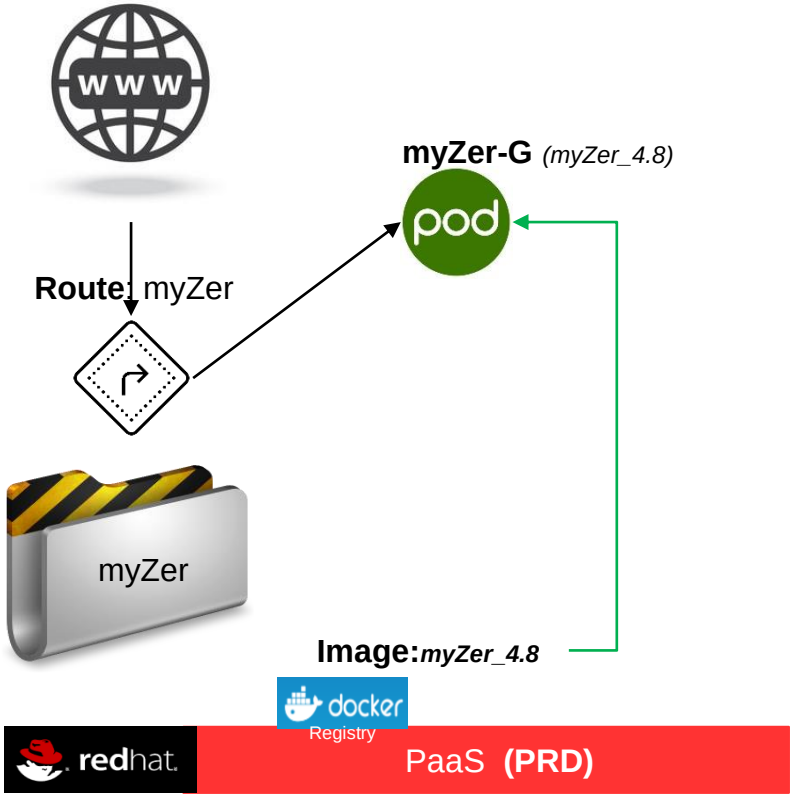
Blue-Green Deployment with OCP

Hoo Chuan Wu , 符传武
Senior DevOps Specialist Architect

Blue-Green Deployment

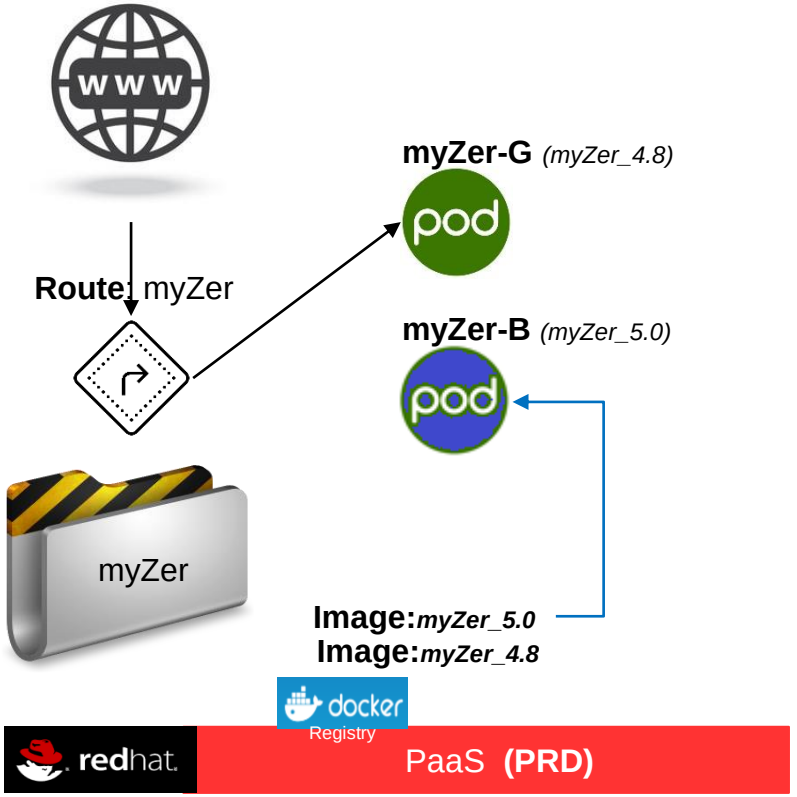
Eg. Current Production environment is deployed with release-tag : myZer_4.8



