

# Deployment Troubleshooting



kubernetes Summit 2018

James

# About James

- James Lu
- Zyxel Communications Corp.
- Author of “k8s 不自賞”, ITHome  
2018 Ironman



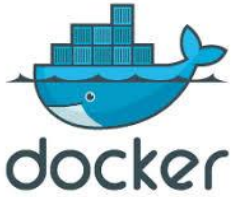
# Purpose

- Reveal errors you might encounter when you operate k8s
- Understand causes, review symptoms, check suggestions, then take actions

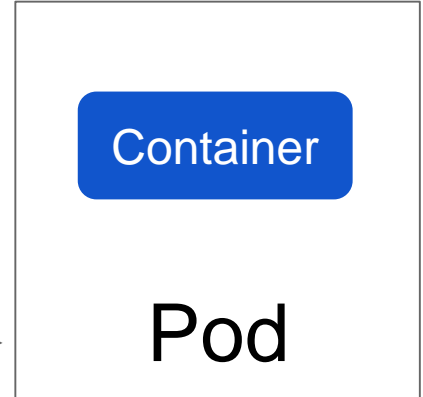


k8s master might  
encounter the  
following issues

# Container



- Wrong image
  - typo
  - inexistent image
  - without permission



- `kubectl run typo --image=james/typo:v1.0`

You will find “**ImagePullBackOff**” or “**ErrImgPull**” status via `kubectl get pod`

NAME	READY	STATUS	RESTARTS	AGE
typo-79c86f6b5f-56j6m	0/1	ImagePullBackOff	0	11m

Events section shows “**Failed to pull image**” or “**Back-off pulling image**” via `kubectl describe pod`




Type	Reason	Subshooting	Age	From	Message
Normal	MountVolume		1m	default-scheduler	Successfully assigned typo-79c86f6b5f-56j6m to minikube
Normal	MountVolume		1m	kubelet, minikube	MountVolume.SetUp succeeded for volume "default-token-hpb9c"
Warning	Failed		15s (x4 over 1m)	kubelet, minikube	pulling image "james/typo:v1.0"
Warning	Failed		11s (x4 over 1m)	kubelet, minikube	Failed to pull image "james/typo:v1.0": rpc error: code = Unknown desc = Error response from daemon: repository james/typo not found: does not exist or no pull access
Warning	FailedSync			kubelet, minikube	Error syncing pod
Normal	BackOff			kubelet, minikube	Back-off pulling image "james/typo:v1.0"

Events section shows “**pull access denied**” via `kubectl describe pod`

Events:					
Type	Reason	Age	From	Message	
Normal	Scheduled	19s	default-scheduler	Successfully assigned zero-limit-74bc6899d4-9gkn4 to minikube	
Normal	SuccessfulMountVolume	19s	kubelet, minikube	MountVolume.SetUp succeeded for volume "default-token-r6qdp"	
Normal	Pulling	18s	kubelet, minikube	pulling image "jlptf/privateapp"	
Warning	Failed	14s	kubelet, minikube	Failed to pull image "jlptf/privateapp": rpc error: code = Unknown desc = Error response from daemon: pull access denied for jlptf/privateapp, repository does not exist or may require 'docker login'	
Warning	Failed	14s	kubelet, minikube	Error: ErrImagePull	
Normal	BackOff	14s	kubelet, minikube	Back-off pulling image "jlptf/privateapp"	
Warning	Failed	14s	kubelet, minikube	Error: ImagePullBackOff	


## Suggestions

- Typo correction, copying instead of typing (**be carefully**)
- Inexistent image is also considered as typo
- Inspect Pod via kubectl get/describe
- Check Events section
- Docker pull
- Repository authentication

	frequency	hard to fix
typo		
inexistent image		
without permission		



# Yaml

- Typo & Incorrect Format (IF)
  - list or map
- Correct Format  Correct Spec (CFIS)
- Correct Format, Correct Spec, Incorrect Version (CFCSIV)
  - deprecated group
    - Cronjob: batch/v2alpha1 (1.8) to batch/v1beta1 (1.9)

# CFIS Example

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: err-obj-nginx
spec:
  replicas: 3
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx
          ports:
            - containerPort: 80
```

- correct format
- incorrect spec because of *ports* position

You will receive “**unknown field**” after deploy

```
error: error validating "incorrect.yaml": error validating data:
ValidationError(Deployment.spec.template.spec): unknown field "ports" in io.k8s.api.core.v1.PodSpec; if you choose to ignore these
errors, turn validation off with --validate=false
```

## CFC Siv Example

