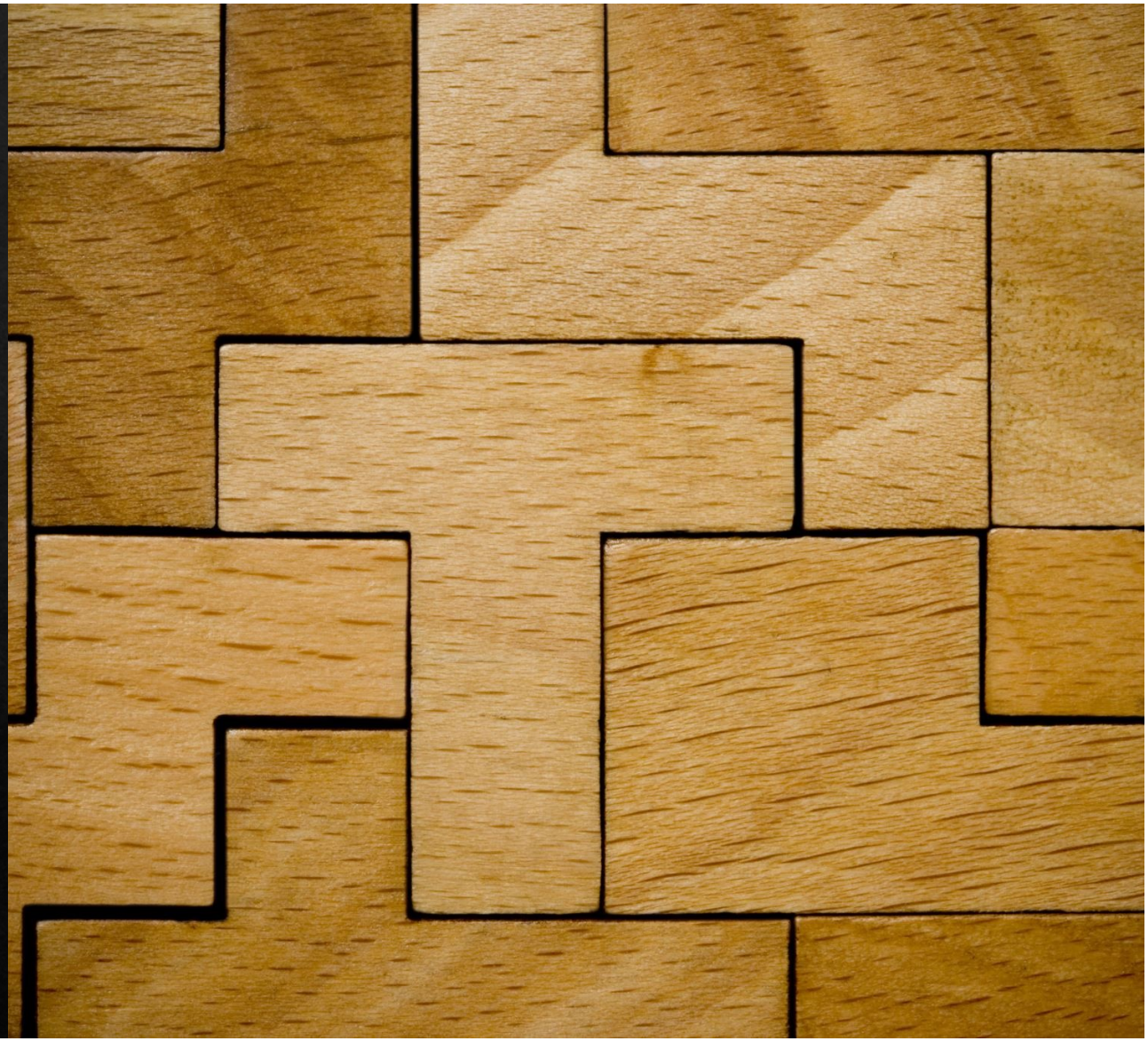
Pod Restarts – How to get your App to keep going

