



Worldwide Cloud Services Partner

© Copyright by Alibaba Cloud All rights reserved

Release the Kraken: Bring Sidecar Containers to Next Level

Di Xu (Ant Financial)

Xiaoyu Zhang (Alibaba Cloud)

Who We Are



Di Xu
Kubernetes Member
Top 50 Code Contributor
Ant Financial
@dixudx



Xiaoyu Zhang
Kubernetes Member
Alibaba Cloud
@zhangxiaoyu-zidif

Agenda

CONTENT

**Sidecar
Introduction**

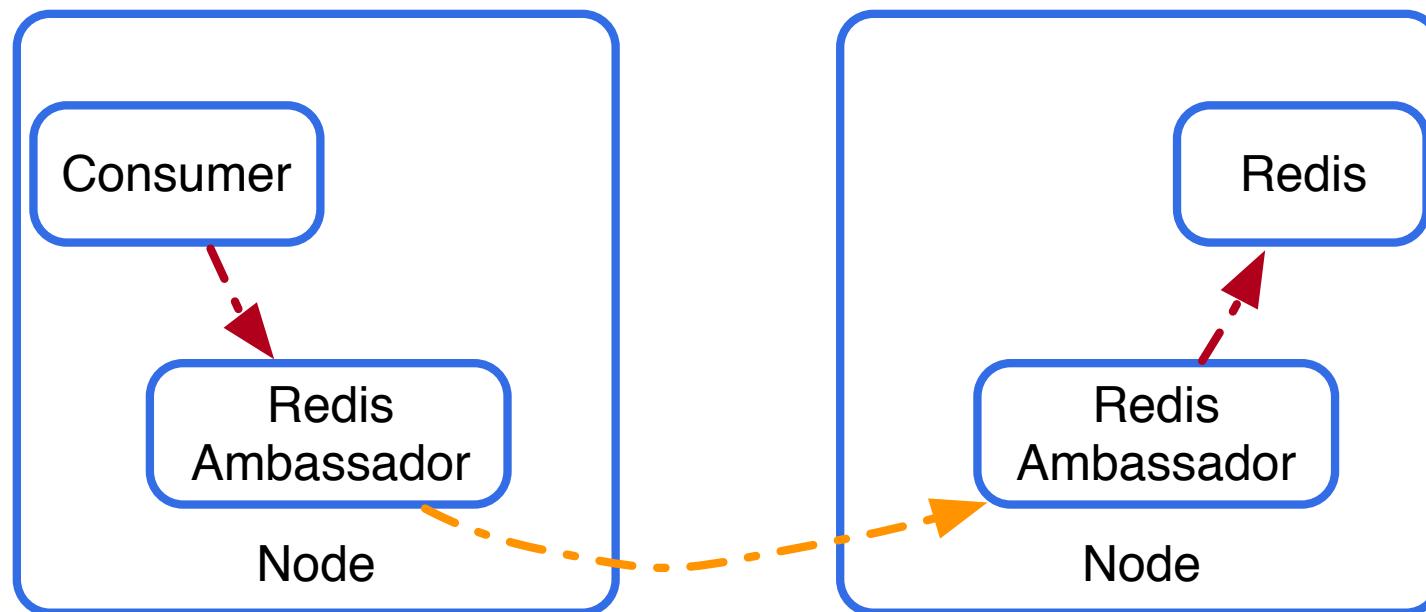
**Sidecar Use
Cases**

**Challenges &
Our Solutions**

What's Next

Sidecar is NOT a New Thing...

- A design pattern
- Docker Swarm: Ambassador



What Can Sidecars Do?

- Log Agent / Log Forwarding, such as Fluentd;
- Service Mesh, like Istio, Linkerd;
- Ambassador / Proxy;
- Act as LivenessProber to check if some component works well;
- Other accessory works, like copying files and etc;
- ...

That's All? No, We Want More!

This pattern is best used when there is a clear difference between a **primary** container and any **secondary** tasks that need to be done for it.

The demands of using sidecar containers in production environments are **rapidly increasing...**

Use Case 1

- **Sidecar Order / Dependency**

```
#1 proxy_container  
#2 mysql_client  
#3 svc_container
```

```
spec:  
  containers:  
    - name: proxy_container  
      image: myproxy:v1  
    - name: svc_container  
      image: myservice:v2  
    - name: mysql_client  
      image: mysql_client:v3.5
```

Use Case 2

- **Sidecar Management**
 - Updating sidecar container Images;
 - Mounting sidecar volumes;
 - Decoupling sidecars with normal containers;
 - ...

