İlk komutumuz kubectl cluster -info

Kubectl bizi kubernetesa bağlayan arayüzümüz.

Kubernetes te hayatımız objeler

Objeleri yöneten kubernetes i halledr.

Objelerin kısaltılmışı var.

Kubernetes te çok komut var ancak toplamda kullacağımız komut sayıs 3-4 tür.

Kubernetes in olayı yaml dosyasını yazmak;

Object Model 
apiVersion: apps/vl 
kind: Deployment 
metadata: 
name: nginx-deployment 
spec: 
selector: 
matchLabels: 
app: nginx 
replicas: 2 
template: 
metadata: 
labels: 
app: nginx 
spec: 
containers: 
- name: nglnx 
image: nginx:1.14.2 
ports: 
- containerPort: 8b 
All objects must have apiVersion, kind, 
metadata and spec fields. 
• 
• 
of 
the 
apiVersion: Which 
version 
Kubernetes API you're using to create this 
object 
kind: What kind of object you want to create 
metadata: Data that helps uniquely identify 
the object, including a name string, labels, 
and optional namespace 
spec: What state you desire for the object 

code mypod.aml dediğimizde yeni bir yaml dosyasını açıyor vscode.

4 temel kısım var;

! mypodyaml 
1 
2 
3 
4 
apiVersion: 
kind : 
metadata : 
spec:l 

kind, hangi objeyi oluşturacaksak o, her objenin bir versionu var.



birinci yöntem bu komut.

• ubuntuekube-mastör:-$ kubectt explain pod 
KIND: 
VERSION: VI 

bu kod da ikinci yöntem.

metadata objeyi tanımlayan datalar.



name çalışması içn yeterlidr. bir de tag gibi label verebilirz key-value şeklinde:

3 
metadata : 
name: nginx-pod 
app: nginx 

spec kısmında ise docker da run komutnda girdğimiz değerleri tanımlıyoruz aslında:

8 
12 
spec : 
containers: 
• name: mynglnx 
image: nginx 
ports: 
- • containerport 

isim verdik, image ve port belirledik. burad bir POD u yazmış olduk.

Objeyi hangi komutlarla oluşturacaz?

Find more information at: https://kubernetes.io/docs/reference/kubectl/ 
Basic Ccnnands 
create 
expose 
Basic Connands 
explain 
ed it 
delete 
(Beginner) : 
Create a resource from a file or from stdin 
Take a replication controller, service, deployænt or pod and expose it as a new Kubernetes se 
Run a particular image on the cluster 
Set specific features on objects 
(Internediate) : 
Get documentation for a resource 
Display one or many resources 
Edit a resource on the server 
Delete resources by file nanes, stdin, resources and names, or by resources and label selector 
Deploy Comands: 
rol lout 
scale 
autoscale 
Manage the rollout of a resource 
Set a new size for a deployment, replica set. or replication controller 
Auto-scale a deployment, replica set, stateful set, or replication controller 
Cluster Managerænt Connands: 
certificate 
clus ter-info 
top 
co rdon 
Modify certificate resources. 
Display cluster information 
Display resource (CPU/memory) usage 
Mark node as unschedulable 

kubectl create -f mypod.yaml komutu ile POD u oluşturuz:

• ubuntu@kube-naster: -S 
pod/nqinx•pod created 
c ubuntu@kube-master: -S 
kubectl 
1 
create 
1 

• N$ kubectl get pods 
• ubuntu@kube-master . 
READY 
STATUS 
nginx-pod 
ContainerCreating 
• N$ kubectl get pods 
• ubuntu@kube-master . 
READY 
STATUS 
nginx-pod 
ContainerCreating 
-master:æ$ 
o ubuntu@kube 
RESTARTS 
-o wide 
RESTARTS 
AGE 
9s 
AGE 
17s 
<none> 
NODE 
kube-worker 
'MINATED NODE 
<none> 
READINESS GATES 
<none> 

kubectl get pods pod un özelliklerini gösterir, -o wide ile dah geniş bilgi gösterir.

