OpenKruise

https://github.com/openkruise/kruise

OpenKruise/Kruise

Kruise is at the core of the OpenKruise project. It is a set of controllers which extends and complements Kubernetes core controllers on application workload management.

Today, Kruise offers three application workload controllers:

- Advanced StatefulSet: An enhanced version of default StatefulSet with extra functionalities such as `inplace-update`, sharding by namespace.
- BroadcastJob: A job that runs pods to completion across all the nodes in the cluster.
- SidecarSet: A controller that injects sidecar container into the pod spec based on selectors.

Please see documents for more technical information.

Several tutorials are provided to demonstrate how to use the workload controllers.

OpenKruise — 自动化部署Kubernetes应用的新方法

张振(守辰) | 阿里云高级技术专家

About me

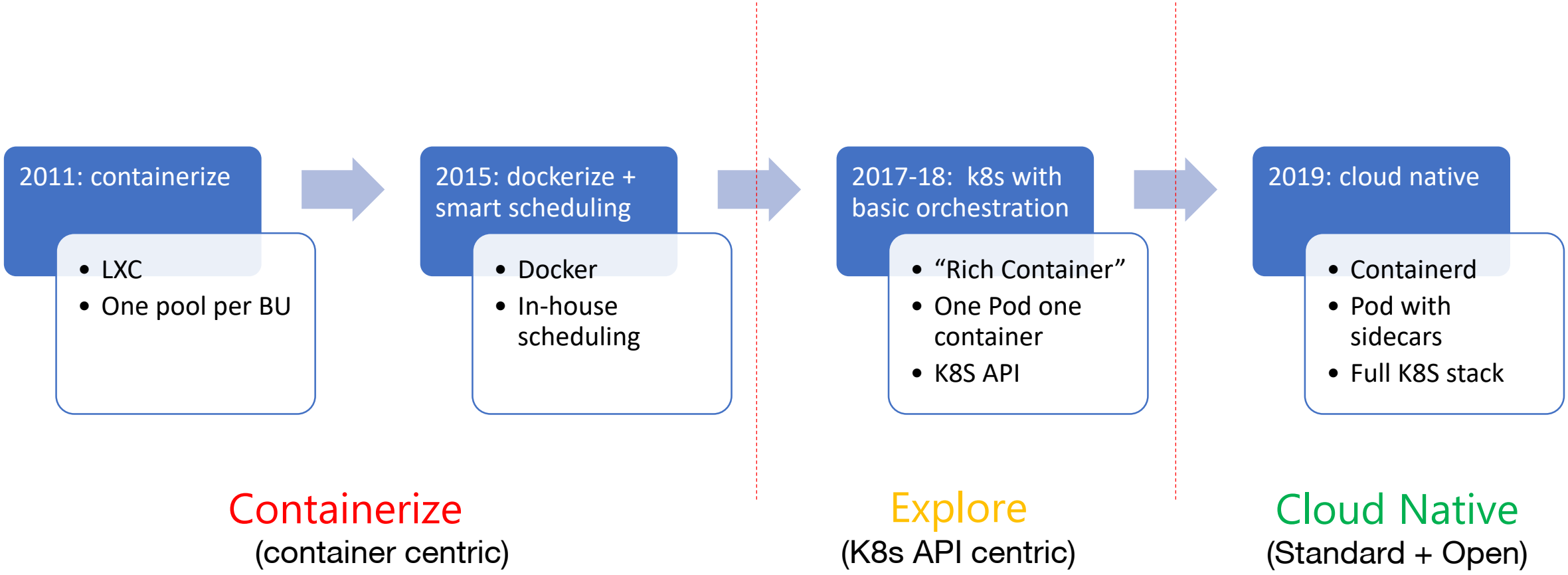
Alibaba Cloud | Worldwide Cloud Services Partner

Zhen Zhang, Staff Engineer, Alibaba Cloud

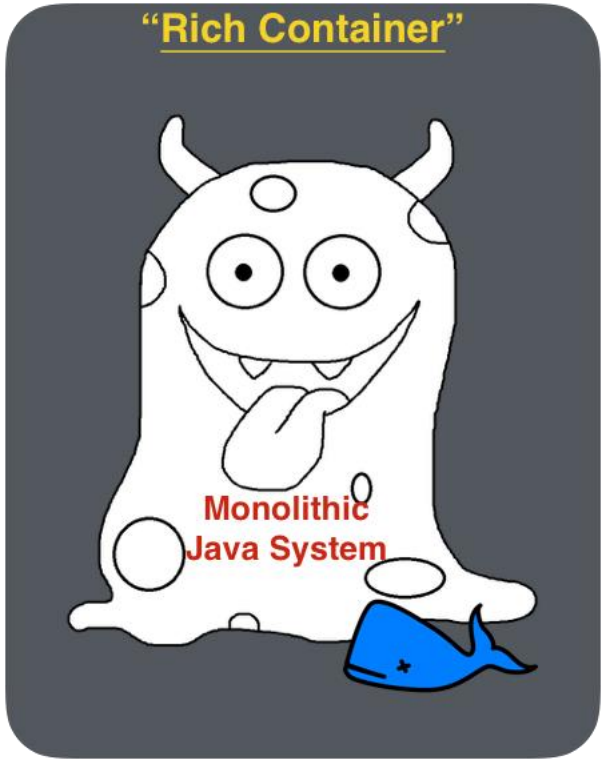
- Container platform team
- Drive the container orchestration evolution
- Open source projects

Alibaba's Journey to Cloud Native

Alibaba Cloud | Worldwide Cloud Services Partner



The “Container Headache”



Before 2018

- Java
- PID 1 process is Systemd
- ALL in ONE container (“Rich Container”), independent upgrade
  - app, sshd, log, monitoring, cache, VIP, DNS, proxy, agent, start/stop scripts …
- Traditional operating workflow
  - Start container -> SSH into container -> Start the app
  - Log files & user data are distributed everywhere in the container
- In-house orchestration & scheduling system