Use Case 2 – Cont.

- Example: Updating sidecar container Images
 - **Re-Inject** Sidecars as Istio Official Guide Suggested (<https://istio.io/docs/setup/upgrade/steps/#sidecar-upgrade>)
 - **Seriously?** Hundreds of Deployments? Thousands of Deployments?

```
containers:  
  name: istio-proxy  
  image: istio.io/proxy:1.2.9
```



```
containers:  
  name: istio-proxy  
  image: istio.io/proxy:1.3.4
```

Upstream Status

- [Sidecar KEP API implementation](#) merged.
 - Thanks Joseph Irving (@Joseph-Irving) for doing such a great job. 

Sidecar KEP API implementation

API Changes:

As this is a change to the Container spec we will be using feature gating, you will be required to explicitly enable this feature on the api server as recommended [here](#).

New field `Type` will be added to the lifecycle struct:

```
type Lifecycle struct {
    // Type
    // One of Standard, Sidecar.
    // Defaults to Standard
    // +optional
    Type LifecycleType `json:"type,omitempty" protobuf:"bytes,3,opt,name=type,casttype=LifecycleType"`
}
```

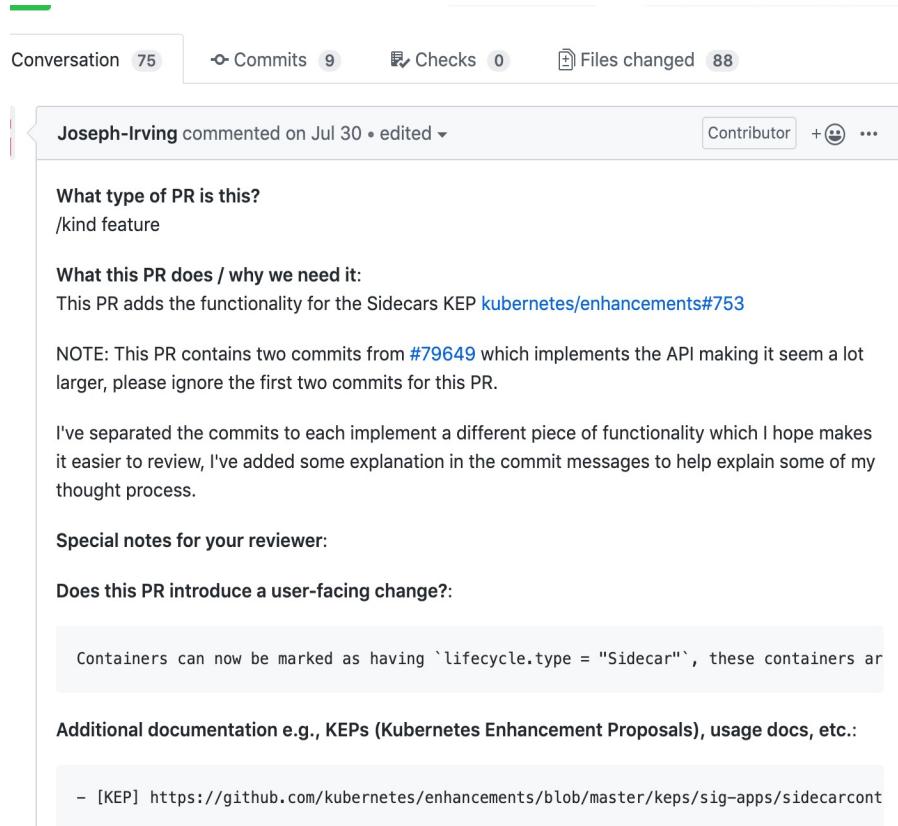
New type `LifecycleType` will be added with two constants:

```
// LifecycleType describes the lifecycle behaviour of the container
type LifecycleType string

const (
    // LifecycleTypeStandard is the default container lifecycle behaviour
    LifecycleTypeStandard LifecycleType = "Standard"
    // LifecycleTypeSidecar means that the container will start up before standard containers and be terminated after
    LifecycleTypeSidecar LifecycleType = "Sidecar"
)
```

Upstream Status

- Sidecar KEP API implementation merged.
 - Thanks Joseph Irving (@Joseph-Irving) for doing such a great job. 
- Kubelet implementation is ongoing.