• kubectl describe pods nginx-pod 
Name : 
Namespace: 
Priority : 
Service Account: 
Node: 
nginx-pod 
default 
default 
kube-worker/172.31.21.123 

describe ile detaylı bilgileri görürüz.

kubectl get pod/pods/po hepsi aynı işi görür.

objenin yaml halini alalım:

• kubectl get pod/nginx-pod 
apiVersion: VI 
kind: Pod 
metadata : 
creation Timestamp: 
labels : 
app: ng1nx 
name: nginx-pod 
namespace: default 
resourceVersion: "4558" 
uid: 6f3Ø8d64-4714-48c5-be62-1ef91bØ19ba9 
-o yaml 
spec : 
containers : 
Image: ng1nx 
imagePu11P01icy : 
name: mynglnx 
ports : 
containerPort: 
protocol: TCP 
resources: 
Always 
80 
terminationMessagePath: /dev/termination-log 
terminationMessageP01icy: File 
volumeMounts : 
mountPath: /var/run/secrets/kubernetes . io/serviceaccount 
name: kube-api-access-8cfdr 
readOn1v: true 

kubectl get pod/nginx-pod -o yaml ; burada / yerine boşluk da kullanılabilir.

Biz kısa bir şey yazdık, output u ise çok uzun. çünkü default değerler var.

pod u silmek için yaml dosyasını da silebiliriz, pod u da:

#kubectl delete -t mypod.yaml 
• kubectl delete pod ngrnx-pod 
pod •nginx-pod" deleted 
• kubectl get po 
No resources found in default namespace. 
Lv ubuntu@kube-master: -S 

kubectl delete -f mypod.yaml ya da kubectl delete pod nginx-pod

Kubernetes için dokümantasyonu okumak çok önemli.

Birden fazla podu çalıştrmak isteyebiliriz. ReplicaSet de bu işe yarar.

yaml dosyasın oluşturalım:

• kubectl explain rs 
KIND : 
ReplicaSet 
VERSION: apps/vl 

versionuna baktık, rs, ReplicaSet kısaltılmışı

1 
2 
3 
5 
6 
8 
9 
io.k8s.api.apps.v1 ReplicaSet 
apiVersion: apps/vl 
kind: Replicaset 
v metadata: 
name: nglnx-rs 
v labels: 
environment: dev 
replicas : 

yaml dosyasını nsıl yazacğımzı dokümanasyondan bakıyoruz.

spec : 
replicas: 3 
selector : 
matchLabe1s : 
app: ng1nx 

replicas: 3 sayıyı tutar.

template ile de hangi pod un rs si olacğını belirlyoruz. bunu da pod un metadatasını ve spec kısmını alarak yapıyoruz:

template : 
metadata : 
labels : 
app: nginx 
spec : 
containers : 
name: mynginx 
image: nginx 
ports : 
- containerport 
: 80 

yapıştırırken name kısmın siliyruz.

kubernetes te apply komutu ile oluşturaağız, create kullanmamıza gerek yok.

• ubuntuekube-naster:-$ kubectl apply -f myrepucaset.yaml 
replicaset. apps/nginx-rs created 
• kubectl get replicaset 
DESIRED CURRENT READY AGE 
nginx- rs 3 
• kubectl get po 
READY STATUS RESTARTS 
nginx• rs -9tp2j 1/1 
Running e 
nqinx- rs- f knuk 1/1 
Running 
nginx- rs -tmr7f 
Running e 
AGE 
• ubuntuekube-master:-S kubectl delete -f myreplicaset.yæl 
replicaset.apps "nginx- rs" deleted 

aynı anda 3 pod oluşturduk, bunu docker da yapamazdık.