bind app & container together

Alibaba Cloud | Worldwide Cloud Services Partner

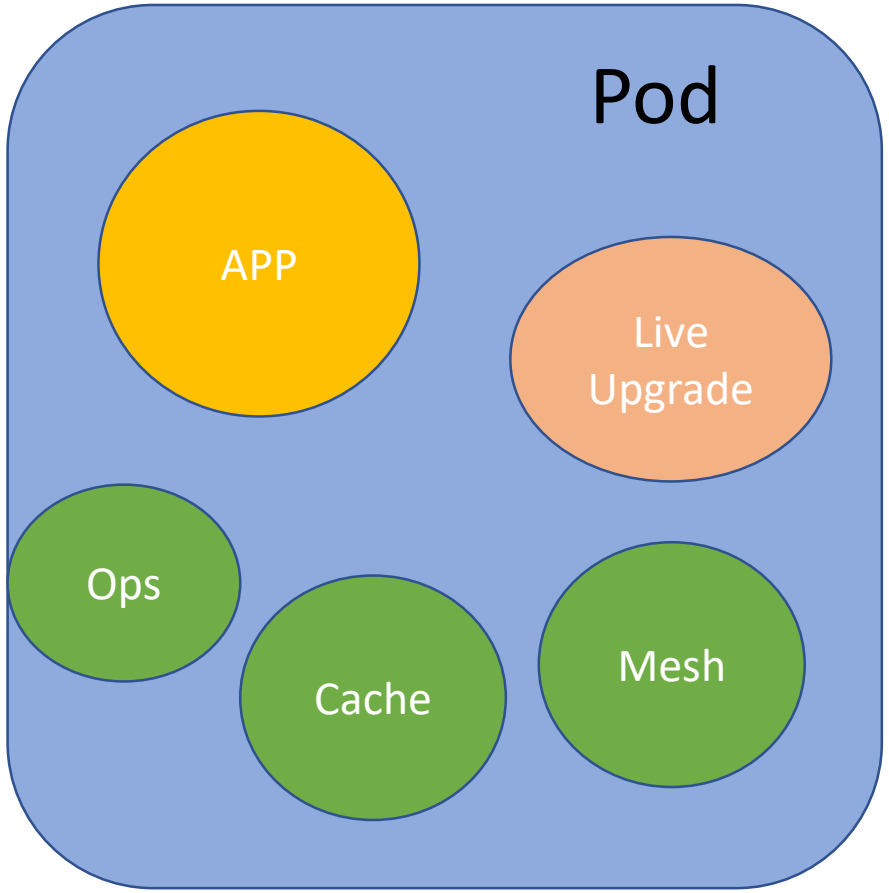
```
apiVersion: v1
kind: Pod
spec:
  containers:
  - env:
    - name: ali_start_app
      value: “no”
    name: main
    lifecycle:
      postStart:
        exec:
          command:
          - /bin/sh
          - -c
          - for i in $(seq 1 60); do [ -x /home/admin/.start ] && break ; sleep 5
            ; done; sudo -u admin /home/admin/.start>/var/log/kubeapp/start.log 2>&1
            && sudo -u admin /home/admin/health.sh>>/var/log/kubeapp/start.log 2>&1
      preStop:
        exec:
          command:
          - /bin/sh
          - -c
          - sudo -u admin /home/admin/stop.sh>/var/log/kubeapp/stop.log 2>&1
    livenessProbe:
        exec:
            command:
            - /bin/sh
            - -c
            - sudo -u admin /home/admin/health.sh>/var/log/kubeapp/health.log 2>&1
        initialDelaySeconds: 20
        periodSeconds: 60
        timeoutSeconds: 20
```

App start script

App stop script

Health check

replace rich container with pod

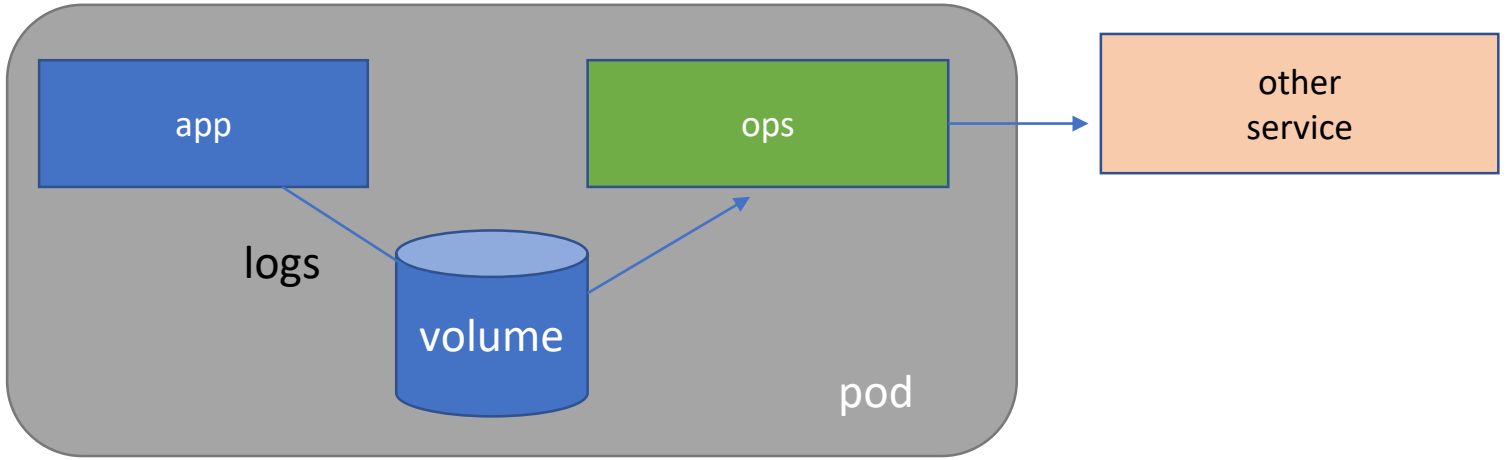


Process decomposed from app:

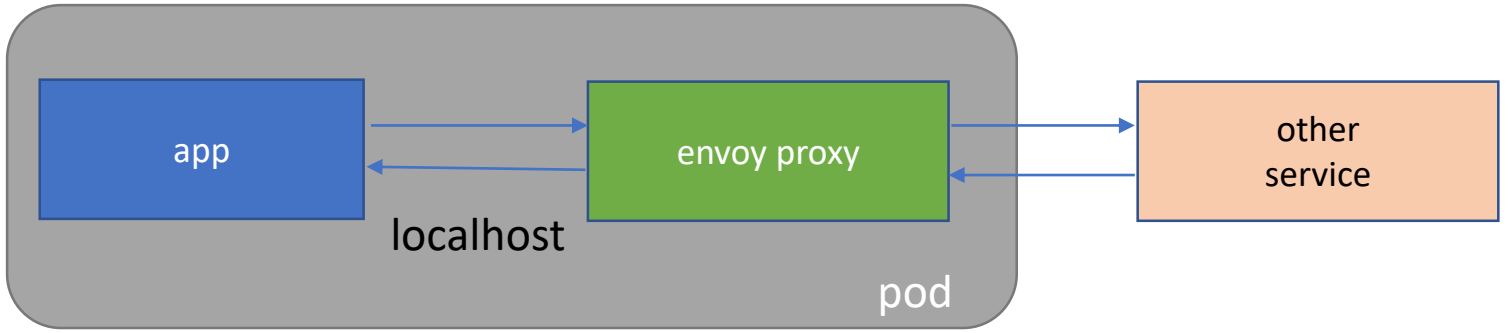
- Ops
  - logging, monitoring, debugger
- Cache
  - local cache
- Mesh
  - traffic proxy
- Live Upgrade
  - data loader

