

SHANGHAI JIAOTONG UNIVERSITY

CLOUD COMPUTING

---

# Report on Hello-Kubernetes

---

*Author:*

LUO Dihao  
118260910039

*Supervisor:*

He RUAN

June 9, 2019



# 1 Pod

## 1.1 1 Pod with 1 Container

We can see after creating pod1 by pod1.yaml, we can execute any command by kubectl exec -it pod1 – command.

```
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/pod ➤ kubectl create -f pod1.yaml
pod/pod1 created
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/pod ➤ kubectl exec -it pod1 -- env
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=pod1
TERM=xterm
KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_PORT=tcp://10.96.0.1:443
KUBERNETES_PORT_443_TCP=tcp://10.96.0.1:443
KUBERNETES_PORT_443_TCP_PROTO=tcp
KUBERNETES_PORT_443_TCP_PORT=443
KUBERNETES_PORT_443_TCP_ADDR=10.96.0.1
KUBERNETES_SERVICE_HOST=10.96.0.1
KUBERNETES_SERVICE_PORT=443
HOME=/root
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/pod ➤ kubectl exec -it pod1 -- /bin/sh
/ # ls
bin dev etc home proc root sys tmp usr var
/ # cd etc/network/if-
if-down.d if-post-down.d if-pre-up.d if-up.d
/ # cd home/
/home # ls
/home # exit
```

Figure 1: 1 Pod with 1 Container

## 1.2 1 Pod with 2 Containers

We can see after we change index.html in container ct-debian, we can also see the change in container ct-nginx.

```
luodiha@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/pod ➤ kubectl create -f pod2.yaml
pod/pod2 created
luodiha@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/pod ➤ kubectl exec -it pod2 -c ct-nginx -- /bin/bash
error: unable to upgrade connection: container not found ("ct-nginx")
x luodiha@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-mas
ter/pod ➤ kubectl exec -it pod2 -c ct-nginx -- /bin/bash
root@pod2:/# apt-get update
Get:1 http://security-cdn.debian.org/debian-security stretch/updates InRelease [94.3
kB]
Ign:2 http://cdn-fastly.deb.debian.org/debian stretch InRelease
```

Figure 2: 1 Pod with 2 Container

```
Get:8 http://cdn-fastly.deb.debian.org/debian stretch/main amd64 Packages [7082 kB]
Fetched 7907 kB in 1min 32s (85.7 kB/s)
Reading package lists... Done
root@pod2:/# apt-get install curl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ca-certificates krb5-locales libcurl3 libffi16 libgmp10 libgnutls30
  libgssapi-krb5-2 libhogweed4 libidn11 libidn2-0 libk5crypto3 libkeyutils1
```

Figure 3: 1 Pod with 2 Container

```
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
root@pod2:/# curl localhost
Hello from the debian container
root@pod2:/# exit
exit
luodiha@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/pod ➤ kubectl describe pod pod2
Name:           pod2
Namespace:      default
Priority:       0
PriorityClassName: <none>
Node:           minikube/10.0.2.15
Start Time:     Thu, 06 Jun 2019 16:58:27 +0800
Labels:          <none>
Annotations:    <none>
Status:         Running
IP:             172.17.0.6
Containers:
  ct-nginx
```

Figure 4: 1 Pod with 2 Container

```

Normal  Pulling   4m    kubelet, minikube  Pulling image "debian"
Normal  Pulled    3m53s kubelet, minikube  Successfully pulled image "debian"
Normal  Created   3m53s kubelet, minikube  Created container ct-debian
Normal  Started   3m53s kubelet, minikube  Started container ct-debian
luodihao@luodeMacBook-Pro ~Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/pod ➤ kubectl exec -it pod2 -c ct-debian -- /bin/bash
root@pod2:/# echo Change message from pod2-ct-debian > /data/index.html
root@pod2:/# exit
exit
luodihao@luodeMacBook-Pro ~Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/pod ➤ curl 172.17.0.6
^C
x luodihao@luodeMacBook-Pro ~Desktop/2018-2019 春/CC tencent/report/hello-k8s-mas
ter/pod ➤ kubectl exec -it pod2 -c ct-nginx -- /bin/bash
root@pod2:/# curl localhost
Change message from pod2-ct-debian
root@pod2:/# 

```

Figure 5: 1 Pod with 2 Container

### 1.3 Pod with resource limitation

We can see the status of memory-demo-2 is immediately CrashLoopBack-Off because of resource limitation.

```

root@pod2:/# exit
exit
luodihao@luodeMacBook-Pro ~Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/pod ➤ kubectl apply -f pod3.yaml
pod/memory-demo-2 created
luodihao@luodeMacBook-Pro ~Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/pod ➤ kubectl get pods
NAME        READY  STATUS      RESTARTS  AGE
memory-demo-2  0/1   CrashLoopBackOff  1          10s
pod1        1/1   Running     0          21m
pod2        2/2   Running     0          9m11s
luodihao@luodeMacBook-Pro ~Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/pod ➤ 