The screenshot shows a GitHub pull request comment from Joseph-Irving. The comment is dated Jul 30 and edited. It includes several sections of explanatory text:

- What type of PR is this?**
/kind feature
- What this PR does / why we need it:**
This PR adds the functionality for the Sidecars KEP [kubernetes/enhancements#753](#)
- NOTE:** This PR contains two commits from #[79649](#) which implements the API making it seem a lot larger, please ignore the first two commits for this PR.
- I've separated the commits to each implement a different piece of functionality which I hope makes it easier to review, I've added some explanation in the commit messages to help explain some of my thought process.**
- Special notes for your reviewer:**
- Does this PR introduce a user-facing change?:**
- Containers can now be marked as having `lifecycle.type = "Sidecar"`, these containers ar**
- Additional documentation e.g., KEPs (Kubernetes Enhancement Proposals), usage docs, etc.:**
- [KEP] <https://github.com/kubernetes/enhancements/blob/master/keps/sig-apps/sidecarcont>**

Upstream Status

- [Sidecar KEP API implementation](#) KEP merged.
 - Thanks Joseph Irving (@Joseph-Irving) for doing such a great job. 
- [Kubelet implementation](#) is ongoing.
- [Supports PreSidecars and PostSidecars](#) KEP is ongoing.

Supports PreSidecars and PostSidecars

Open Supports PreSidecars and PostSidecars #1135 Changes from all commits ▾ File filter... ▾ Jump to... ▾ ⚙ ▾

0 / 1 files viewed ⓘ Review changes ▾

81 keps/sig-apps/sidecarcontainers.md

the `container.lifecycle` spec: `type` which can be either `Standard` (default) or `'Sidecar'`.

the `container.lifecycle` spec: `type` which can be either `Standard` (default), `'PreSidecar'` or `'PostSidecar'`.

This conversation was marked as resolved by dixudx Show conversation

120
121 e.g:
122 `'''yaml`

133
134 `@@ -131,23 +144,33 @@ spec:`
131 + `- name: myapp`
132 `image: myapp`
133 `command: ['do something']`
134 - `- name: sidecar`
135 - `image: sidecar-image`
136 `lifecycle:`
137 - `type: Sidecar`
138 `command: ["do something to help my app"]`

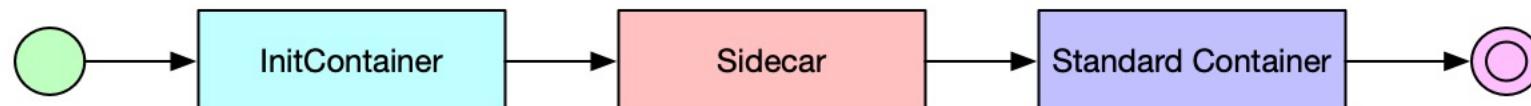
144 `- name: myapp`
145 `image: myapp`
146 `command: ['do something']`
147 + `- name: presidecar`
148 + `image: presidecar-image`
149 `lifecycle:`
150 + `type: PreSidecar`
151 `command: ["do something to help my app"]`
152 + `- name: postsidecar`
153 + `image: postsidecar-image`
154 + `lifecycle:`
155 + `type: PostSidecar`
156 + `command: ["do something else to help my app"]`

139
157

PreSidecar & PostSidecar

- Pre Startup

Sometimes sidecars should start before normal containers to do some preparations such as releasing credentials, creating shared volumes, copying some files and etc.