```
apiVersion: v1
kind: Deployment
metadata:
  name: err-unknown-nginx
spec:
  replicas: 3
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx
        ports:
        - containerPort: 80
```

- correct format
- correct spec
- incorrect version because **v1** does not support ***Deployment*** object




You will receive “**unknown object type**” after deploy

```
error: error validating "incorrect.yaml": error validating data:
unknown object type schema.GroupVersionKind{Group:"", Version:"v1",
Kind:"Deployment"}; if you choose to ignore these errors, turn
validation off with --validate=false
```

## Suggestions

- Practice more
- Read document carefully
- Copy / Paste might not avoid

\* Object will not be deployed into k8s

	frequency	hard to fix
IF		
CFIS		
CFCSIV		

## Insufficient Resource



Hey k8s, deploy a Pod  
with 100 CPUs and 100T  
memory for me !



No problem, I will deploy the pod  
**when I have sufficient resources !**

You will find “**Pending**” status in both insufficient memory and insufficient cpu

NAME	READY	STATUS	RESTARTS	AGE
err-limit-nginx-546d7f8478-2v9nm	0/1	Pending	0	33s

Events section shows “**Insufficient memory**” via kubectl describe pod

```
Events:
  Type      Reason            Age          From          Message
  ----      -
Warning    FailedScheduling  8s (x5 over 15s)  default-scheduler  0/1 nodes are
available: 1 Insufficient memory.
```

Events section shows “**Insufficient cpu**” (via kubectl describe pod)

```
Events:
  Type      Reason            Age          From          Message
  ----      -
Warning    FailedScheduling  7s (x5 over 14s)  default-scheduler  0/1 nodes are
available: 1 Insufficient cpu.
```

# Insufficient Cluster Resource



Hey k8s, deploy a Pod  
with 1 CPUs and 1G  
memory for me !



No problem, I will deploy the pod  
**if cluster has more rooms !**



You can find “**Allocated resources**” section via `kubectl describe node`

```
Allocated resources:
```

```
 (Total limits may be over 100 percent, i.e., overcommitted.)
```

CPU Requests	CPU Limits	Memory Requests	Memory Limits
-----	-----	-----	-----
1999m (99%)	1884m (94%)	1900Mi (99%)	1900Mi (99%)

Can you deploy more pods ?

```

apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: zero-limit
spec:
  replicas: 1
  template:
    metadata:
      name: zero-limit
      labels:
        app: zero-limit
    spec:
      containers:
      - name: zero-limit
        image: nginx
        resources:
          requests:
            memory: 0
            cpu: 0
          limits:
            memory: 0
            cpu: 0

```

You can find “**Non-terminated Pod**” section via `kubectl describe node`

Non-terminated Pods: (8 in total)

Namespace	Name	CPU Requests	CPU Limits	Memory Requests	Memory Limits
default	zero-limit-64d8598b5-zbgwx	0 (0%)	0 (0%)	0 (0%)	0 (0%)

CPU Requests	CPU Limits	Memory Requests	Memory Limits
0 (0%)	0 (0%)	0 (0%)	0 (0%)

specify requests/limits to zero

# Resource Limitation



Hey k8s, deploy a Pod  
with 1 CPUs and 1G  
memory for me !



Resource  
Limitation

No problem, I will deploy the pod

**if requested resources is permitted !**

You can find DESIRED != the others via kubectl get deployment

NAME	DESIRED	CURRENT	UP-TO-DATE	AVAILABLE	AGE
quotas-nginx	2	1	1	1	1m

Find ReplicaSet via kubectl describe deployment