Blue-Green Deployment - Process

New deployment (**newService**)

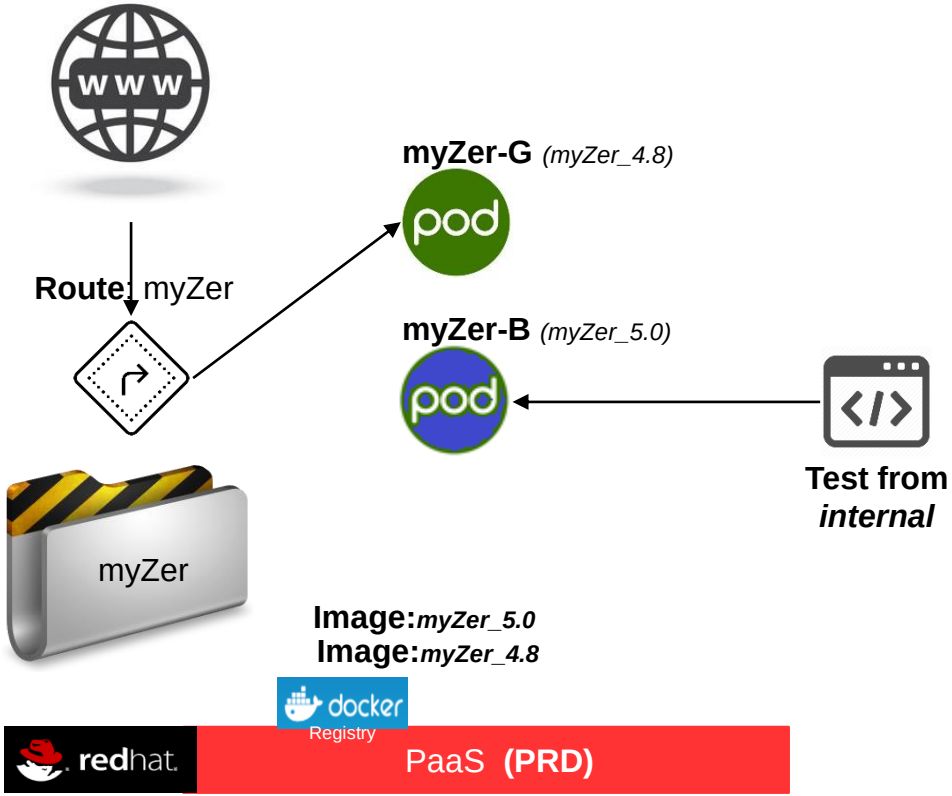
1. **Create blue pod with myZer_5.0**



Blue-Green Deployment - Process

New deployment

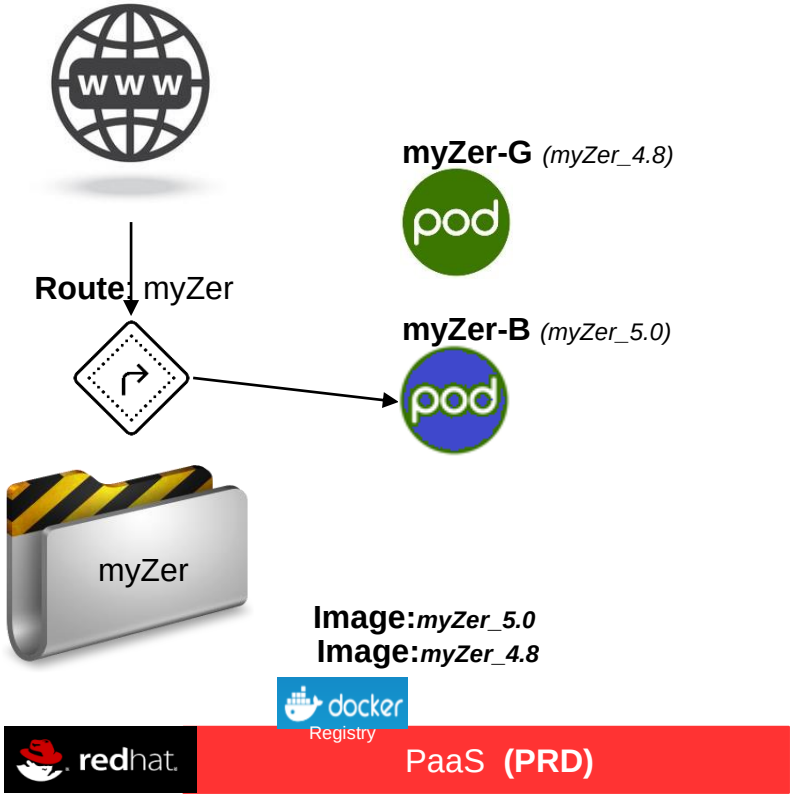
1. Create blue pod with myZer_5.0
2. **Test new service is ok**