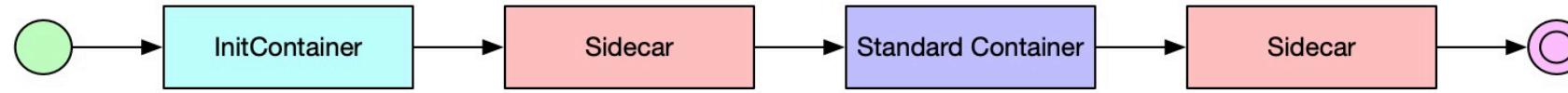


- Post Startup

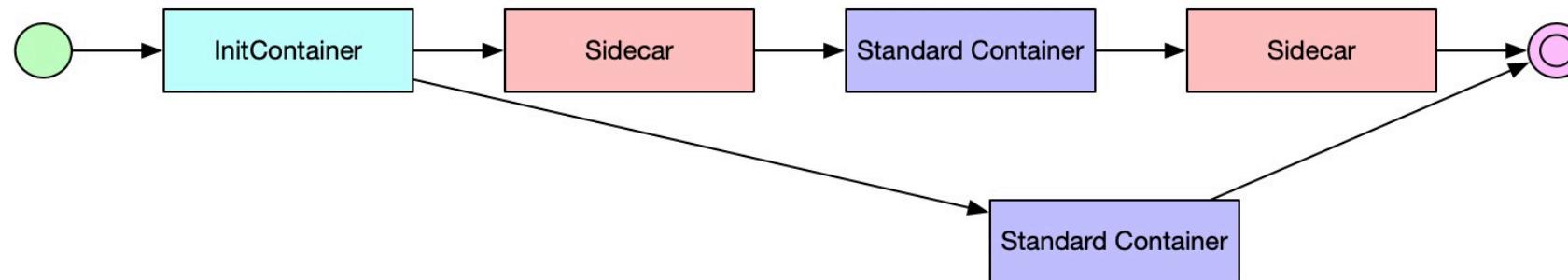
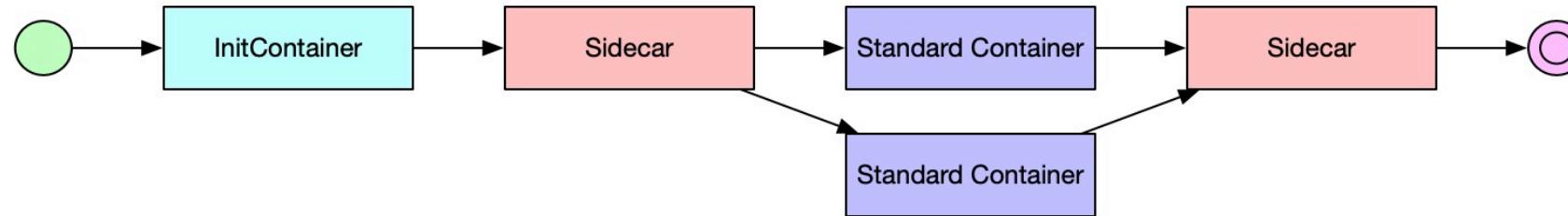
An application that has a proxy container acting as a sidecar may fail when it starts up as it's unable to communicate until its proxy has started up successfully. Readiness probes don't help if the application is trying to talk outbound.



More complex: DAG



Ref : KEP
[Supports](#)
[PreSidecars](#) and
[PostSidecars](#) is
ongoing.



By the way, if the startup dependency is complex enough, a **DAG** order is fairly necessary.

Alibaba Cloud & Ant Financial Status

Screenshot of the GitHub repository page for OpenKruise.

The URL in the address bar is github.com/openkruise.

The repository name is **OpenKruise**.
Description: Automate everything on Kubernetes.
Website: <https://openkruise.io>

Repository statistics: 4 Repositories, 1 People, 0 Projects.

Search bar: Find a repository...
Filters: Type: All, Language: All.

The screenshot shows the kruise repository details:

Name: **kruise**
Description: Automate application workloads management on Kubernetes
Tags: kubernetes, container, orchestration, cloudnative, workloads
License: Go
Apache-2.0
Issues: 145
Stars: 1,056
Needs help: 11 (1 issue needs help)
Pull requests: 2
Last updated: Updated yesterday



OpenKruise/Kruise



- A set of controllers which extends and complements [Kubernetes core controllers](#) on workload management.
- Three workloads are ready for now (More will be released soon.)
 - AdvancedStatefulset
 - BroadcastJob
 - **SidecarSet**
- See the project roadmap here
(<https://github.com/openkruise/kruise/projects>)

Ref: A [video](#) (<https://www.youtube.com/watch?v=elB7reZ6eAQ>) by [Lachlan Evenson](#) is also provided for demonstrating the controllers.

The problem before we have SidecarSet is

Sidecars are less controlled and treated
equally with normal containers. But actually
they are **different** and **NOT** normal.

How to Solve

Fine Grained Controlling

```
// SidecarContainer defines the container of Sidecar
type SidecarContainer struct {
    corev1.Container
}

// SidecarSetUpdateStrategy indicates the strategy that the SidecarSet
// controller will use to perform updates. It includes any additional parameters
// necessary to perform the update for the indicated strategy.
type SidecarSetUpdateStrategy struct {
    RollingUpdate *RollingUpdateSidecarSet `json:"rollingUpdate,omitempty"`
}

// RollingUpdateSidecarSet is used to communicate parameter
type RollingUpdateSidecarSet struct {
    MaxUnavailable *intstr.IntOrString `json:"maxUnavailable,omitempty"`
}
```