```

Figure 6: Pod with resource limitation

Then I deleted all the pods.

```

memory-demo-2  0/1    CrashLoopBackOff  1           10s
pod1          1/1    Running     0            21m
pod2          2/2    Running     0            9m11s
luodihao@luodeMacBook-Pro ~Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/pod ➤ kubectl delete pod pod1
pod "pod1" deleted
luodihao@luodeMacBook-Pro ~Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/pod ➤ kubectl delete pod pod2 && kubectl delete pod memory-demo-2
pod "pod2" deleted
pod "memory-demo-2" deleted
luodihao@luodeMacBook-Pro ~Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/pod ➤ kubectl get pods
No resources found.

```

Figure 7: Pod with resource limitation

## 1.4 Pod Liveness CMD Check

By using the command kubectl get pods, we can see liveness-exec has restarted multiple times.

```

x luodihao@luodeMacBook-Pro ~Desktop/2018-2019 春
ter/pod ➤ kubectl apply -f pod-liveness-cmd.yaml
pod/liveness-exec created
luodihao@luodeMacBook-Pro ~Desktop/2018-2019 春/
r/pod ➤ kubectl get pods
NAME        READY   STATUS    RESTARTS   AGE
liveness-exec 1/1     Running   0          10s

```

Figure 8: Pod Liveness CMD Check

```

luodihao@luodeMacBook-Pro ~Desktop/2018-2019 春/CC tence
r/pod ➤ kubectl get pods
NAME        READY   STATUS    RESTARTS   AGE
liveness-exec 1/1     Running   2          2m55s

```

Figure 9: Pod Liveness CMD Check

## 1.5 Pod Liveness HTTP Check

We can see through command that pod has restarted. But also we can get the ip of pod and then check its status through HTTP by using curl.

```
r/pod ➤ kubectl apply -f pod-liveness-http.yaml
pod/liveness-http created
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC
r/pod ➤ kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
liveness-exec  1/1     Running   6          9m17s
liveness-http  1/1     Running   0          18s
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC
r/pod ➤ kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
liveness-exec  1/1     Running   6          9m20s
liveness-http  1/1     Running   1          21s
```

Figure 10: Pod Liveness HTTP Check

```
ter/pod ➤ kubectl describe pod liveness-http
Name:           liveness-http
Namespace:      default
Priority:       0
PriorityClassName: <none>
Node:           minikube/10.0.2.15
Start Time:     Sat, 08 Jun 2019 10:03:34 +0800
Labels:         test=liveness
Annotations:    kubectl.kubernetes.io/last-applied-configuration={"apiVersion":"v1","kind":"Pod","metadata":{"labels":{"test":"liveness"},"name":"liveness-http","namespace":"default"}},kubernetes.io/pod-updated-at=2019-06-08T10:03:34Z
Status:         Running
IP:             172.17.0.6
```

Figure 11: Pod Liveness HTTP Check

```
$ curl 172.17.0.6:8080/healthz
error: 12.700569145$ curl 172.17.0.6:8080/healthz
error: 15.231275604$ curl 172.17.0.6:8080/healthz
error: 16.20401937$ █
```

Figure 12: Pod Liveness HTTP Check

## 1.6 NodeSelector

Since Minikube only has one node, we cannot do this section.

```
luodihao@luodeMacBook-Pro ~ ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-master
r/pod ➔ kubectl get node
NAME      STATUS   ROLES    AGE     VERSION
minikube  Ready    master   2d22h   v1.14.2
luodihao@luodeMacBook-Pro ~ ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-master
r/pod ➔ kubectl get nodes minikube --show-labels
NAME      STATUS   ROLES    AGE     VERSION   LABELS
minikube  Ready    master   2d22h   v1.14.2   beta.kubernetes.io/arch=amd64,beta.kub
ernetes.io/os=linux,kubernetes.io/arch=amd64,kubernetes.io/hostname=minikube,kubern
etes.io/os=linux,node-role.kubernetes.io/master=
```