Glenn West

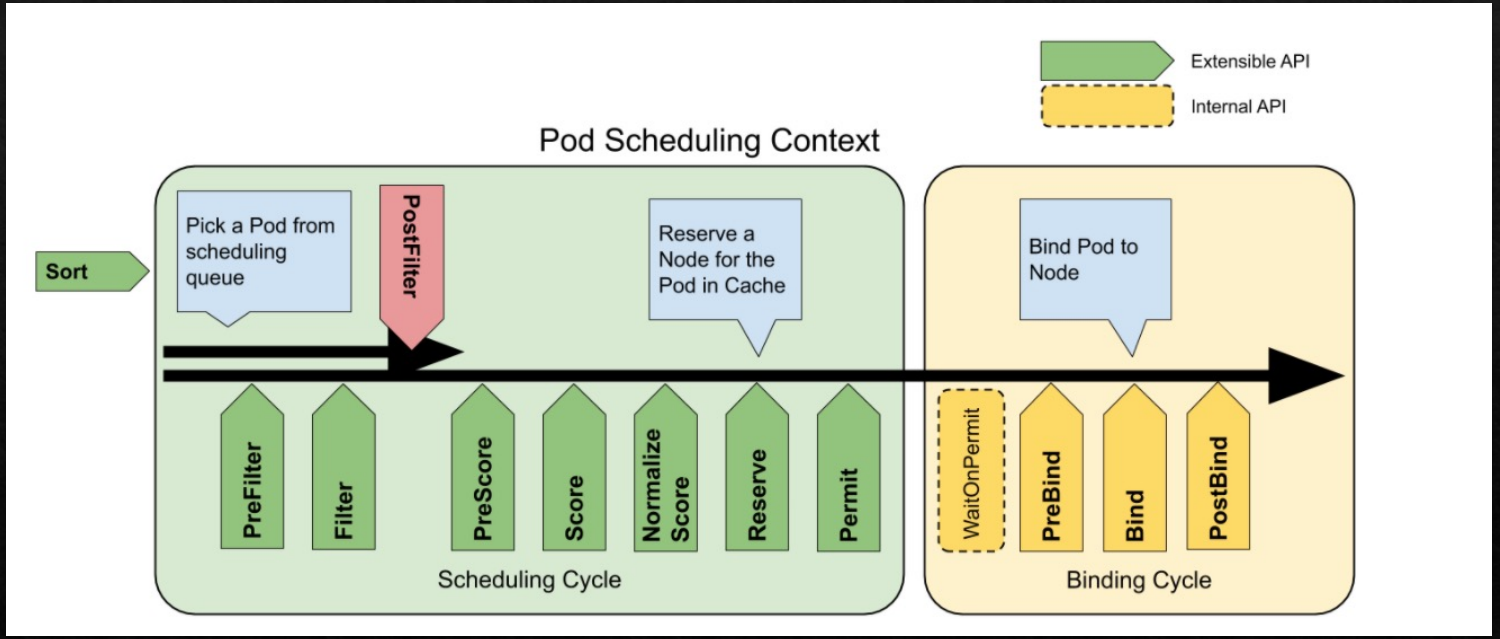
gwest@redhat.com



Overview

- OpenShift defaults to scheduling pod to node one time.
  - Pod will stay on node thru pod reboot, or node failure. On boot - pod will continue
  - This is upstream Kubernetes default.
- To have a different behavior, settings of application/pod need to be adjusted.

Pod Life Cycle



Test Setup

- 5 Node Cluster – 3 masters – 2 workers – Bare metal Install – UPI - PXE
- Openshift 4.7.24
- External “Traffic” manager like load balancer (ie dns level load balancer)

Test Application

- Rezex is the test application
  - Golang Implementation
  - Simple REST Server
  - Implemented with a two-stage build – container is 6.52mb (Bare container – NO OS)
  - Handles Health Checks
- Two possible usage models – Simple pod, or With a Replication Controller
- Example is at: https://github.com/glennswest/rezex