Roadmap of SidecarSet

Kruise Roadmap  Updated 6 days ago

4 To do

[M4] Support SlowStart in BroadcastJob
The flag specified the new Pods creation rate by the controller. If the flag is set to True, Pods will be created in batch mode (for example, 2, 4, 8 and so on) until reaching Parallelism number of Pods. Otherwise, all Parallelism number of Pods will be created at the same time.
Added by Fei-Guo

[M4] Selective Upgrade SidecarSet Pods
An upgrade Pod label selector is added. The new sidecar container version will only be applied to the Pods that match the upgrade selector.
Added by Fei-Guo

1 In progress

[M3] Add Partition support for SidecarSet
When upgrading SidecarSet, the number of Pods that keep the old version sidecar container can be specified through partition. This helps for performing a phased rollout.
Added by Fei-Guo

9 Done

[M2] Parallel Upgrade SidecarSet Pods
Support MaxUnavailable feature, which allows upgrading sidecar containers for multiple Pods simultaneously.
Added by Fei-Guo

[M2] Support Sidecar Volumes
A Volumes field will be added to SidecarSetSpec. When injecting sidecar containers for matching Pods during Pods creation, the storage volumes specified in SidecarSetSpec.Volumes will be merged into

Filter cards

Evolve with upstream

 [openkruise / kruise](#) Unwatch 50 Unstar 1.1k

[Code](#) [Issues 11](#) [Pull requests 2](#) [Actions](#) [Projects 1](#) [Wiki](#) [Security](#) [Insights](#)

[feature request] Using sidcar type container for SidecarSet #161

Open resouer opened this issue 9 days ago · 1 comment

 resouer commented 9 days ago Member + ...

What would you like to be added:

We may want to consider have <https://github.com/kubernetes/enhancements/blob/master/keps/sig-apps/sidecarcontainers.md> in plan as soon as it's ready.

Let's use this issue as tracker.

1. feature tracker: [kubernetes/enhancements#753](#)
2. main kubelet change: [kubernetes/kubernetes#80744](#)