```
NewReplicaSet:      quotas-nginx-75f4785b7 (1/2 replicas created)
```


Events section shows “**exceeded quota**” (via kubectl describe replicaset)

Events:				
Type	Reason	Age	From	Message
Normal	SuccessfulCreate	2m	replicaset-controller	Created pod: quotas-nginx-75f4785b7-jlqpk
Warning	FailedCreate	1m (x16 over 2m)	replicaset-controller	Error creating: pods "quotas-nginx-75f4785b7-" is forbidden: <b>exceeded quota:</b> resource-quota, requested: pods=1, used: pods=1, limited: pods=1

# Suggestions

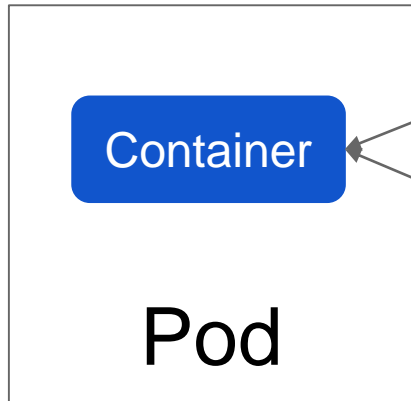
- Get more resources in (extend k8s cluster)
- Ask for less resources
- Modify resources limitation (by administrator)
- Remove unused pods
- kubectl get/describe
- Check LimitRange (cloud platform)

Name:	limits				
Namespace:	default				
Type	Resource	Min	Max	Default	Request
-----	-----	---	---	-----	-----
Container	cpu	-	-	100m	

	frequency	hard to fix
insufficient resource		
insufficient cluster resource		
resource limitation		

## ConfigMap / Secret not found

1. refer incorrect key



Mount as volume

ConfigMap

Secret

2. not exist

You will find “**CreateContainerConfigError**” status via kubectl get pod

NAME	READY	STATUS	RESTARTS	AGE
cm-pod	0/1	CreateContainerConfigError	0	6s

Events section shows “**Couldn't find key**” via kubectl describe pod

Events:	Type	Reason	Age	From	Message
	Normal	Scheduled	5m	default-scheduler	Successfully assigned cm-pod to minikube
	Normal	SuccessfulMountVolume	5m	kubelet, minikube	MountVolume.SetUp succeeded for volume "default-token-mgkxx"
	Normal	Pulled	3m (x8 over 5m)	kubelet, minikube	Successfully pulled image "gcr.io/google_containers/busybox"
	Warning	Failed	3m (x8 over 5m)	kubelet, minikube	Error: Couldn't find key abc in ConfigMap default/myconf
	Normal	Pulling	18s (x24 over 5m)	kubelet, minikube	pulling image "gcr.io/google_containers/busybox"



# Example

Events section shows “**configmaps not found**” via `kubectl describe`

```
Events:
  Type    Reason          Age    From                      Message
  ----    -
  Normal  Scheduled       13s    default-scheduler        Successfully assigned cm-pod to minikube
  Normal  SuccessfulMountVolume 13s    kubelet, minikube        MountVolume.SetUp succeeded for volume "default-token-mgkxx"
  Normal  Pulling         11s (x2 over 12s)    kubelet, minikube        pulling image "gcr.io/google_containers/busybox"
  Normal  Pulled          10s (x2 over 11s)    kubelet, minikube        Successfully pulled image "gcr.io/google_containers/busybox"
  Warning Failed          10s (x2 over 11s)    kubelet, minikube        Error: configmaps "myconf" not found
```

## Suggestions

- Typo correction
- Deploy missing ConfigMap / Secret
- `kubectl get/describe`

	frequency	hard to fix
ConfigMap	★	★
Secret	★	★

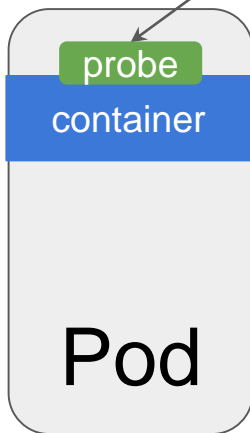
# Probe



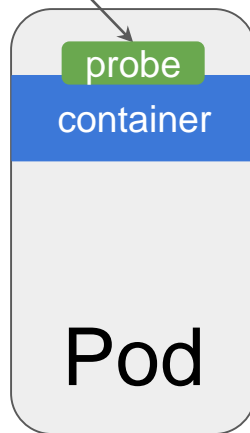
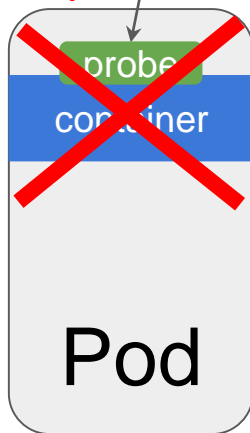
## container alive ?

HTTP 200 ~ 400 OK

App Crash



Respond Late



...

Typo



## Example

You will find “**CrashLoopBackOff**” and “**RESTARTS**” status via `kubectl get pod`

NAME	READY	STATUS	RESTARTS	AGE
err-probe-nginx	0/1	CrashLoopBackOff	3	56s