1. objeye geldik: Deployment

RS yi olduğu gibi kopyalıp buraya yaptışrıtıroyuz.

mydeployment.yaml > {l metadata > @ nam 
1 
2 
3 
5 
6 
7 
8 
9 
le 
11 
12 
13 
14 
15 
16 
17 
18 
19 
2e 
21 
io.k8s.api.apps.v1 Deployment (VI @deploym. 
apiVersion: apps/vl 
kind: Deployment 
metadata : 
name: nginx-deployment 
labels : 
environment: dev 
spec : 
replicas: 3 
selector : 
matchLabe1s : 
app: ng1nx 
template : 
metadata : 
labels : 
app: nglnx 
spec : 
containers : 
name: m n Inx 
Ima e: n Inx 
ports : 
containerPort : 
80 

sadece kind i değiştirdik. vesionu konrol ettik aynı:

kubectl apply -f mydeployment.yaml

komutuyla deployment ı olşturduk:

• kubectl apply -f mydeployment.yaml 
deployment. apps/nginx-deployment created 
• kubectl get deployment 
READY UP-TO-DATE AVAILABLE AGE 
nginx•deployrnt 3/3 
• ubuntu@kube-master:-S kubectl get replicaset 
DESIRED CURRENT READY 
nginx-dep10yment-569655 
o ubuntu@kube-master 
3 
3 
AGE 
28s 

dployment ile bbirlikte rs de oluşturdu. onu ayrıca oluşturmaya gerek yok. biz k8s de bir uygulama çalıştıracaksak ana objemiz deployment tır. deployment oluşturnca rs ve pod lar da oluşur.

Pod Selector 
apaversxon: apps/vl 
kind: 
metadata: 
name: nginx-rs 
labe Is : 
environment: dev 
spec : 
replicas: 3 
se Lector: 
matchLabe Is : 
app: nginx 
template: 
metadata: 
Labels: 
. n Inx 
spec : 
containers: 
- name: mynginx 
image: nginx:1.19 
ports: 
containerPort: 89 

pod un deployment a bağlanabilmesi için selector altındaki matchlabel ile eşlştirmemiz gerekiyor. aynı template dosyasında olması yetmez.

yaml dosyasında günceleme yaptık ve create ile çalştıralım:

ubuntu€kube-master:-S kubectl create -f mydeploynent.yaml 
Error from server (AlreadyExists).; error when creating •mydeployænt.yam 
• ubuntuekube-master:-$ kubectl ap*y -f mydeployment.yaml 
deployment. apps/nginx- deployment configured 
1 
ubuntu@kube-master: -S 

çalışmaz zaten var diyor. apply dediğimizde çalışır.

selector altındaki matchlabels ise değiştirilemez, değişirse deployment ın silinip yeniden apply edilmesi gerekir.

• kubectl 
get deploy 
READY UP-TO-DATE AVAILABLE 
nginx-deployment 
• -S kubectt 
• ubuntuekube-naster. 
get rs 
AGE 
995 
DESIRED CURRENT READY AGE 
nginx-deployment-569655b64d 
• ubuntu@kube-master : -S kubectl 
3 
get po 
3 
READY 
• 2 j zll 
nqinx• deployment -569655b64d - fbmvm 
nqinx-deployment-569655b64d-wds6w 
ubuntu@kube-naster: 
STATUS 
Running 
Running 
Running 
104 s 
RESTARTS 
e 
e 
AGE 
198s 
108s 

rs , pod ve deploymentlara kubectl get dedik, sağ tarafta ise kpcl get pods -w (watch) diyouruz, sol tarta bir pod silelim:

• ubuntu@kube-naster:-S kubectl 
nginx-deployment 
ubuntuekube-naster. 
nginx-deployment-569655b64d-2jzll 
nginx - dep -569655b64d - qrh29 
get deploy 
READY UP-TO-DATE AVAILABLE 
• -S kubectt 
get rs 
AGE 
To run a comand as 
administrator 
See "man sudo root" 
for details. 
• -S kubectl get 
o ubuntu@kube-naster. 
(user "root"), use 
"sudo . 
DESIRED CURRENT READY AGE 
AGE 
1985 
1085 
1085 
NME 
nginx - deployænt 
nginx-deployment 
nginx-deployment 
nqinx•deploytænt 
m52s 
nginx-deployænt 
nginx -deployment 
nginx-deployænt 
nginx - dept oyment 
2m53s 
nginx•deployænt 
2m53s 
2m53s 
pods -w 
READY 
1/1 
1/1 
1/1 
STATUS 
Running 
Running 
Running 
RESTARTS 
nginx- deployænt-569655b64d 
• kubectl 
get po 
3 
READY 
nqinx•deployment-569655b64d - fbmvm 1/1 
nqinx-deployment-569655b64d -wds6w 1/1 
STATUS 
Running 
Running 
Running 
104 s 
RESTARTS 
e 
e 
-569655b64d-2jzU 
-569655b64d - fbmvm 
-569655b64d-wds6w 
-569655b64d.2jzU 
-569655b64d -qrh29 
-569655b64d-qrh29 
-569655b64d-qrh29 
-569655b64d 
-2jzll 
-569655b64d-2jzU 
Terminating 
Pendi nq 
• ubuntuekube-naster:-$ kubectl delete pod nginx-deploynent-569655b64d-2jz11 
pod "nginx-dep10yment-569655b64d-2jz11" deleted 
• kubectl get po 
READY 
nginx - deployment -569655b64d - fbmvm 
nginx-deployment-569655b64d -qrh29 
c ubuntu€kube-master : -S 
STATUS 
Running 
Running 
Runni ng 
RESTARTS AGE 
3m1s 
e 
0 
ContainerCreating 
Te mina ting 
Terminating 
Terminating 
Running 
AGE 
2m26s 
2m26s 
2m26s 
0 

hemen yenisi oluştu sağ tarfta gördük. çünkü rs de sayıyı 3 verdik.

podların loglarına baklım:

• kubectl logs nginx-dep10yment-569655b64d- fbmvm 
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration 
/docker-entrypoint .sh: Looking for shell scripts in /docker-entrypoint .d/ 
/docker•entrypoint.sh: Launching /docker•entrypoint.d/le• 
le- info: Getting the checkstm of /etc/nginx/conf .d/default.conf 
listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf .d/default.conf 
/docker-entrypoint .sh: Launching /docker-entrypoint.d/2e-envsubst-on-templates.sh 
/docker-entrypoint . sh: Launching /docker-entrypoint .d/30-tune-worker-processes.sh 
/docker-entrypoint. sh: Configuration complete; ready for start up 
2923/03/29 19:45:14 
2923/03/29 
2923/03/29 
2023/03/29 
2e23,'d3/29 
2023/03/29 
2e23/03/29 
2023/03/29 
[notice) 
[notice] 
[notice) 
[notice) 
[notice) 
[notice] 
[notice) 
[notice] 
141: using the "epoll" event method 
nginx/1.23.4 
built by gcc 18.2.1 20216110 (Debian 19.2.1-6) 
OS: Linux 5.15.e.1e28.aws 
141: 1048576: 1048576 
141 : 
start worker processes 
: start worker process 29 
1#1: start worker process 30 

pod a bağlanmak içn burad da exec kulanırz:

• kubectl exec nginx-dep10yænt-569655b64d-fbmvm date 
wed Mar 29 UTC 2023 
• kubectl exec nginx-deploynent-569655b64d-fbmvm Is 
bin 
boot 
dev 
docker-entrypoint. d 
docker -entrypoint. sh 
etc 
home 
lib 
lib64 
media 
proc 
root 
sbin 
sys 
tmp 
us r 
var 
1 

date ile tarihi, ls ile içerisinin listelenmesi, -it ile de terminaline bağlanırız.