container patterns

Sidecar



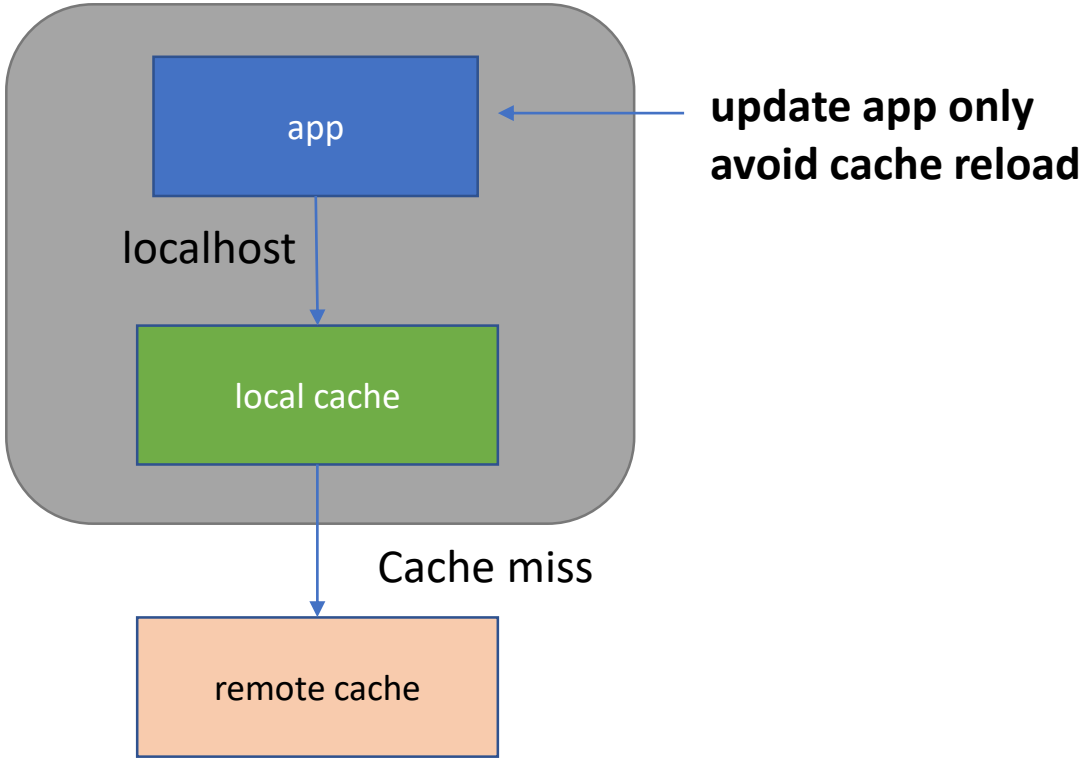
Ambassador



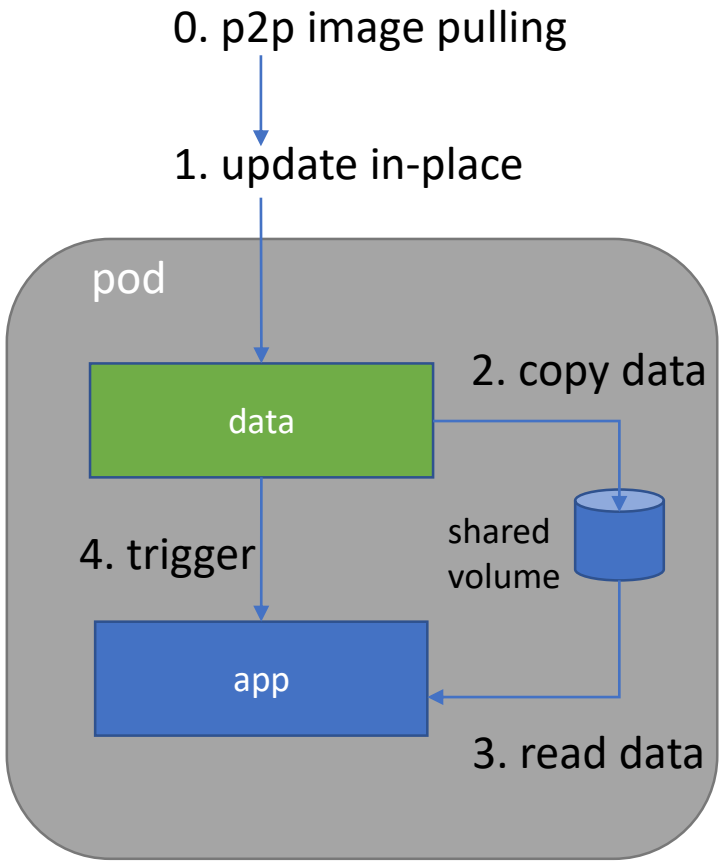
new container pattern: hot upgrade

Alibaba Cloud | Worldwide Cloud Services Partner

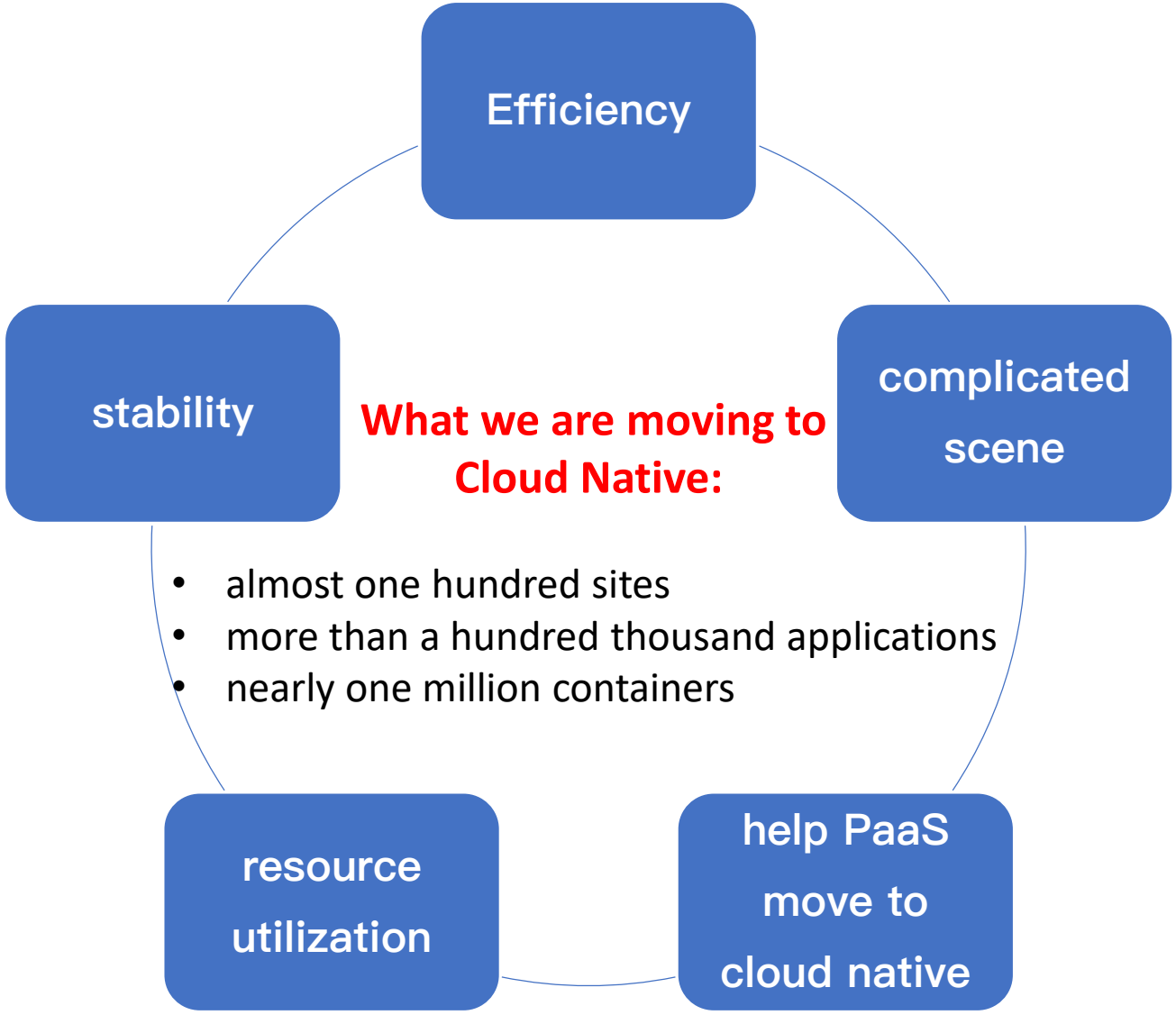
Caches



data & image & library



Workload Mgmt in Web-scale is Challenging

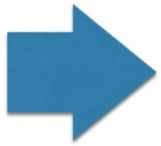


Workloads Management

Alibaba Cloud | Worldwide Cloud Services Partner

- **Kubernetes Application** = **YAML**
- **Kubernetes Workloads** = **Operating Model**
  - StatefulSet
  - Deployment
  - Job
  - CronJob
  - DaemonSet

- **Pre-defined models of**
  - Rollout Policy
  - Instance Recovery
  - Batch Deploy
  - Blue-Green Deploy
  - Canary Deploy



- **Lessons learned:**
  - They are well defined & convenient;