```
Events:
  Type    Reason                  Age    From                      Message
  ----    -
  Normal  Scheduled               1m     default-scheduler        Successfully assigned err-probe-nginx to minikube
  Normal  SuccessfulMountVolume  1m     kubelet, minikube        MountVolume.SetUp succeeded for volume "default-token-r6gdp"
  Warning Unhealthy               1m (x4 over 1m) kubelet, minikube        Readiness probe failed: HTTP probe failed with statuscode: 404
  Normal  Pulling                1m (x5 over 1m) kubelet, minikube        pulling image nginx
  Normal  Killing                1m (x2 over 1m) kubelet, minikube        Killing container with id docker://nginx:Container failed liveness probe.. Container will be killed and recreated.
  Normal  Pulled                 1m (x3 over 1m) kubelet, minikube        Successfully pulled image "nginx"
  Normal  Created                1m (x3 over 1m) kubelet, minikube        Created container
  Normal  Started                1m (x3 over 1m) kubelet, minikube        Started container
  Warning Unhealthy               1m (x6 over 1m) kubelet, minikube        Liveness probe failed: HTTP probe failed with statuscode: 404
```

Events section shows “**Readiness probe failed**” and “**Liveness probe failed**” via `kubectl describe pod`




# Example

Log shows “**GET /healthy HTTP/1.1 404**” via kubectl logs pod

```
172.17.0.1 - - [15/Mar/2018:03:43:19 +0000] "GET /healthy HTTP/1.1" 404 169 "-" "kube-probe/
." "-"
2018/03/15 03:43:19 [error] 6#6: *1 open() "/usr/share/nginx/html/healthy" failed (2: No suc
h file or directory), client: 172.17.0.1, server: localhost, request: "GET /healthy HTTP/1.1
", host: "172.17.0.7:80"
2018/03/15 03:43:20 [error] 6#6: *2 open() "/usr/share/nginx/html/healthy" failed (2: No suc
h file or directory), client: 172.17.0.1, server: localhost, request: "GET /healthy HTTP/1.1
", host: "172.17.0.7:80"
172.17.0.1 - - [15/Mar/2018:03:43:20 +0000] "GET /healthy HTTP/1.1" 404 169 "-" "kube-probe/
." "-"
```

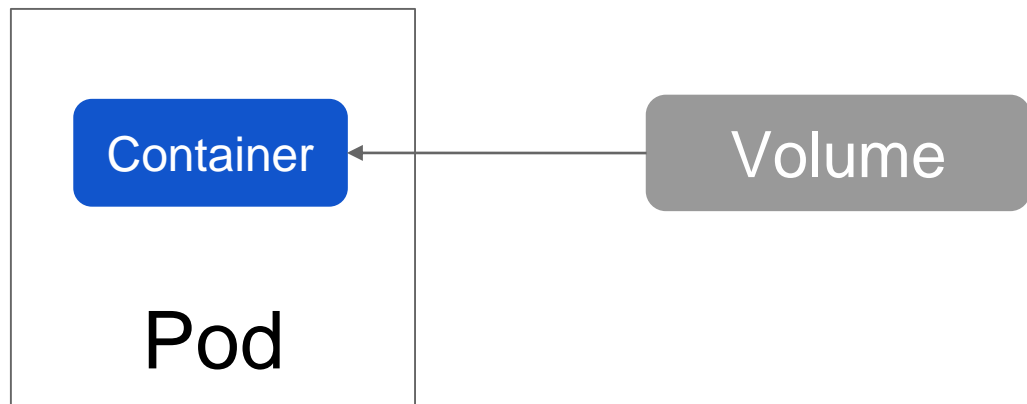
## Suggestions

- Typo correction
- Provide proper probe
  - probe might need to be postponed
- Check response code
- `kubectl get/describe`
- `kubectl logs`

	frequency	hard to fix
typo		
respond late		
app crash		



# Volume



## Example

You will find “**ContainerCreating**” status via `kubectl get pod`

NAME	READY	STATUS	RESTARTS	AGE
err-unmount-nginx	0/1	ContainerCreating	0	1m

Events section shows “**mount failed**” via `kubectl describe pod`

```
MountVolume.SetUp failed for volume "test" : mount failed: exit status 32
```

# Suggestions

- Typo correction
- Deploy missing Volume
- `kubectl get/describe`

# Conclusion

- Utilize kubectl get/describe
- Check event section
- Utilize kubectl logs