Simple Pod – With Taints

```
gwest@gwest@redhat resexample % cat appod.yaml
apiVersion: v1
kind: Pod
metadata:
  labels:
  name: rezex
spec:
  tolerations:
    - key: node.kubernetes.io/not-ready
      operator: Exists
      effect: NoExecute
      tolerationSeconds: 30
    - key: node.kubernetes.io/unknown
      operator: Exists
      effect: NoExecute
      tolerationSeconds: 30
    - key: node.kubernetes.io/unreachable
      operator: Exists
      effect: NoExecute
      tolerationSeconds: 30
  containers:
  - name: rezex
    image: glennswest/rezex:8084378
    args:
    - /server
    livenessProbe:
      httpGet:
        # host: my-host
        # scheme: HTTPS
        path: /healthz
        port: 8080
      initialDelaySeconds: 15
      timeoutSeconds: 1
    name: liveness
```

```
gwest@gwest@redhat resexample % cat runpod.sh
oc delete project/rezex
sleep 30
oc new-project rezex
oc create -f appod.yaml
```

```
gwest@gwest@redhat gw.lo % oc get nodes
NAME              STATUS   ROLES    AGE   VERSION
control-plane-0   Ready    master   18h   v1.20.0+558d959
control-plane-1   Ready    master   18h   v1.20.0+558d959
control-plane-2   Ready    master   18h   v1.20.0+558d959
worker-0          Ready    worker   18h   v1.20.0+558d959
worker-1          Ready    worker   18h   v1.20.0+558d959
```

```
gwest@gwest@redhat gw.lo % oc get nodes
NAME              STATUS     ROLES    AGE   VERSION
control-plane-0   Ready      master   18h   v1.20.0+558d959
control-plane-1   Ready      master   18h   v1.20.0+558d959
control-plane-2   Ready      master   18h   v1.20.0+558d959
worker-0          Ready      worker   18h   v1.20.0+558d959
worker-1          NotReady   worker   18h   v1.20.0+558d959
```

```
gwest@gwest@redhat resexample % oc get all -o wide
NAME         READY   STATUS    RESTARTS   AGE   IP             NODE       NOMINATED NODE
pod/rezex    1/1     Running   0          17h   10.128.2.18    worker-1   <none>
```

```
gwest@gwest@redhat resexample % oc get all -o wide
NAME         READY   STATUS        RESTARTS   AGE   IP             NODE       NOMINATED
pod/rezex    1/1     Terminating   0          17h   10.128.2.18    worker-1   <none>
```

```
gwest@gwest@redhat resexample % oc get all -o wide
No resources found in rezex namespace.
```

App With Replication Controller

```
gwest@gwest@redhat resexample % cat app.yaml
apiVersion: v1
kind: ReplicationController
metadata:
  name: rezex-1
spec:
  replicas: 1
  selector:
    name: rezex
  template:
    metadata:
      labels:
        name: rezex
    spec:
      containers:
      - name: rezex
        image: glennswest/rezex:8084378
        args:
        - /server
        livenessProbe:
          httpGet:
            # host: my-host
            # scheme: HTTPS
            path: /healthz
            port: 8080
          initialDelaySeconds: 15
          timeoutSeconds: 1
        name: liveness
```

Adding a Pod Disruption Budget

```
gwest@gwest@redhat resexample % cat mypdb.yaml
apiVersion: policy/v1beta1
kind: PodDisruptionBudget
metadata:
  name: rezex-pdb
spec:
  minAvailable: 1
  selector:
    matchLabels:
      name: rezex

gwest@gwest@redhat resexample %
```

Running Test

- run.sh
  - oc delete project/rezex
  - sleep 15
  - oc new-project rezex
  - oc create -f app.yaml

After Run

```
gwest@gwest@redhat resexample % oc get all -o wide
NAME                 READY   STATUS    RESTARTS   AGE   IP            NODE       NOMINATED NODE   READINESS GATES
pod/rezex-1-pv9wd    1/1     Running   0          12m   10.128.3.51   worker-1   <none>           <none>

NAME                              DESIRED   CURRENT   READY   AGE   CONTAINERS   IMAGES                          SELECTOR
replicationcontroller/rezex-1     1         1         1       12m   liveness     glennswest/rezex:8084378        name=rezex
gwest@gwest@redhat resexample % oc get nodes
NAME              STATUS   ROLES    AGE   VERSION
control-plane-0   Ready    master   41h   v1.20.0+bbbc079
control-plane-1   Ready    master   41h   v1.20.0+bbbc079
control-plane-2   Ready    master   41h   v1.20.0+bbbc079
worker-0          Ready    worker   40h   v1.20.0+bbbc079
worker-1          Ready    worker   40h   v1.20.0+bbbc079
gwest@gwest@redhat resexample %
```