Figure 13: NodeSelector

## 1.7 InitContainer

We can see InitContainer executes before container, so we can see it creates testfile then xxx.

```
luodihao@luodeMacBook-Pro ~ ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-master
r/pod ➔ kubectl apply -f pod-initcontainer.yaml
pod/myapp-pod created
luodihao@luodeMacBook-Pro ~ ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-master
r/pod ➔ kubectl get pod
NAME        READY   STATUS      RESTARTS   AGE
liveness-exec  0/1    CrashLoopBackOff  9          21m
liveness-http  1/1    Running     0          12m
myapp-pod     0/1    Init:0/1    0          3s
```

Figure 14: InitContainer

```
r/pod ➔ kubectl exec -it myapp-pod -- ls /storage/
testfile  xxx
```

Figure 15: InitContainer

## 2 Deployment

### 2.1 deployment by CMD

We can see we have created a deployment by command, and it creates 2 pods.

```

luodiha@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/deployment ➔ kubectl run nginx-app --image=nginx:1.9.0 --image-pull-policy=IfNotPre
sent --replicas=2
kubectl run --generator=deployment/apps.v1 is DEPRECATED and will be removed in a fut
ure version. Use kubectl run --generator=run-pod/v1 or kubectl create instead.
deployment.apps/nginx-app created
luodiha@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/deployment ➔ kubectl get deployments
NAME READY UP-TO-DATE AVAILABLE AGE
nginx-app 0/2 2 0 5s
luodiha@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/deployment ➔ kubectl get deployments
NAME READY UP-TO-DATE AVAILABLE AGE
nginx-app 0/2 2 0 16s
luodiha@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/deployment ➔ kubectl get pod
NAME READY STATUS RESTARTS AGE
nginx-app-575df867d8-qxw8s 0/1 ContainerCreating 0 22s
nginx-app-575df867d8-w8fpt 0/1 ContainerCreating 0 22s
luodiha@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/deployment ➔ kubectl get deployments
NAME READY UP-TO-DATE AVAILABLE AGE
nginx-app 2/2 2 2 32s
luodiha@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/deployment ➔ kubectl get pod
NAME READY STATUS RESTARTS AGE
nginx-app-575df867d8-qxw8s 1/1 Running 0 34s
nginx-app-575df867d8-w8fpt 1/1 Running 0 34s

```

Figure 16: InitCoDeploymentntainer

## 2.2 Deployment & Scale & Upgrade

We can see we have created one deployment and 1 pod in it.

```

luodiha@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/deployment ➔ kubectl create -f deployment.yaml
deployment.apps/nginx-deployment created
luodiha@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/deployment ➔ kubectl get deployment
NAME READY UP-TO-DATE AVAILABLE AGE
nginx-deployment 1/1 1 1 5s
luodiha@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/deployment ➔ kubectl get pods -o wide
NAME READY STATUS RESTARTS AGE IP NODE NOMINATED-NODE READINESS GATES
nginx-deployment-547b877857-kwm7w 1/1 Running 0 12s 172.17.0.5 minikube <none> <none>

```

Figure 17: Deployment &amp; Scale &amp; Upgrade

Then I checked nginx is working in Minikube.

```
$ curl 172.17.0.5
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
}
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
$
```

Figure 18: Deployment &amp; Scale &amp; Upgrade

I scaled the pods by 3, so now there are 3 pods in one deployment.

```
[root@deployment ~]# kubectl scale deployment nginx-deployment --replicas=3
deployment.extensions/nginx-deployment scaled
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
[root@deployment ~]# kubectl get deployment
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   3/3     3           3           77s
[luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
[root@deployment ~]# kubectl get pods -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP          NODE   NOMINATED NODE   READINESS GATES
nginx-deployment-547b877857-8sbzf   1/1     Running   0      16s   172.17.0.7   m
nikube        <none>  <none>
nginx-deployment-547b877857-kwm7w   1/1     Running   0      83s   172.17.0.5   m
nikube        <none>  <none>
nginx-deployment-547b877857-qrlvp   1/1     Running   0      16s   172.17.0.6   m
nikube        <none>  <none>
```

Figure 19: Deployment &amp; Scale &amp; Upgrade

When we upgrade the deployment, the pods automatically upgrade by terminating and creating.

```
r/deployment> kubectl set image deployment nginx-deployment nginx=nginx:latest --record
deployment.extensions/nginx-deployment image updated
luodiha@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-master
r/deployment> kubectl get pods -o wide
NAME                               READY   STATUS    RESTARTS   AGE     IP
      NODE   NOMINATED-NODE   READINESS GATES
nginx-deployment-547b877857-8sbfz  0/1    Terminating   0        35s   172.17.0.
7  minikube <none>           <none>
nginx-deployment-547b877857-kwm7w  0/1    Terminating   0        102s  172.17.0.
5  minikube <none>           <none>
nginx-deployment-547b877857-qrlvp  0/1    Terminating   0        35s   172.17.0.
6  minikube <none>           <none>
nginx-deployment-9f5984f5b-5wvxd  1/1    Running     0        4s    172.17.0.
6  minikube <none>           <none>
nginx-deployment-9f5984f5b-vhkhk  1/1    Running     0        5s    172.17.0.
8  minikube <none>           <none>
nginx-deployment-9f5984f5b-vnkgr  1/1    Running     0        3s    172.17.0.
7  minikube <none>           <none>
```