nginx -deployment -569655b64d -Wd sow l/ I 
Running 
5m55s 
u kubectl exec -it nginx-deptoyment-569655b64d-fbmvm 
bin boot dev docker-entrypoint.d docker-entrypoint.sh etc home lib 
cd I 
- bash 
I ib64 
media 
mnl 

scale ile de sayıyı artırırz ya da azaltırız:

• kubectl scale deployment nqnx-deployment 
nginx- deployment -569655b64d - f52mv 
deployment. apps/nginx-deployment scaled 
• ubuntuekube-naster:-$ kubectl get po 
READY 
nginx-deployment-569655b64d -2hq4t 1/1 
nginx- deployment -569655b64d -5mh6t 
nginx-deployment-569655b64d - fbmvm 1/1 
•qrh29 1/1 
nginx•deployment-569655b64d -wd s6w 1/1 
STATUS 
Running 
Running 
Running 
Running 
Running 
RESTARTS 
e 
• kubectl scale deployment nginx-deployment 
deployment. apps/nginx-deployment scaled 
• ubuntuekube-master:-$ kubectl get po 
READY STATUS 
- - replicas—5 
AGE 
7m48s 
4m56s 
7m48s 
- - replicas=15 
RESTARTS AGE 
nginx- deployment 
-569655b64d 
-2hq4t 
nginx-deployment 
-569655b64d 
-5mh6t 
ngznx-deployment 
-569655b64d 
-6trj8 
ngznx•deployment 
.569655b64d 
•72grm 
nginx•deployrnt 
-569655b64d 
-9tcng 
nqinx-deployment 
-569655b64d 
-b6kzk 
nginx -deployment 
-569655b64d-d87zs 
nginx - deployment -569655b64d 
-dkblb 
nginx-deployment-569655b64d- 
fbmvm 
nginx-deployment 
-569655b64d-qrh29 
nqinx-deployment 
-569655b64d 
- r2czq 
nglnx•deployment 
.569655b64d 
• vvkb7 
nginx•deployment 
-569655b64d 
-'„ds6w 
nqinx deployment 
-569655b64d 
- zbzm4 
en 
en 
en 
en 
en 
Running 
Running 
ContainerCreating 
ContainerCreating 
ContainerCreating 
Pending 
pen d ng 
ContainerCreating 
ContainerCreating 
Running 
Runn ing 
ContainerCreating 
ContainerCreating 
Running 
ContainerCreating 
0 
0 
0 
25s 
25 s 
35 
3s 
7n58s 

ancak gidp yaml dsasında replicas sayısnı deiştirip apply dersek sayı değişir:

S2-kubernetes-e2-basic -operations mydeployment.yaml mypod.yaml myrepticaset .yaml 
• kubectl apply -f mydeployment.yaml 
deployment .apps/nginx-deployment configured 
• kubectl get po 
STATUS 
ContainerCreating 
Running 
Running 
Running 
Running 
Running 
Runni ng 
RESTARTS 
nginx•deployment -569655b64d 
-8x1mg 
nginx deployment -569655b64d - k8sqm 
nqinx deployment -569655b64d - nhbm4 
nginx - deployment -569655b64d - pxnvr 
ngi - deployment -569655b64d 
-qt975 
nginx - deployment -569655b64d 
- v2mx9 
nqinx- deployment -569655b64d- zmpm7 
1 
ubuntu@kube-master: -S 
READY 
1/1 
1/1 
o 
0 
0 
1 
AGE 
5s 
5s 
55 

• ubuntuekube-naster:-S kubectl delete -f mydeptoyænt.yam 
deployment. apps "nginx-deployment" deleted 
1 
ubuntuekube-master : -S 

deployment ı silelim.

• alias k—kubect1Üi 
• k get no 
STATUS ROLES 
AGE 
kube-naster 
kube•wo rker 
Ready 
control -plane 127m 
Ready 
127m 
VERSION 
VI. 26.3 
VI. 26.3 

alias ile kubecl yerine k yi tanımlarsak her seferinde kubectl yazmamıza gerek kalmaz.