Perform Test

```
gwest@gwest@redhat gw.lo % ./poweroff-vm.sh worker-1.gw.lo
Powering off VM:
gwest@gwest@redhat gw.lo %
```

```
gwest@gwest@redhat gw.lo % oc get nodes
NAME                STATUS     ROLES    AGE   VERSION
control-plane-0     Ready      master   41h   v1.20.0+bbbc079
control-plane-1     Ready      master   41h   v1.20.0+bbbc079
control-plane-2     Ready      master   41h   v1.20.0+bbbc079
worker-0            Ready      worker   40h   v1.20.0+bbbc079
worker-1            NotReady   worker   40h   v1.20.0+bbbc079
gwest@gwest@redhat gw.lo %
```

```
gwest@gwest@redhat resexample % oc get all
NAME                 READY   STATUS    RESTARTS   AGE
pod/rezex-1-pv9wd    1/1     Running   0          17m

NAME                               DESIRED   CURRENT   READY   AGE
replicationcontroller/rezex-1      1         1         0       17m
gwest@gwest@redhat resexample %
```

Success State

```
gwest@gwest@redhat resexample % oc get all
NAME                 READY   STATUS        RESTARTS   AGE
pod/rezex-1-g24lc    1/1     Running       0          3m9s
pod/rezex-1-pv9wd    1/1     Terminating   0          24m

NAME                             DESIRED   CURRENT   READY   AGE
replicationcontroller/rezex-1    1         1         1       24m
gwest@gwest@redhat resexample % oc describe replicationcontroller/rezex-1
Name:         rezex-1
Namespace:    rezex
Selector:     name=rezex
Labels:       name=rezex
Annotations:  <none>
Replicas:     1 current / 1 desired
Pods Status:  2 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  name=rezex
  Containers:
   liveness:
    Image:      glennswest/rezex:8084378
    Port:       <none>
    Host Port:  <none>
    Args:
      /server
    Liveness:     http-get http://:8080/healthz delay=15s timeout=1s period=10s #success=1 #failure=3
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Events:
  Type    Reason            Age    From                    Message
  ----    ------            ----   ----                    -------
  Normal  SuccessfulCreate  25m    replication-controller  Created pod: rezex-1-pv9wd
  Normal  SuccessfulCreate  3m19s  replication-controller  Created pod: rezex-1-g24lc
```

Replication Controller Status

```
gwest@gwest@redhat gw.lo % oc describe replicationcontroller/rezex-1
Name:         rezex-1
Namespace:    rezex
Selector:     name=rezex
Labels:       name=rezex
Annotations:  <none>
Replicas:     1 current / 1 desired
Pods Status:  1 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  name=rezex
  Containers:
   liveness:
    Image:      glennswest/rezex:8084378
    Port:       <none>
    Host Port:  <none>
    Args:
      /server
    Liveness:     http-get http://:8080/healthz delay=15s timeout=1s period=10s #success=1 #failure=3
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Events:
  Type    Reason            Age    From                    Message
  ----    ------            ----   ----                    -------
  Normal  SuccessfulCreate  2m59s  replication-controller  Created pod: rezex-1-9gk27
```

Application Restarted

```
gwest@gwest@redhat gw.lo % oc describe replicationcontroller/rezex-1
Name:         rezex-1
Namespace:    rezex
Selector:     name=rezex
Labels:       name=rezex
Annotations:  <none>
Replicas:     1 current / 1 desired
Pods Status:  2 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  name=rezex
  Containers:
   liveness:
    Image:      glennswest/rezex:8084378
    Port:       <none>
    Host Port:  <none>
    Args:
      /server
    Liveness:     http-get http://:8080/healthz delay=15s timeout=1s period=10s #success=1 #failure=3
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Events:
  Type    Reason            Age    From                    Message
  ----    ------            ----   ----                    -------
  Normal  SuccessfulCreate  7m57s  replication-controller  Created pod: rezex-1-9gk27
  Normal  SuccessfulCreate  110s   replication-controller  Created pod: rezex-1-l587r
```

```
gwest@gwest@redhat gw.lo % oc get all -o wide
NAME                 READY   STATUS        RESTARTS   AGE     IP             NODE       NOMINATED NODE   READINESS GATES
pod/rezex-1-9gk27    1/1     Terminating   0          6m29s   10.128.3.159   worker-1   <none>           <none>
pod/rezex-1-l587r    1/1     Running       0          8s      10.131.0.47    worker-0   <none>           <none>

NAME                              DESIRED   CURRENT   READY   AGE     CONTAINERS   IMAGES                          SELECTOR
replicationcontroller/rezex-1   1         1         1       6m14s   liveness     glennswest/rezex:8084378   name=rezex
gwest@gwest@redhat gw.lo %
```