Figure 20: Deployment &amp; Scale &amp; Upgrade

We can also rolling back to its previous version. Now there are 3 pods in its previous version.

```

luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/deployment kubectl rollout history deployment nginx-deployment
deployment.extensions/nginx-deployment
REVISION CHANGE-CAUSE
1 <none>
2 kubectl set image deployment nginx-deployment nginx=nginx:latest --record=true

luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/deployment kubectl rollout history deployment nginx-deployment --revision=1
deployment.extensions/nginx-deployment with revision #1
Pod Template:
  Labels:      app=nginx
                pod-template-hash=547b877857
  Containers:
    nginx:
      Image:      nginx:1.9.0
      Port:       80/TCP
      Host Port:  0/TCP
      Environment: <none>
      Mounts:     <none>
      Volumes:    <none>

luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/deployment kubectl rollout undo deployment nginx-deployment --to-revision=1
deployment.extensions/nginx-deployment rolled back
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/deployment kubectl get pods -o wide
NAME          READY   STATUS    RESTARTS   AGE     IP
  NODE        NOMINATED-NODE  READINESS GATES
nginx-deployment-547b877857-9zblx  1/1     Running   0        4s    172.17.0.6
  minikube  <none>           <none>
nginx-deployment-547b877857-n5pxs  1/1     Running   0        5s    172.17.0.7
  minikube  <none>           <none>
nginx-deployment-547b877857-nffpt  1/1     Running   0        7s    172.17.0.5
  minikube  <none>           <none>
nginx-deployment-9f5984f5b-5wvxd  0/1     Terminating   0        55s   172.17.0.6
  minikube  <none>           <none>
nginx-deployment-9f5984f5b-vhkhk  0/1     Terminating   0        56s   172.17.0.8
  minikube  <none>           <none>
nginx-deployment-9f5984f5b-vnkgr  0/1     Terminating   0        54s   172.17.0.7
  minikube  <none>           <none>

```

Figure 21: Deployment &amp; Scale &amp; Upgrade

### 2.3 Auto-scale

We create a deployment and set autoscale.

```
ter/deployment ➤ kubectl delete deployment nginx-deployment
deployment.extensions "nginx-deployment" deleted
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/deployment ➤ kubectl run hpa-test --image=radial/busyboxplus --replicas=1 --image-p
ull-policy=IfNotPresent --command -- sleep 10000
kubectl run --generator=deployment/apps.v1 is DEPRECATED and will be removed in a fut
ure version. Use kubectl run --generator=run-pod/v1 or kubectl create instead.
deployment.apps/hpa-test created
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/deployment ➤ kubectl autoscale deployments hpa-test --min=3 --max=5 --cpu-percent=1
0
horizontalpodautoscaler.autoscaling/hpa-test autoscaled
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/deployment ➤ kubectl get pods
NAME READY STATUS RESTARTS AGE
hpa-test-78c7984749-gxlf5 1/1 Running 0 28s
```

Figure 22: Auto-scale

Then we do a loopless job, so it should automatically increase the number of pods.

```
* luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/r
ter/deployment ➤ kubectl exec -it hpa-test-78c7984749-gxlf5 sh
/ # while true;do touch 123;done
```

Figure 23: Auto-scale

The number of pods increase from 1 to 3.

NAME	READY	STATUS	RESTARTS	AGE
hpa-test-78c7984749-4t9qp	1/1	Running	0	69s
hpa-test-78c7984749-6qqqf	1/1	Running	0	69s
hpa-test-78c7984749-gxlf5	1/1	Running	0	105s

  

NAME	READY	STATUS	RESTARTS	AGE
hpa-test-78c7984749-4t9qp	1/1	Running	0	71s
hpa-test-78c7984749-6qqqf	1/1	Running	0	71s
hpa-test-78c7984749-gxlf5	1/1	Running	0	107s

Figure 24: Auto-scale

## 2.4 DaemonSet

When we set a daemonset, if the node meets its requirement, it will automatically creates the pod.

```
r/deployment ➤ kubectl apply -f daemonset.yaml
daemonset.apps/nginx-daemonset created
luodihao@luodeMacBook-Pro ➤ ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/deployment ➤ kubectl get pods -o wide
NAME           READY   STATUS    RESTARTS   AGE     IP          NODE   NO
MINDED NODE   READINESS GATES
nginx-daemonset-hdqdv  1/1     Running   0          4s    172.17.0.5  minikube <n
one>           <none>
```