Blue-Green Deployment - Process

New deployment (**switch**)

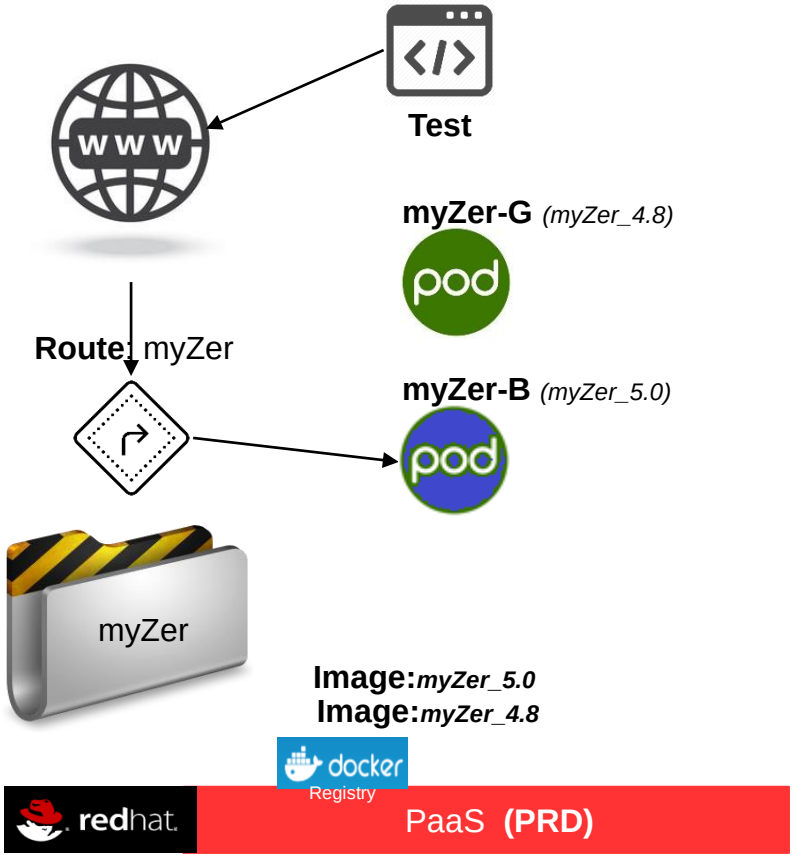
1. Create blue pod with myZer_5.0
2. Test new service is ok
3. **Switch from Green to Blue pod**