    - may not fit to all cases though …

OpenKruise

https://github.com/openkruise/kruise

OpenKruise/Kruise

Kruise is at the core of the OpenKruise project. It is a set of controllers which extends and complements Kubernetes core controllers on application workload management.

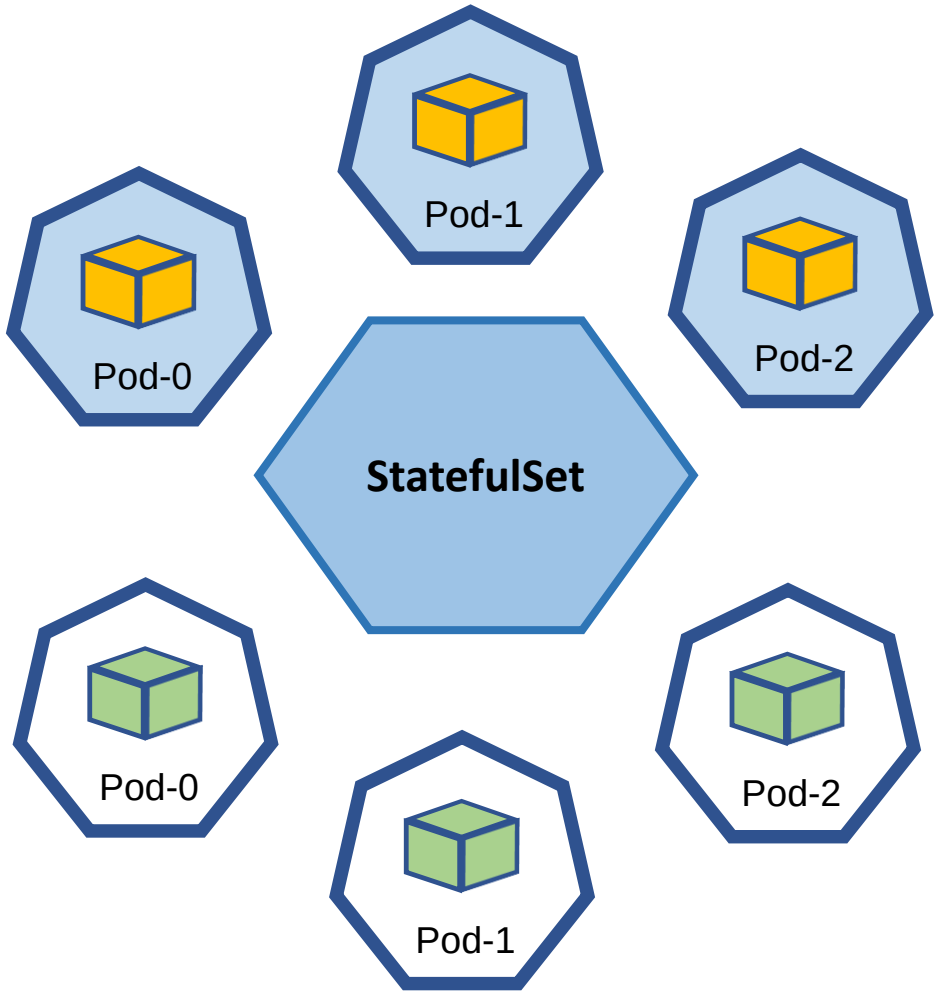
Today, Kruise offers three application workload controllers:

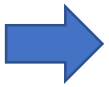
- Advanced StatefulSet: An enhanced version of default StatefulSet with extra functionalities such as `inplace-update`, sharding by namespace.
- BroadcastJob: A job that runs pods to completion across all the nodes in the cluster.
- SidecarSet: A controller that injects sidecar container into the pod spec based on selectors.

Please see documents for more technical information.