Figure 25: DaemonSet

## 2.5 Simple Job

We can see it creates a job which includes 1 pod and prints pi.

```
r/deployment ➤ kubectl apply -f job.yaml
job.batch/pi created
luodihao@luodeMacBook-Pro ➤ ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/deployment ➤ kubectl get job
NAME  COMPLETIONS DURATION  AGE
pi    0/1          11s       11s
luodihao@luodeMacBook-Pro ➤ ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/deployment ➤ kubectl logs pi
Error from server (NotFound): pods "pi" not found
x luodihao@luodeMacBook-Pro ➤ ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-mas
ter/deployment ➤ kubectl get pod
NAME           READY   STATUS    RESTARTS   AGE
nginx-daemonset-hdqdv  1/1     Running   0          5m41s
pi-69zg8        0/1     ContainerCreating  0          28s
luodihao@luodeMacBook-Pro ➤ ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/deployment ➤ kubectl logs pi-69zg8
Error from server (BadRequest): container "pi" in pod "pi-69zg8" is waiting to start:
ContainerCreating
```

Figure 26: Simple Job

```
luodihao@luodeMacBook-Pro ~ ~/Desktop/2018-2019 春/CC_tencent/report/hello-k8s-master
r/deployment ➤ kubectl get job
NAME   COMPLETIONS   DURATION   AGE
pi     1/1           73s        89s
luodihao@luodeMacBook-Pro ➤ ~/Desktop/2018-2019 春/CC_tencent/report/hello-k8s-master
r/deployment ➤ kubectl logs pi-69zg8
3.141592653589793238462643338327950288419716939937510582097494459230781640628620899862
8034825342117067982148086513282306647093844609550582231725359408128481117450284102701
93852110555964462948954930381964428810975665933461284756482337867831652712019091456
485669234603486104543266480726024914127372458700660631558817488152092096282925
4091715364367892590360011330530548820466521384146951941511609433057270365759591953092
1861173819326117931051185480744623799627495673518857527248912279381830119491298336733
6244065664308602139494639522473719070217986094370277053921717629317675238467481846766
9405132000568127145263560827785771342757789609173637178721468440901224953430146549585
3710507922796892589235420199561121290219608640344181598136297747713099605187072113499
99983729780499510597317328160963185950244594553469083026452230825334468503526193118
8171010003137838752886587533208381420617177669147303598253490428755468731159562863882
3537875937519577818577805321712268066130019278766111959092164201989380952572010654858
632788659361533818279682303019520353018529689957736259941389124972177528347913151557
4857242454150695950829533116861727855889075098381754637464939319255060400927701671139
0098488240128583616035637076601047101819429555961989467678374494482553797747268471040
475346462080466842590694912933136770289815210475216205696602405803815019351125338243
003558764024749647326391419927260426992796782354781636009341721641219924586315030286
182974555706749838505494588586926995690927210797509302953211653449872027559602364806
654991198818347977535663698074265425278625518184175746728909777279380008164706001614
5249192173217214772350141441973568548161361157352552133475741849468438523323907394143
334547762416862518983569485620992192221842725502542568876717904946016534668049886272
327917860857843838279679766814541009538378636095068006422512520511739298489608412848
862694560424196528502210661186306744278622039194945047123713786960956364371917287467
7646575739624138908658326459958133904780275901
```

Figure 27: Simple Job

## 2.6 Completion Job

We can see it creates a job which includes 10 pods and it creates gradually.

```
r/deployment ➤ kubectl apply -f job-completion.yaml
job.batch/busyboxplus created
luodiha@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/deployment ➤ kubectl get pod
NAME READY STATUS RESTARTS AGE
busyboxplus-452tr 0/1 Completed 0 2s
busyboxplus-bvw6l 0/1 Completed 0 5s
busyboxplus-jl7nn 0/1 ContainerCreating 0 1s
busyboxplus-lktf7 0/1 Completed 0 3s
pi-69zg8 0/1 Completed 0 6m32s
luodiha@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/deployment ➤ kubectl get pod
NAME READY STATUS RESTARTS AGE
busyboxplus-452tr 0/1 Completed 0 3s
busyboxplus-bvw6l 0/1 Completed 0 6s
busyboxplus-jl7nn 0/1 Completed 0 2s
busyboxplus-lktf7 0/1 Completed 0 4s
busyboxplus-t9v52 0/1 ContainerCreating 0 0s
busyboxplus-t9v52 0/1 Completed 0 2s
pi-69zg8 0/1 Completed 0 6m33s
luodiha@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/deployment ➤ kubectl get pod
NAME READY STATUS RESTARTS AGE
busyboxplus-452tr 0/1 Completed 0 5s
busyboxplus-bvw6l 0/1 Completed 0 8s
busyboxplus-cn2fh 0/1 ContainerCreating 0 0s
busyboxplus-ct6ph 0/1 Completed 0 1s
busyboxplus-jl7nn 0/1 Completed 0 4s
busyboxplus-lktf7 0/1 Completed 0 6s
busyboxplus-t9v52 0/1 Completed 0 2s
pi-69zg8 0/1 Completed 0 6m35s
```

