cd istio-1.3.0

$ for i in install/kubernetes/helm/istio-init/files/
crd*yaml; do kubectl apply -f $i; done
```

Go to Labs (3)

$ vim install/kubernetes/istio-demo.yaml

line 17647

```
apiVersion: v1
kind: Service
metadata:
 name: istio-ingressgateway
 namespace: istio-system
 annotations:
 labels:
   chart: gateways
   heritage: Tiller
   release: istio
   app: istio-ingressgateway
   istio: ingressgateway
spec:
```

save n exit

Go to Labs (3)

$ kubectl apply -f install/kubernetes/istio-demo.yaml

Testing deploy 2 version apps

$ vim my-app.yaml

$ kubectl apply -f my-app.yaml

$ kubectl get pods

$ kubectl get svc

Listen Apps

Open two console:

$ kubectl port-forward --address 0.0.0.0 deployment/myapp-v1 8080:80

$ kubectl port-forward --address 0.0.0.0 deployment/myapp-v2 8081:80

Create Gateway

```
$ vim app-gateway.yaml

$ kubectl apply -f app-gateway.yaml

# edit, save and update

$ kubectl replace -f app-gateway.yaml
```

Testing Load

Custom load value

```
    - destination:
        host: myapp
        subset: v1
      weight: 100
    - destination:
        host: myapp
        subset: v2
      weight: 0
```

Debug Config

```
$ ip addr # get local ipaddress

$ while : ;do export GREP_COLOR='1;33';curl -s
172.31.25.31:31380 \
 |  grep --color=always "V1" ; export GREP_COLOR='1;36';\
 curl -s  172.31.25.31:31380 \
 | grep --color=always "vNext" ; sleep 1; done
```

See the magic

FROM:

* https://www.radishlogic.com/kubernetes/running-minikube-in-aws-ec2-ubuntu/

* https://thenewstack.io/tutorial-blue-green-deployments-with-kubernetes-and-istio/

* https://cheatsheet.dennyzhang.com/cheatsheet-minikube-a4

Conclusion

- It is still prove of concept at the simple case.
- Production, of course not simple as my demo.
- Ping me after this workshop :D

Q&A

- ?

Closing

Thank you for join my workshop

See you next year!!