Blue-Green Deployment - Process

New deployment

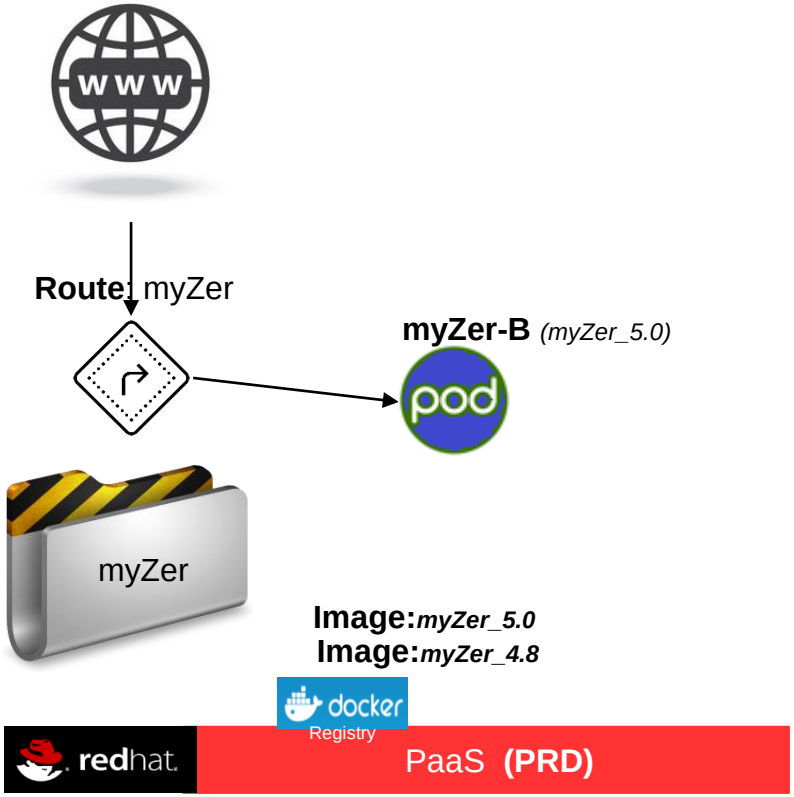
1. Create blue pod with myZer_5.0
2. Test new service is ok
3. Switch from Green to Blue pod
4. **Validate Production is ok**



Blue-Green Deployment - Process

New deployment (**commit**)

1. Create blue pod with myZer_5.0
2. Test new service is ok
3. Switch from Green to Blue pod
4. Validate Production is ok
5. **Delete green pod**



Blue-Green Deployment

Implementation – Pseudo Code

Blue-Green Deployment (create new service)

newService

```
String AppName="myZer"
String release-tag="myZer_4_8"
String prodProjectName="myZer"
String AppName-G=$AppName+"-G"
String AppName-B=$AppName+"-B"

String count = sh script: "oc get route $AppName -n $prodProjectName | grep $AppName-G  | wc -l | tr -d \"\n\"", return Stdout: true

// echo "count = '$count'"
if (count == "1"){
// Green is active on the route
   echo "$AppName-G is active."
   activeSvc = $AppName-G
   targetSvc = $AppName-B    }
else{
// Blue is active on the route
   echo "$AppName-B is active."
   activeSvc = $AppName-B
   targetSvc = $AppName-G    }
//
// create the new release and name the Pod with the targetSvc name
//
oc new-app $AppName:$release-tag —name=$targetSvc —labels=$targetSvc
```

Blue-Green Deployment (switch service)

Switch (also used for rollback purpose)

```
String AppName="myZer"
String release-tag="myZer_4_8"
String prodProjectName="myZer"
String AppName-G=$AppName+"-G"
String AppName-B=$AppName+"-B"

String count = sh script: "oc get route $AppName -n $prodProjectName | grep $AppName-G  | wc -l | tr -d \"\n\"", return Stdout: true

// echo "count = '$count'"
if (count == "1"){
// Green is active on the route
   echo "$AppName-G is active."
   activeSvc = $AppName-G
   targetSvc = $AppName-B    }
else{
// Blue is active on the route
   echo "$AppName-B is active."
   activeSvc = $AppName-B
   targetSvc = $AppName-G    }
//
// switch route
//
if (activeSvc == $AppName-B){
   sh "oc patch route/$AppName -p \'{\"spec\":{\"to\":{\"name\":\"$AppName-G\"}}}\' -n $prodProjectName"    }
else{
   sh "oc patch route/$AppName -p \'{\"spec\":{\"to\":{\"name\":\"$AppName-B\"}}}\' -n $prodProjectName"    }
```

Blue-Green Deployment (commit service)