Implement a Replication Controller

- Its job is to monitor that you have x pods up. And to reschedule pods based on your configuration.
- Replication Controller Doc

```
gwest@gwest@redhat resexample % cat appfix.yaml
apiVersion: v1
kind: ReplicationController
metadata:
  name: rezex-1
spec:
  replicas: 1
  selector:
    name: rezex
  template:
    metadata:
      labels:
        name: rezex
    spec:
      tolerations:
        - key: node.kubernetes.io/not-ready
          operator: Exists
          effect: NoExecute
          tolerationSeconds: 30
        - key: node.kubernetes.io/unknown
          operator: Exists
          effect: NoExecute
          tolerationSeconds: 30
        - key: node.kubernetes.io/unreachable
          operator: Exists
          effect: NoExecute
          tolerationSeconds: 30
      containers:
      - name: rezex
        image: glennswest/rezex:8084378
        args:
        - /server
        livenessProbe:
          httpGet:
```

Best Practices - Taints

- Add Taints to Apps
  - Affects Recovery Time
  - Watch out for recovery times vs transient
- Taints and Toleration Doc

```
spec:
  tolerations:
    - key: node.kubernetes.io/not-ready
      operator: Exists
      effect: NoExecute
      tolerationSeconds: 30
    - key: node.kubernetes.io/unknown
      operator: Exists
      effect: NoExecute
      tolerationSeconds: 30
    - key: node.kubernetes.io/unreachable
      operator: Exists
      effect: NoExecute
      tolerationSeconds: 30
```

For Critical Apps – Pod Disruption Budget

- Sets how many pods must be up
- Set the minAvailable
- **oc get poddisruptionbudget --all-namespaces**
- Pod Disruptrion Budget Doc

```
gwest@gwest@redhat resexample % cat mypdb.yaml
apiVersion: policy/v1beta1
kind: PodDisruptionBudget
metadata:
  name: rezex-pdb
spec:
  minAvailable: 1
  selector:
    matchLabels:
      name: rezex

gwest@gwest@redhat resexample %
```

```
gwest@gwest@redhat resexample %  oc get poddisruptionbudget --all-namespaces
NAMESPACE                  NAME                   MIN AVAILABLE   MAX UNAVAILABLE   ALLOWED DISRUPTIONS   AGE
openshift-etcd             etcd-quorum-guard      N/A             1                 1                     19h
openshift-ingress          router-default         N/A             50%               0                     19h
openshift-ovn-kubernetes   ovn-raft-quorum-guard  2               N/A               1                     20h
```

ReplicaSet

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: rezex-replicaset
  labels:
    app: rezex
    tier: frontend
spec:
  # modify replicas according to your case
  replicas: 1
  selector:
    matchLabels:
      tier: frontend
  template:
    metadata:
      labels:
        tier: frontend
    spec:
      containers:
      - name: rezex
        image: glennswest/rezex:8084378
```

```
gwest@gwest@redhat resexample % oc version
Client Version: 4.6.0-0.ci-2020-08-20-144110
Server Version: 4.8.14
Kubernetes Version: v1.21.1+a620f50
gwest@gwest@redhat resexample %
```

```
gwest@gwest@redhat resexample % oc get all -o wide
NAME                            READY   STATUS        RESTARTS   AGE    IP             NODE       NOMINATED NODE   READINESS GATES
pod/rezex-replicaset-kv8kg      1/1     Running       0          5m9s   10.131.0.58    worker-0   <none>           <none>
pod/rezex-replicaset-ph65w      1/1     Terminating   0          13m    10.128.2.55    worker-1   <none>           <none>

NAME                                DESIRED   CURRENT   READY   AGE   CONTAINERS   IMAGES                        SELECTOR
replicaset.apps/rezex-replicaset    1         1         1       13m   rezex        glennswest/rezex:8084378      tier=frontend
gwest@gwest@redhat resexample %
```

Issues and Problems

- When using a replication controller – need to make sure nfs mounts mode are set such that the container can use its pvc
- Make sure you balance your recovery time with temporary node unavaialbe
- Be aware of pod life cycles issues

Pod Life Cycle

- In testing, you will notice that the terminating pod stays in a terminating state.
- It will fully terminate once the node reboots.
- This is a known issue, and is partially resolved in some releases of 4.7 and fully resolved in later 4.8 releases, and all 4.9 releases
- https://access.redhat.com/solutions/6331221
- https://bugzilla.redhat.com/show_bug.cgi?id=1952224