Figure 28: Completion Job

```
r/deployment ➤ kubectl get job.batch
NAME COMPLETIONS DURATION AGE
busyboxplus 10/10 12s 24s
pi 1/1 73s 6m51s
```

Figure 29: Completion Job

## 2.7 Cronjob

In this part, we create a cronjob with schedule `*/1 * * * *`, which means 1 time per minute.

```
r/deployment ➤ kubectl apply -f cronjob.yaml
cronjob.batch/hello created
luodiha@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/repo
r/deployment ➤ kubectl get cronjobs.batch
NAME SCHEDULE SUSPEND ACTIVE LAST SCHEDULE AGE
hello */1 * * * * False 0 <none> 4s
```

Figure 30: Cronjob

So after 2 minutes, we can see 2 pods has been created.

NAME	SCHEDULE	SUSPEND	ACTIVE	LAST SCHEDULE	AGE
hello	*/1 * * * *	False	0	60s	79s

Figure 31: Cronjob

NAME	READY	STATUS	RESTARTS	AGE
hello-1559963760-6jq8h	0/1	Completed	0	73s
hello-1559963820-2j4pq	0/1	Completed	0	13s

  

luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春 /CC tencent/r/deployment ➔ kubectl logs hello-1559963760-6jq8h
Sat Jun 8 03:16:07 UTC 2019
Hello from the Kubernetes cluster

  

luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春 /CC tencent/r/deployment ➔ kubectl logs hello-1559963820-2j4pq
Sat Jun 8 03:17:07 UTC 2019

Figure 32: Cronjob

## 2.8 Static Pod

I have created a file named pod-static.yaml in Minikube:/etc/kubernetes/manifests/. Then it automatically creates a static pod. Even I try to delete it, it creates again.

\$ cd /etc/kubernetes/manifests/
\$ ls
addon-manager.yaml.tmpl    kube-apiserver.yaml                kube-scheduler.yaml
etcd.yaml                    kube-controller-manager.yaml
\$ vi pod-static.yaml
\$ ls
addon-manager.yaml.tmpl    kube-apiserver.yaml                kube-scheduler.yaml
etcd.yaml                    kube-controller-manager.yaml
\$ sudo vi pod-static.yaml
\$ ls
addon-manager.yaml.tmpl    kube-apiserver.yaml                kube-scheduler.yaml
etcd.yaml                    kube-controller-manager.yaml    pod-static.yaml

Figure 33: Static Pod

```
r/deployment ➤ kubectl get pod
NAME             READY   STATUS    RESTARTS   AGE
hello-1559964000-t2g46  0/1     Completed  0          2m52s
hello-1559964060-7fgb4  0/1     Completed  0          112s
hello-1559964120-l495w  0/1     Completed  0          52s
static-web-minikube    1/1     Running   0          5s
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/deployment ➤ kubectl delete pod static-web-minikube
pod "static-web-minikube" deleted
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/deployment ➤ kubectl get pod
NAME             READY   STATUS    RESTARTS   AGE
hello-1559964000-t2g46  0/1     Completed  0          3m10s
hello-1559964060-7fgb4  0/1     Completed  0          2m10s
hello-1559964120-l495w  0/1     Completed  0          70s
hello-1559964180-sscm2  0/1     Completed  0          9s
static-web-minikube    0/1     Pending   0          3s
```

Figure 34: Static Pod

### 3 Service

#### 3.1 NodePort

We've created a node port and successfully access it.

```
r/service ➤ kubectl apply -f service1.yaml
deployment.apps/nginx-deployment created
service/svc1 created
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/service ➤ kubectl get svc
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
kubernetes  ClusterIP  10.96.0.1      <none>        443/TCP      2d23h
svc1      NodePort  10.111.81.115  <none>        8888:30888/TCP  22s
```

Figure 35: NodePort

```
$ curl 10.111.81.115:8888
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
}
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
```