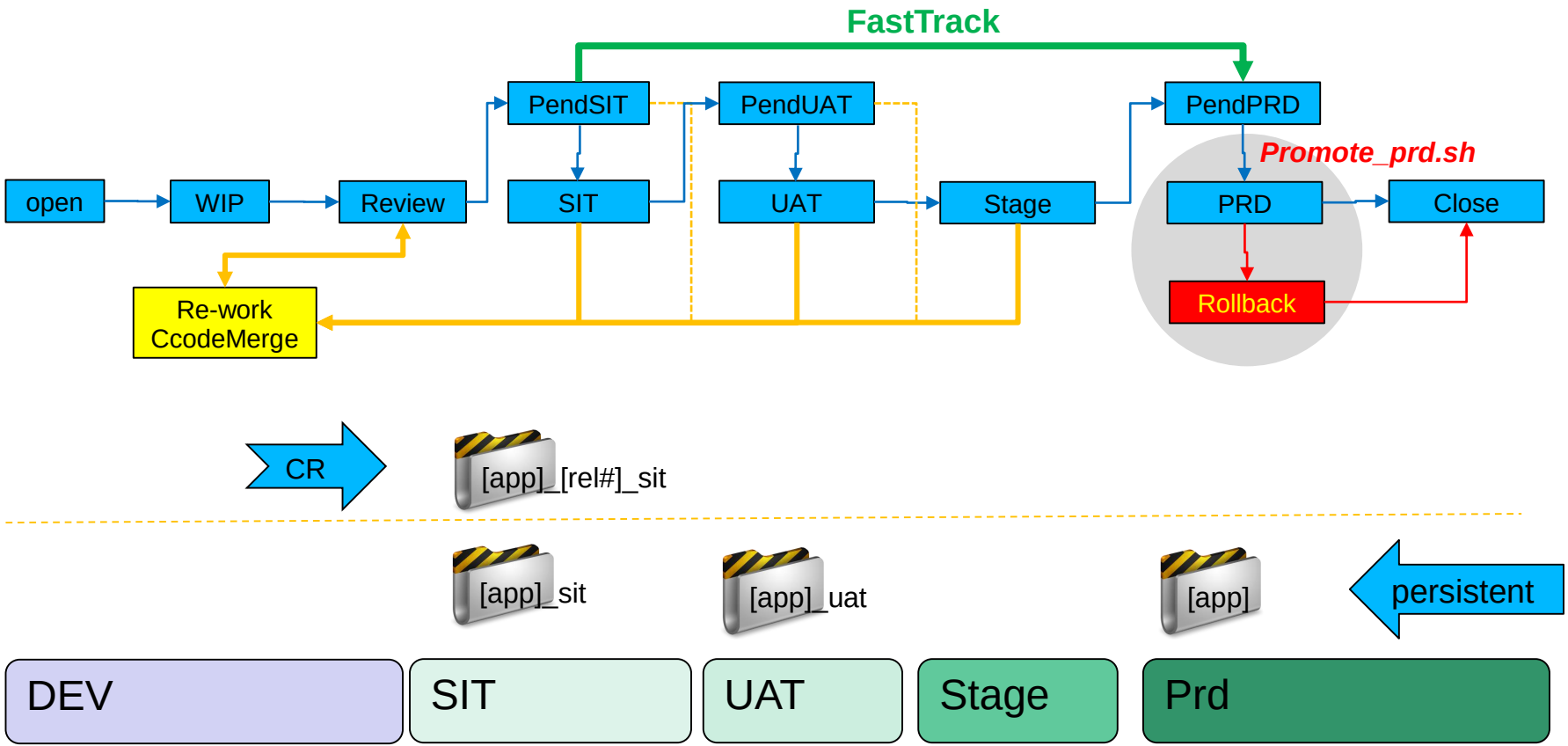
Commit (also used for rollback purpose)

```
String AppName="myZer"
String release-tag="myZer_4_8"
String prodProjectName="myZer"
String AppName-G=$AppName+"-G"
String AppName-B=$AppName+"-B"

String count = sh script: "oc get route $AppName -n $prodProjectName | grep $AppName-G  | wc -l | tr -d \"\n\"", return Stdout: true

// echo "count = '$count'"
if (count == "1"){
// Green is active on the route
   echo "$AppName-G is active."
   activeSvc = $AppName-G
   targetSvc = $AppName-B    }
else{
// Blue is active on the route
   echo "$AppName-B is active."
   activeSvc = $AppName-B
   targetSvc = $AppName-G    }
//
// Delete old service
//
if (activeSvc == $AppName-B){
   sh "oc delete pods, services -l name= $AppName-G -n $prodProjectName"    }
else{
   sh "oc delete pods, services -l name= $AppName-B -n $prodProjectName"}
```