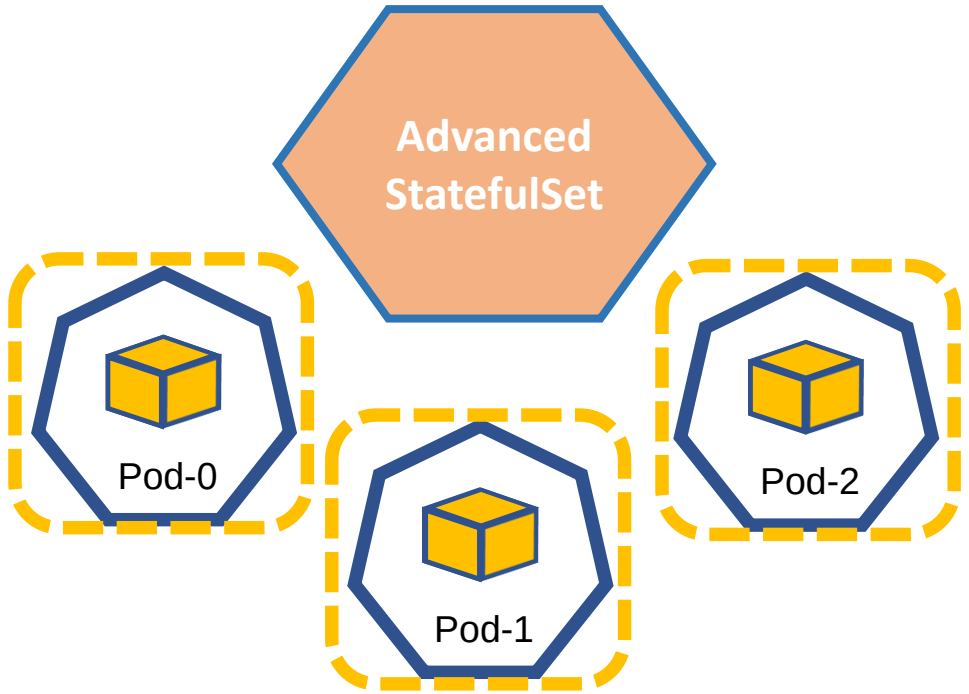
Several tutorials are provided to demonstrate how to use the workload controllers.

Kruise - AdvancedStatefulSet





MaxUnavailabe = 3



- Slow if # of replica is high
- In-place upgrade
  - Retain Pod state, avoid scheduling
- MaxUnavailable

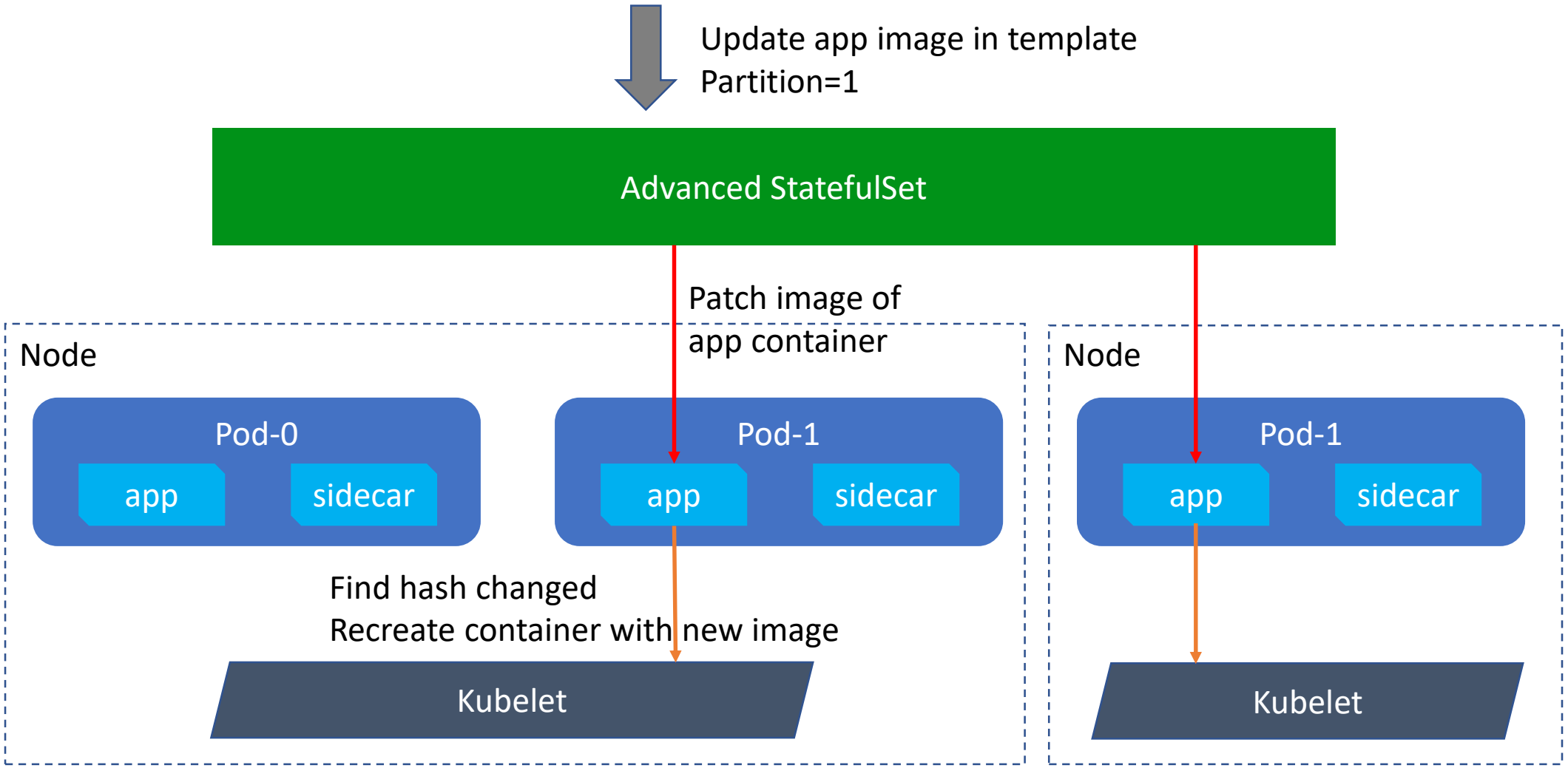
Kruise - AdvancedStatefulSet

- **AdvancedStatefulSet :**
  - **Predictability** is critical in web-scale cluster
    - We prefer **In-Place-Upgrade**, because with thousands of pods reshuffled across cluster:
      - Topology changes, image re-warm, unexpected overhead, resource allocation churn …
  - Generally, we ❤️ *StatefulSet*, **but**:
    - *SS* will still **tear down** pods during rolling upgrade
    - Less rollout strategy than Deployment

	Deployment	StatefulSet	Advanced StatefulSet
Ordering	No	Yes	Yes
Naming	Random	Ordered	Ordered
PVC reserve	No	Yes	Yes
Retry on other nodes	No	No	Yes
Rollout policy	Rolling, Recreate	Rolling, On-delete	Rolling, On-delete, **In-place**
Pause/Resume	Yes	No	Yes
Partition	No	Yes	Yes
Max unavailable	Not yet	Yes	Yes
Pre/Post update hook	No	No	Yes