Deployment uygulamayı güncellemek veya güncellemeyi geri almak için kullanılır.

apiVersion: apps/v1

kind: Deployment

metadata:

name: clarus-deploy

labels:

app: container-info

annotations:

kubernetes.io/change-cause: deploy/clarus-deploy is set as container-info=clarusway/container-info:1.0

spec:

replicas: 2

selector:

matchLabels:

app: container-info

template:

metadata:

labels:

app: container-info

spec:

containers:

- name: container-info

image: clarusway/container-info:1.0

ports:

- containerPort: 80

Burada ek olarak annotations ekledik.

Annotations 
• 
• 
You can use Kubernetes annotations to attach arbitrary 
non-identifying metadata to objects. 
Clients such as tools and libraries can retrieve this metadata. 
You can use either labels or annotations to attach metadata to 
Kubernetes objects. Labels can be used to select objects and to find 
collections of objects that satisfy certain conditions. In contrast, 
annotations are not used to identify and select objects. 

annotations ek açıklama için klullanılr. label olark kullanamadığımız.bilgileri ekle mek istiyoruz. (mail olur, telfon olur, herşey)

ya da kubernetes core yazılımında olmayn third party uygulamaları(AWS sertifika vb.) tanıtmak için kulanıyoruz.

Apply dedik:

• k apply -f mydeployment . yaml 
deployment . apps/clarus-deploy created 
• k get po 
clarus -deploy-7b88c7c4d7-22hhr 
clarus -deploy- 7b88c 7c4d7 -dfxfb 
nginx-pod 
READY 
1/1 
STATUS 
ContainerCreating 
ContainerCreating 
Running 
RESTARTS 
AGE 
9s 
9s 
73m 

Label ları pod ları gruplamak için de kullanıyoruz.

• kubectl get 
clarus- -fc985 
c la rus 7b88c7c4d7 -g4fkf 
pods 
READY 
1/1 
1/1 
1/1 
1/1 
1/1 
1/1 
STATUS 
Running 
Running 
Running 
Pending 
ContainerCreating 
Running 
I Container-creating 
Running 
Running 
Terminating 
Running 
Running 
Running 
Running 
Running 
RESTARTS 
nglnx-deployrent 
ngunx-deployment 
nginx -deployment 
ngznx depl oyrnent 
nglnx-deployrent 
ngxnx-deployment 
nglnx-deployrent 
nginx -deployment 
-deployment 
ngznx 
nginx-deployment 
nglnx-pod 
rs-dh21s 
nglnx- 
nginx-rs-tghwv 
-5444bb6f48-7h5sg 
-5444bb6f48-bj2n8 
- 5444bb6 f48- gm24b 
• 5444bb6f48.n576f 
-5444bb6f48 
- q5zvp 
-5444bb6f48 
-znjq9 
-kldrx 
-569655b64d 
- 569655b64d-n48hg 
-569655b64d 
-tf9t9 
-569655b64d 
-wmlc8 
1 
AGE 
3m22S 
3111225 
2N126s 
2m26s 
2m26s 

Burada yüzlerce pod olacak. Diyelim ki biz environment=dev olan podları görmek istiyoruz:

• kubectl get 
nginx -deployment -5444bb6f48 
- 7h5sg 
nginx -dept -5444bb6f48 
-bj2n8 
nginx -depl oyrnent-5444bb6 f48- gm24b 
nginx-deployment-5444bb6f48-hh2ss 
nginx-deployrent-5444bb6f48 
-n576f 
nginx-deptoyænt-5444bb6f48- q5zvp 
nginx -deployrnent -5444bb6f48- znjq9 
pods -1 
READY 
1/1 
1/1 
1/1 
environment—dev 
STATUS 
Running 
Running 
Running 
Running 
Running 
Running 
Running 
RESTARTS 
AGE 
35S 
285 
30s 
26S 
35S 
32S 
355 

Annotations lar için önceden tanımlı olanlar da var.