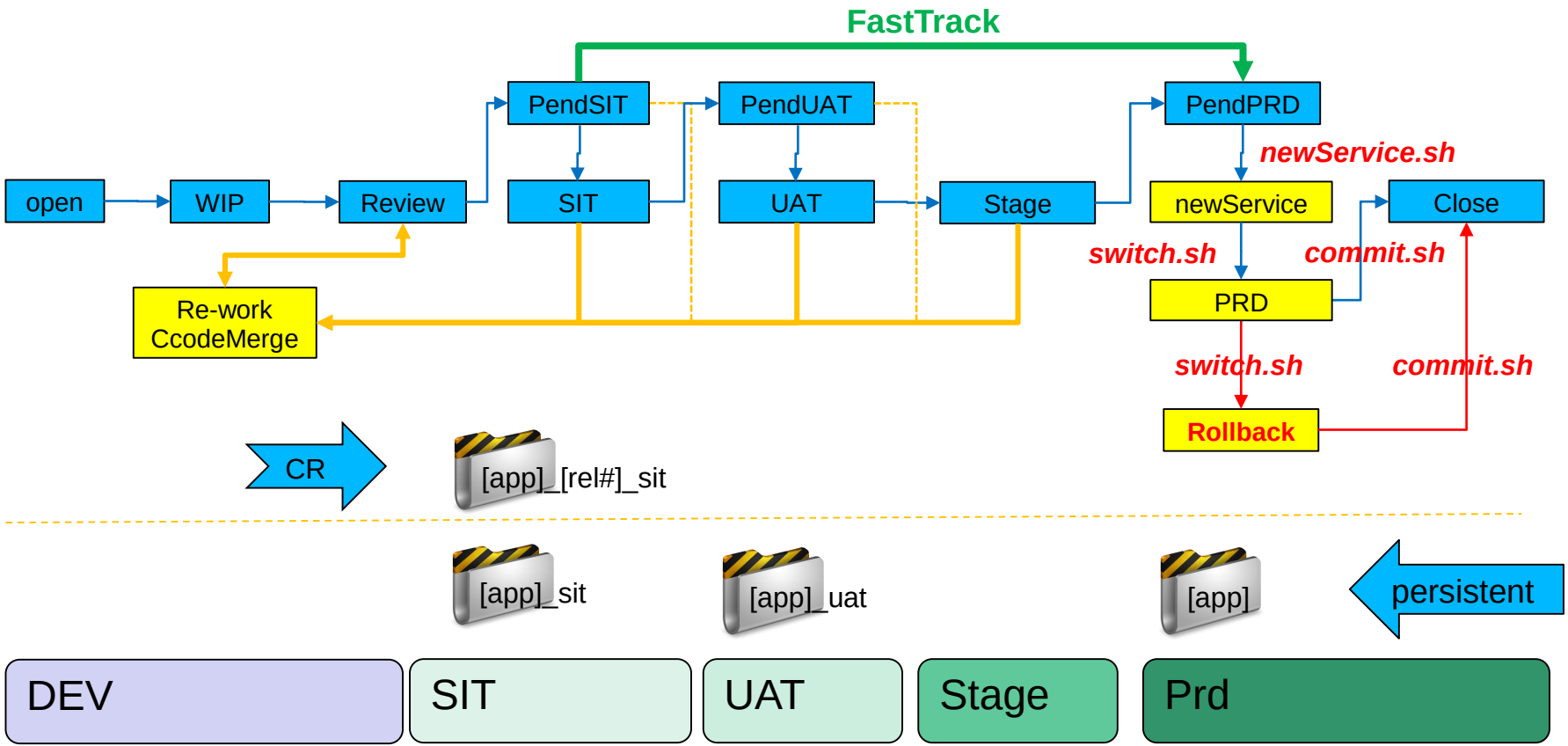
Blue-Green Deployment

Impact to existing Workflow

Pipeline / Workflow (rolling deployment)



Pipeline / Workflow extended (Blue/Green)



Blue-Green Deployment

Considerations

Blue-Green Deployment - Considerations

- ***No downtime constraints*** depends on:
    - Application session management
        - Need application architecture design review
    - Database schema changes needed in the new service
        - Rollback
        - Data patching, etc

- ***Default Kubernetes*** setting limits 1 core supporting 10 PODs
    - but can be changed


?
Q
&
A
!