InplaceSet = A in-place "StatefulSet" with more rollout strategies

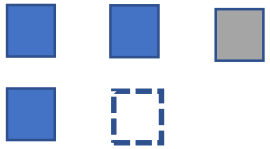
In-place update



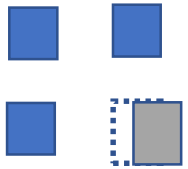
Update comparation

Alibaba Cloud | Worldwide Cloud Services Partner

rolling update



inplace update



	Recreate update	InPlace update
Cluster determinacy		👍
Efficiency of image downloading		👍
Requirement of resource		👍
Rescheduling and service registration		👍
Recovery automatically	👍	
Support all fields update	👍	

Image preheat utilizing inplace update

Alibaba Group

No preheat
Batch 1

No preheat
Batch 2

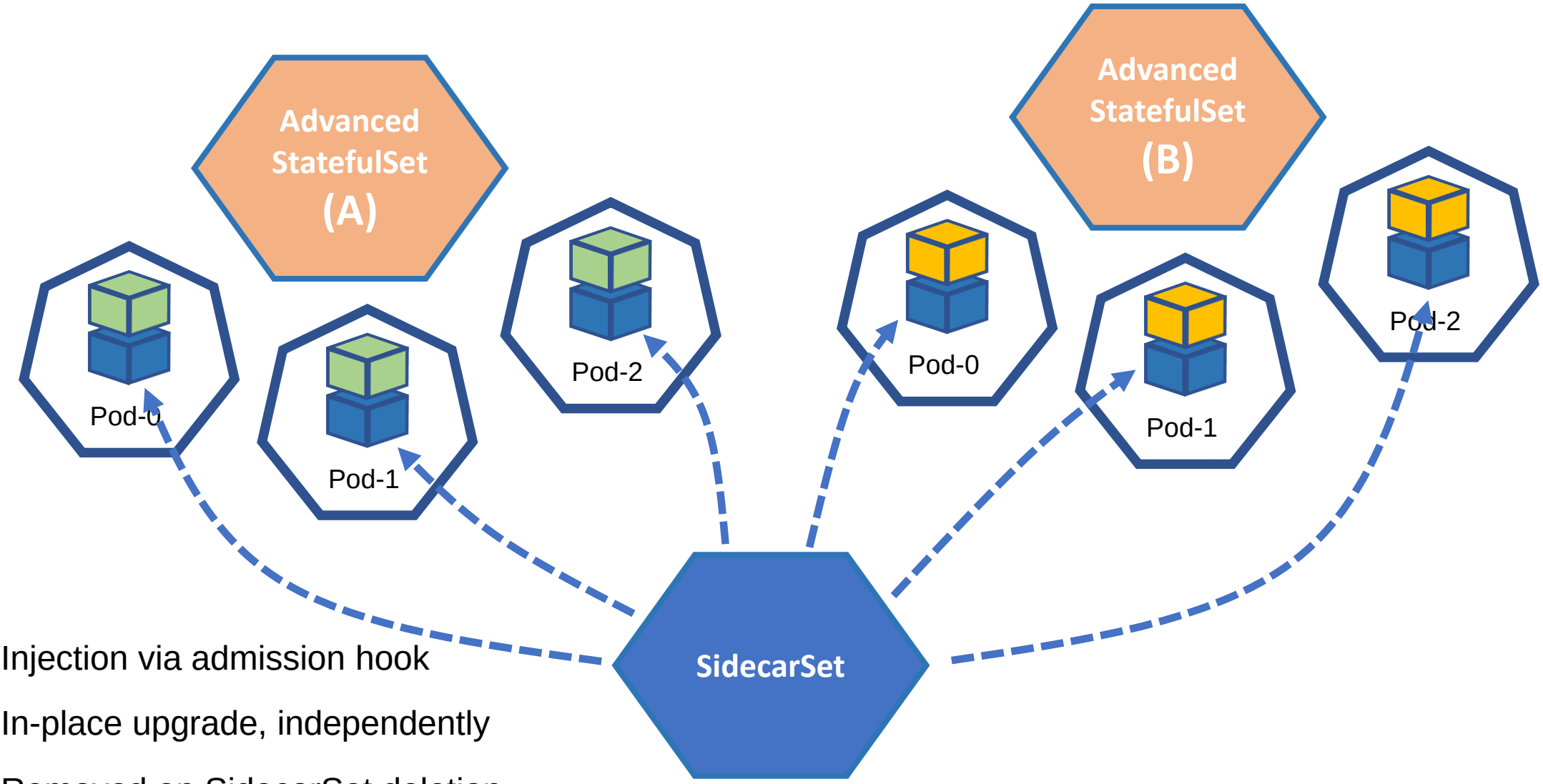
No preheat
Batch 3

Preheat
Batch 1

Preheat
Batch 2

Preheat
Batch 3

Kruise - SidecarSet



- Injection via admission hook
- In-place upgrade, independently
- Removed on SidecarSet deletion

Sidecar management

Alibaba Cloud | Worldwide Cloud Services Partner

Defined in app workloads

1. Hard to manage when there are lots of applications and workloads
2. Application developers don’t know which sidecar to use
3. How to update sidecar in too many workloads