Assignees
 jian-he

Labels
kind/feature-req

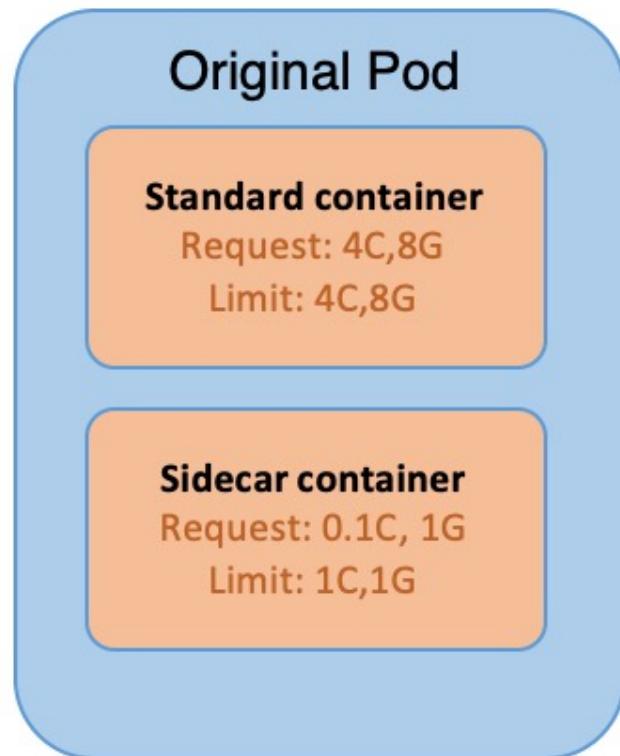
Projects
None yet

Remaining challenges shouldn't be ignored

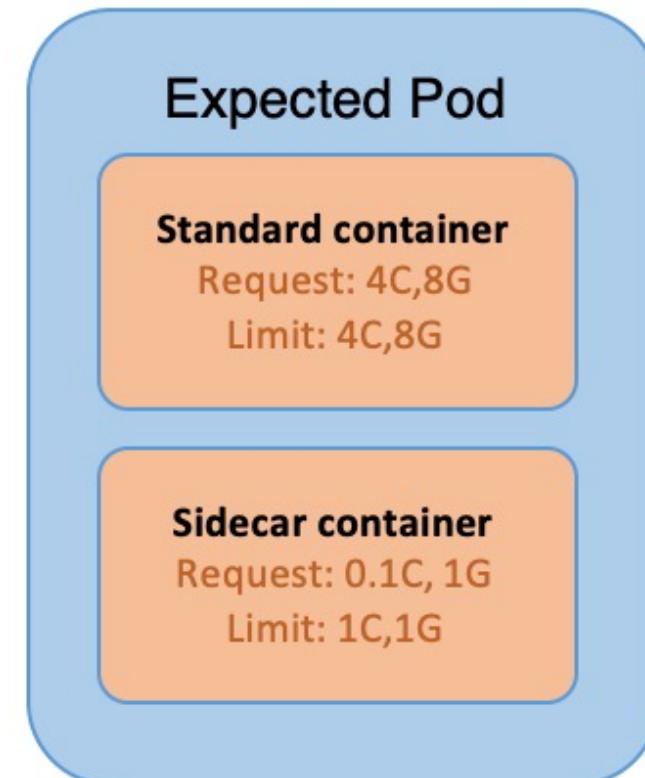
- Resource allocations for sidecar containers in various scenarios.
- Sidecar container maintain
- Failure Tolerance for sidecar containers
- ...

Resource allocations

Request: 4.1C, 9G
Limit: 5C, 9G



Pod-Level Resource Allocation
Request: 4C, 8G
Limit: 4C, 8G



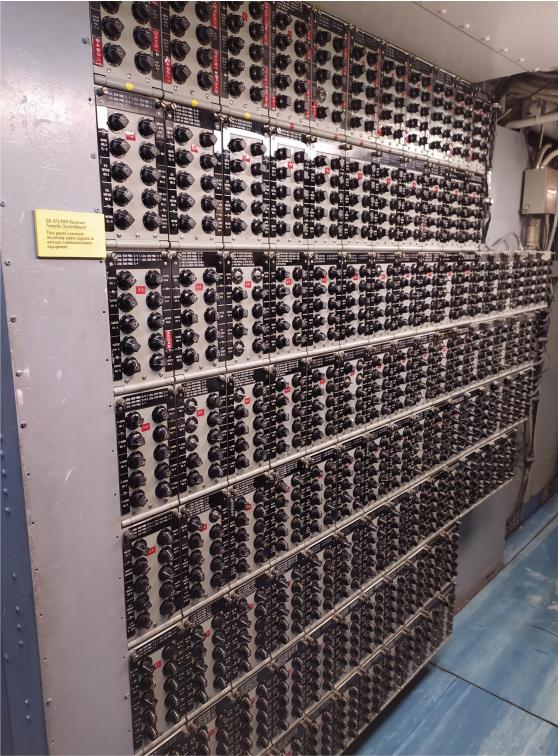
Sidecar container maintain

- Like Kruise's SidecarSet, it has independent workload.

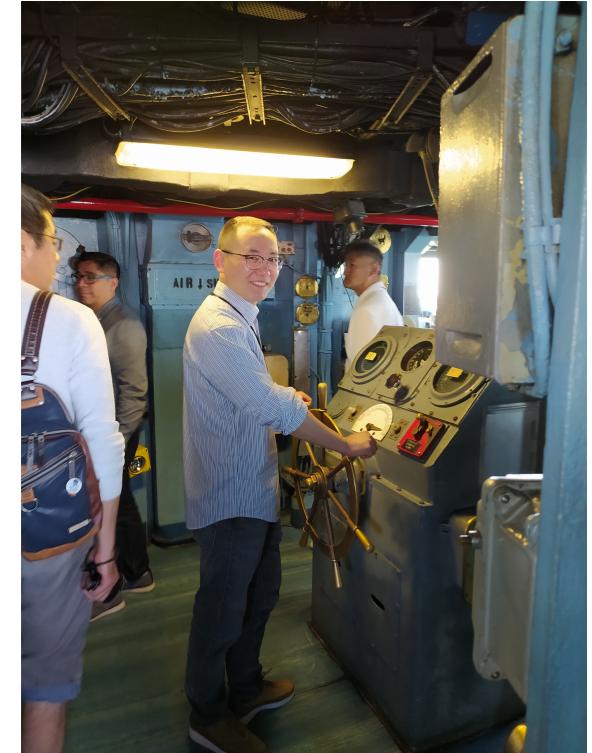
Failure Tolerance

- LivenessProbe or ReadinessProbe failed should be ignored.
- Image Update should not break down Pod Lifecycle.

Salute to USS Midway Aircraft Carrier (CV-41)



Kubernetes is not easy to handle, but far easier than operating USS Midway Aircraft.



Let's try to handle it gradually

Contact US

- Github Issue / Slack Channel
 - dixudx
 - zhangxiaoyu-zidif

Thanks