Figure 36: NodePort

## 3.2 ClusterIP

We've created a cluster port and successfully access it.

```
r/service ➤ kubectl create -f service2.yaml
service/svc2 created
Error from server (AlreadyExists): error when creating "service2.yaml": deployment
ps "nginx-deployment" already exists
x luodihao@LuodeMacBook-Pro ➤ ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s
ter/service ➤ kubectl get svc
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
kubernetes  ClusterIP  10.96.0.1      <none>        443/TCP      2d23h
svc1       NodePort   10.111.81.115  <none>        8888:30888/TCP  10m
svc2       ClusterIP  10.105.119.249  <none>        8889/TCP      9s
```

Figure 37: ClusterIP

```
$ curl 10.105.119.249:8889
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
```

Figure 38: ClusterIP

### 3.3 Service for Pod

We have exposed a service through ClusterIP.

```
ter/service ➤ kubectl apply -f service-pod.yaml
pod/test created
```

Figure 39: Service for Pod

```
x luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-mas
ter/service ➤ kubectl expose pod test --type=ClusterIP --port=80 --target-port=80
service/test exposed
```

Figure 40: Service for Pod

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	2d23h
svc1	NodePort	10.111.81.115	<none>	8888:30888/TCP	14m
svc2	ClusterIP	10.105.119.249	<none>	8889/TCP	4m56s
test	ClusterIP	10.98.157.148	<none>	80/TCP	22s

Figure 41: Service for Pod

```
$ curl 10.98.157.148:80
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
}
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org/">http://nginx.org/</p>
</body>

```

Figure 42: Service for Pod

### 3.4 Health Check

Without health check, this service doesn't work.

```
r/service ➤ kubectl apply -f service-health-check.yaml
deployment.apps/service-health created
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/service ➤ kubectl expose deployment service-health
service/service-health exposed
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/service ➤ kubectl get service
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
kubernetes   ClusterIP  10.96.0.1      <none>        443/TCP      2d23h
service-health ClusterIP  10.109.203.127  <none>        8080/TCP      5s
svc1       NodePort   10.111.81.115  <none>        8888:30888/TCP  22m
svc2       ClusterIP  10.105.119.249  <none>        8889/TCP      12m
test       ClusterIP  10.98.157.148  <none>        80/TCP       8m2s
```

Figure 43: Health Check

```
$ curl 10.109.203.127:8080
curl: (7) Failed to connect to 10.109.203.127 port 8080: Connection refused
$ curl 10.109.203.127:8080
curl: (7) Failed to connect to 10.109.203.127 port 8080: Connection refused
$
```

Figure 44: Health Check

## 4 Storage

### 4.1 Pod-level

We can see with hostPath, the file is connected with pod and host. When we create a file xxx in pod, it is automatically showed in Minikube:/tmp. And with emptyDir, we cannot connect those two.

```
luodihao@luodeMacBook-Pro ~ /Desktop/2018-2019 春/CC tencent/report
r/storage ➤ kubectl create -f pod-vol-hostpath.yaml
pod/storage-vol-hostpath created
luodihao@luodeMacBook-Pro ~ /Desktop/2018-2019 春/CC tencent/report
r/storage ➤ kubectl exec -it storage-vol-hostpath -- /bin/bash
```

Figure 45: Pod-level

```
root@storage-vol-hostpath:/# cd /data/
root@storage-vol-hostpath:/data# ls
crio.INFO
crio.WARNING
crio.minikube.root.log.INFO.20190608-040304.2268
crio.minikube.root.log.INFO.20190608-040305.2404
crio.minikube.root.log.WARNING.20190608-040304.2268
crio.minikube.root.log.WARNING.20190608-040305.2404
hostpath-provisioner
hostpath_pv
storage-provisioner.INFO
storage-provisioner.minikube.unknownuser.log.INFO.20190608-040304.2268
systemd-private-ac867ff60cd74780afb31e4a5434610c-systemd-resolv
root@storage-vol-hostpath:/data# touch xxx
root@storage-vol-hostpath:/data# █
```

Figure 46: Pod-level

```
$ ls /tmp
crio.INFO
crio.WARNING
crio.minikube.root.log.INFO.20190608-040304.2268
crio.minikube.root.log.INFO.20190608-040305.2404
crio.minikube.root.log.WARNING.20190608-040304.2268
crio.minikube.root.log.WARNING.20190608-040305.2404
hostpath-provisioner
hostpath_pv
storage-provisioner.INFO
storage-provisioner.minikube.unknownuser.log.INFO.20190608-040333.1
systemd-private-ac867ff60cd74780afb31e4a5434610c-systemd-resolved.service-DcaVan
xxx
```