…

Injected by SidecarSet

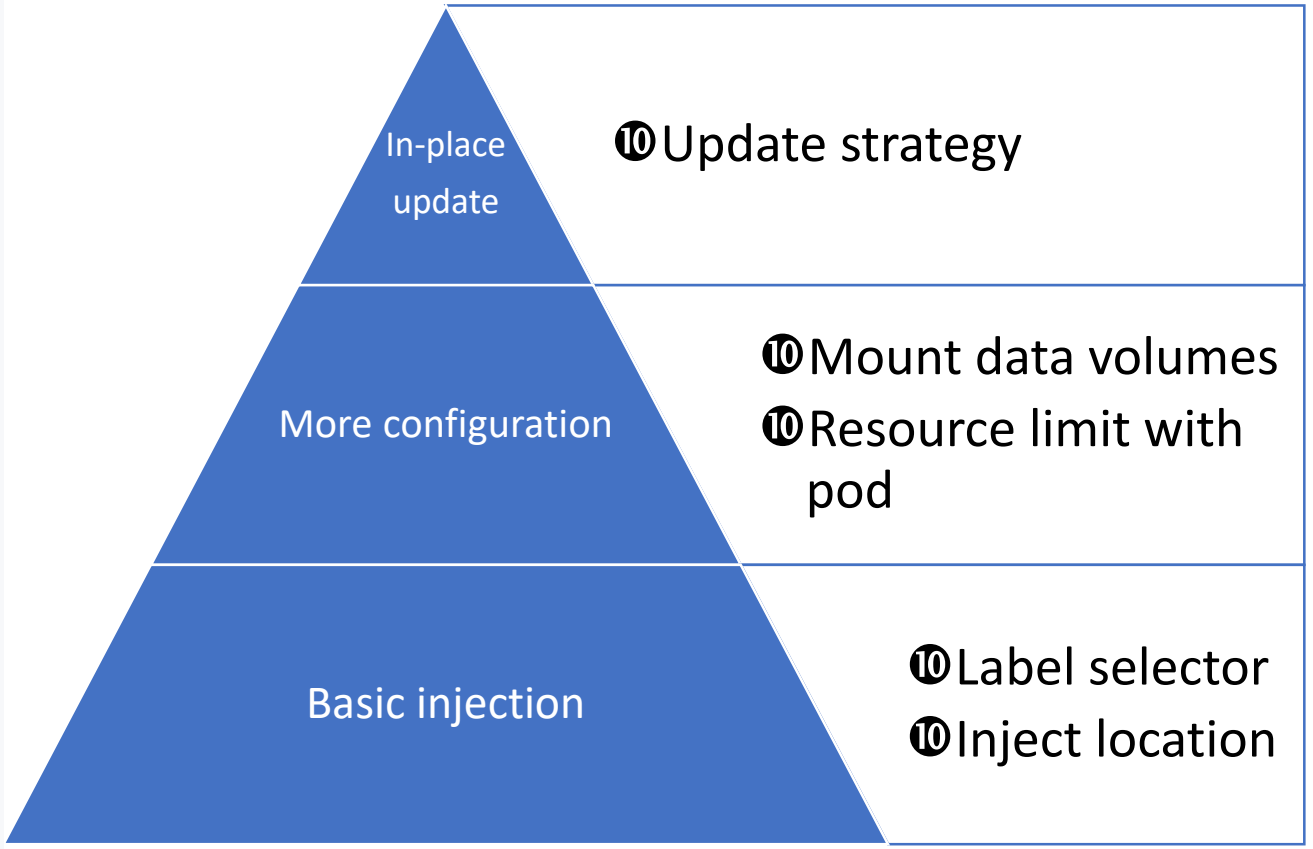
1. Define sidecars alone, part from application workloads
2. Application developers have not to care about sidecars
3. Update sidecar containers in-place, no effect on applications

…

What can SidecarSet do

Alibaba Cloud | Worldwide Cloud Services Partner

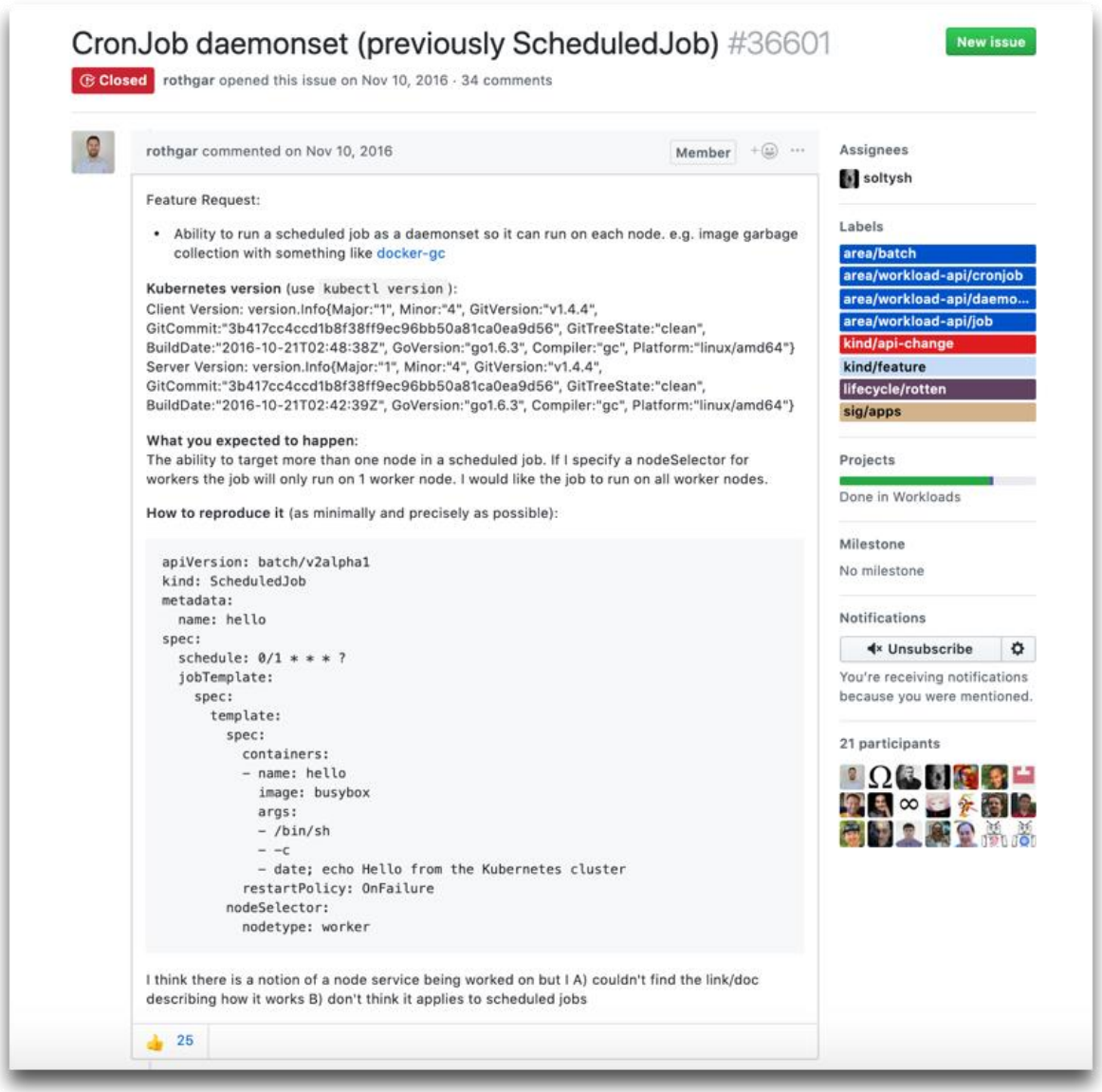
```yaml
apiVersion: apps.kruise.io/v1alpha1
kind: SidecarSet
metadata:
  name: test-sidecarset
spec:
  selector:
    matchLabels:
      app: nginx
  strategy:
    rollingUpdate:
      maxUnavailable: 2
  containers:
  - name: sidecar1
    image: centos:6.7
    command: ["sleep", "999d"] # do nothing at all
    volumeMounts:
    - name: log-volume
      mountPath: /var/log
  volumes: # this field will be merged into pod.spec.volumes
  - name: log-volume
    emptyDir: {}
```



What can Broadcastjob do

BroadcastJob

- A blend of DaemonSet and Job
  - Run pods on all machines exactly once
- Use case: software upgrade, node validator, node labeler etc
  - and tons of other use cases in this issue #36601



CronJob daemonset (previously ScheduledJob) #36601

New issue

Closed rothgar opened this issue on Nov 10, 2016 · 34 comments

rothgar commented on Nov 10, 2016 Member

Feature Request:

- Ability to run a scheduled job as a daemonset so it can run on each node. e.g. image garbage collection with something like docker-gc

Kubernetes version (use `kubectl version`):
Client Version: version.Info{Major:"1", Minor:"4", GitVersion:"v1.4.4", GitCommit:"3b417cc4ccd1b8f38ff9ec96bb50a81ca0ea9d56", GitTreeState:"clean", BuildDate:"2016-10-21T02:48:38Z", GoVersion:"go1.6.3", Compiler:"gc", Platform:"linux/amd64"}
Server Version: version.Info{Major:"1", Minor:"4", GitVersion:"v1.4.4", GitCommit:"3b417cc4ccd1b8f38ff9ec96bb50a81ca0ea9d56", GitTreeState:"clean", BuildDate:"2016-10-21T02:42:39Z", GoVersion:"go1.6.3", Compiler:"gc", Platform:"linux/amd64"}

What you expected to happen:
The ability to target more than one node in a scheduled job. If I specify a nodeSelector for workers the job will only run on 1 worker node. I would like the job to run on all worker nodes.

How to reproduce it (as minimally and precisely as possible):

```
apiVersion: batch/v2alpha1
kind: ScheduledJob
metadata:
  name: hello
spec:
  schedule: 0/1 * * * ?
  jobTemplate:
    spec:
      template:
        spec:
          containers:
          - name: hello
            image: busybox
            args:
            - /bin/sh
            - -c
            - date; echo Hello from the Kubernetes cluster
          restartPolicy: OnFailure
        nodeSelector:
          nodetype: worker
```

I think there is a notion of a node service being worked on but I A) couldn't find the link/doc describing how it works B) don't think it applies to scheduled jobs

25

Assignees: soltysh

Labels: area/batch, area/workload-api/cronjob, area/workload-api/daemo..., area/workload-api/job, kind/api-change, kind/feature, lifecycle/rotten, sig/apps

Projects: Done in Workloads

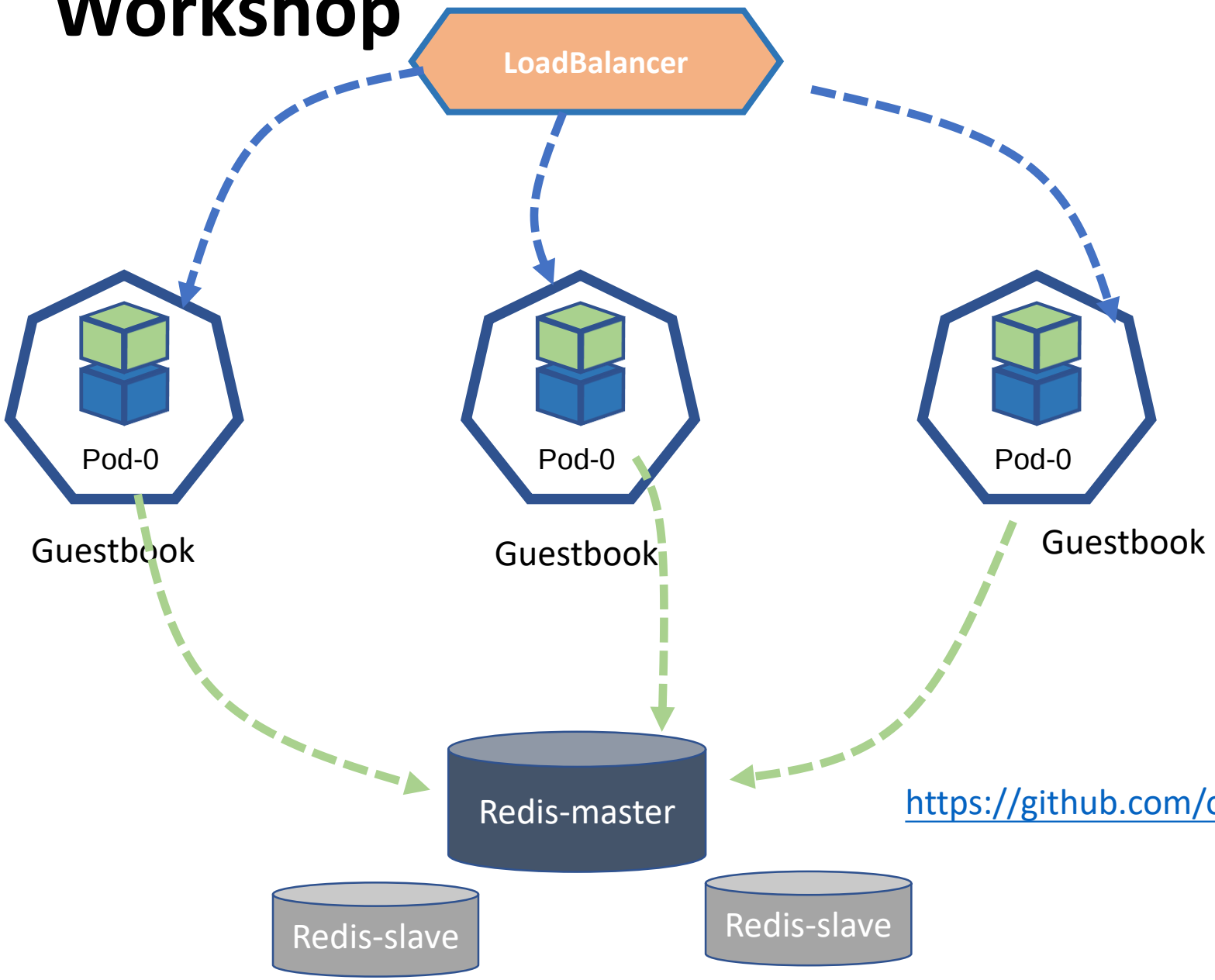
Milestone: No milestone

Notifications: Unsubscribe

You're receiving notifications because you were mentioned.

21 participants

Workshop



Workshop: Guestbook app:

1. Install application using `kubectl apply`
2. In-place update Pod
3. Hot upgrade application
4. Inject sidecar container
5. Image pre-download

https://github.com/openkruise/kruise/tree/master/docs/tutorial

What's more?

Alibaba Cloud | Worldwide Cloud Services Partner

RoadMap: https://github.com/openkruise/kruise/projects/1

- The real app management Workload used in Alibaba
- Workloads like SchedPatched/PodMarker/BatchAdoption
- Others like AHPA/AutoPilot …
- More contribution from community

Alibaba Cloud | Worldwide Cloud Services Partner