Figure 47: Pod-level

## 4.2 Persistent Volume

```
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/storage ➔ kubectl apply -f pv.yaml
persistentvolume/task-pv-volume created
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/storage ➔ kubectl get pv
NAME          CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS     CLAIM      STORA
GECLASS      REASON     AGE
task-pv-volume  10Gi       RWO           Retain        Available   ldiha
o             5s
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/storage ➔ kubectl apply -f pvc.yaml
persistentvolumeclaim/storage-sc created
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/storage ➔ kubectl get pvc
NAME          STATUS      VOLUME      CAPACITY   ACCESS MODES   STORAGECLASS   AGE
storage-sc   Pending    task-pv-volume  10Gi       RWO           manual        3s
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/storage ➔ kubectl apply -f pod-pvc.yaml
pod/storage-pvc-sc created
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/storage ➔ kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
storage-pvc-sc  0/1    Pending   0          3s
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/storage ➔ █
```

Figure 48: Persistent Volume

## 5 TP Wordpress

We can see we can access Wordpress but because we access it in Minikube by ssh, we cannot see the page and register.

```
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-mast
r/tp/wordpress ➤ kubectl apply -f mysql-deployment.yml
deployment.apps/wordpress-mysql created
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-mast
r/tp/wordpress ➤ kubectl apply -f mysql-service.yml
service/wordpress-mysql created
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-mast
r/tp/wordpress ➤ kubectl apply -f wordpress-deployment.yml
deployment.apps/wordpress-frontend created
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-mast
r/tp/wordpress ➤ kubectl apply -f wordpress-service.yml
service/wordpress-frontend created
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-mast
r/tp/wordpress ➤ kubectl get pod
NAME READY STATUS RESTARTS AGE
wordpress-frontend-5f78fcc7d9-2f4hk 1/1 Running 0 16s
wordpress-mysql-56cfbb4598-2gwq7 1/1 Running 0 28s
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-mast
r/tp/wordpress ➤ kubectl apply -f wordpress-ingress.yml
ingress.extensions/wordpress-ingress unchanged
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-mast
r/tp/wordpress ➤ kubectl get svc
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 28h
wordpress-frontend NodePort 10.98.158.196 <none> 80:31627/TCP 33s
wordpress-mysql ClusterIP None <none> 3306/TCP 46s
```

Figure 49: TP Wordpress

```
$ curl 10.98.158.196:80
$ curl 10.98.158.196:80
```

Figure 50: TP Wordpress

We can also do it by creating 2 persistent volumes.

```
r/tp/wordpress ➤ kubectl apply -f mysql-pvc.yml
persistentvolumeclaim/mysql-pvc created
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hell
r/tp/wordpress ➤ kubectl apply -f mysql-deployment2.yml
deployment.apps/wordpress-mysql created
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hell
r/tp/wordpress ➤ kubectl apply -f mysql-service.yml
service/wordpress-mysql created
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hell
r/tp/wordpress ➤ kubectl apply -f wordpress-pvc.yml
persistentvolumeclaim/wordpress-pvc created
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hell
r/tp/wordpress ➤ kubectl apply -f wordpress-deployment2.yml
deployment.apps/wordpress-frontend created
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hell
r/tp/wordpress ➤ kubectl apply -f wordpress-service.yml
service/wordpress-frontend created
```

Figure 51: TP Wordpress

```
ter/tp/wordpress ➤ kubectl get pvc
NAME      STATUS   VOLUME                                     CAPACITY   ACCESS
MODES   STORAGECLASS AGE
mysql-pvc   Bound    pvc-004baaec-8a90-11e9-abf4-080027f31393  20Gi     RWO
          standard   32s
wordpress-pvc   Bound    pvc-08e68402-8a90-11e9-abf4-080027f31393  20Gi     RWO
          standard   17s
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/tp/wordpress ➤ kubectl get pod
NAME                  READY   STATUS    RESTARTS   AGE
wordpress-frontend-65777c67f5-lp6qw   1/1     Running   0          20s
wordpress-mysql-6fc75c9996-hdps8   1/1     Running   0          32s
luodihao@luodeMacBook-Pro ~/Desktop/2018-2019 春/CC tencent/report/hello-k8s-maste
r/tp/wordpress ➤ kubectl get svc
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
kubernetes   ClusterIP  10.96.0.1    <none>        443/TCP     28h
wordpress-frontend   NodePort   10.108.21.61  <none>        80:30842/TCP  21s
wordpress-mysql   ClusterIP  None        <none>        3306/TCP     33s
```

Figure 52: TP Wordpress