mesela change-cause:

kubernetes.io/kfiööuæka 
use 
Exam le: kubernetes. io/change-cause: Okubectt edit 
Used on: All Objects 
- -record deployment foo" 
his annotation is a best guess at why something was changecL 
It is populated when adding - -record to a kubectl command that may change an object. 

bir şey değiştirdiğinde nedenini yazmak için bu kullanılır.

tek komutla deploy,rs ve po görebilirz:

• kubectl get 
READY UP-TO-DATE 
deployment. apps/clarus-deploy 2/2 
2 
DESIRED 
replicaset. apps/clarus -deploy- 7b88c7c4d7 2 
deploy , rs , po 
AVAILABLE AGE 
2 
CURRENT READY 
AGE 
2 
READY 
pod/ clarus- deploy- 7b88c7c4d7- fc986 1/1 
pod/clarus- deploy- 7b88c7c4d7 - g4fkf 1/1 
1 
STATUS 
Running 
Running 
2 
RESTARTS AGE 
12m 

bunu tek bir label için de yapabaiiriz:

• ubuntu@kube-master : -"deployment - tesson$ 
READY 
deployment . apps/ctarus -deploy 2/2 
kubectl get 
UP-TO-DATE 
2 
DESIRED 
deploy, rs,po -l app—container-info 
AVAILABLE AGE 
2 
13m 
CURRENT READY 
AGE 
replicaset . apps/clarus -deploy- 7b88c7c4d7 2 
2 
2 
RESTARTS AGE 
pod/ clarus 
pod/ clarus 
READY 
-deploy- 7b88c7c4d7- fc986 1/1 
-deploy. 7b880c4d7-g4fkf 1/1 
STATUS 
Running 
Running 
1 
o ubuntu@kube-master : -'deployment - tesson$ 

kubectl rollout komutna bakalım:

Available 
history 
pause 
restart 
resume 
status 
undo 
Comands : 
View rollout history 
Mark the provided resource as paused 
Restart a resource 
Resume a paused resource 
Show the status of the rollout 
undo a previous rollout 

history:

• kubectl rollout history deploy clarus-deploy 
deployment . apps/ clarus -deploy 
REVISION CHANGE-CAUSE 
1 
deploy/clarus-deploy is set as 

clarus-deploy un history sine baktık. chnge-cause var, nereden geldi:

app: container •Into 
annotations: 
spec: 
replicas: 2 
deploy/clarus•deploy is set as 

şimdi diyelim ki image versionu güncellendi ve değiştirecez:

containers: 
• name: container-info 
ima e: claruswa 
rts: 
- containerPort: 
container-info:1 0 
80 

kubectl set ile yapacz:

• kubectl set - -help 
Configure application resources. 
These connands help you make changes to existing application resources. 
Avai ble Comands : 
resources 
selector 
servi ceaccount 
subject 
Usage: 
Update environment variables on a pod template 
update the image of a pod template 
Update resource requests/limits on objects with pod templates 
Set the selector on a resource 
update the service account of a resource 
Update the user, group, or service account in a rote binding or cluster role binding 
kubectl set SUBCH•IAND (options] 

var olan kaynakların bileşenlerin deiştirebilir biz image değiştirecez:

kubectl set image deploy clarus-deploy container-info=clarusway/container-info:2.0

clarus-deploy içnde ismi container-info olan konateynırın image in değiştir dioruz:

o k get po 
clarus-deploy 
clarus-deploy 
clarus-deploy 
clarus-deploy 
clarus-deploy 
clarus-deploy 
clarus-deploy 
-w 
READY 
1/1 
1/1 
1/1 
1/1 
1/1 
1/1 
1/1 
STATUS 
ContainerCreating 
Running 
Running 
Running 
Running 
Terminating 
Pending 
Pending 
ContainerCreating 
Running 
Terminating 
RESTARTS 
clarus-deploy 
clarus-deploy 
clarus-deploy 
nginx-pod 
-7476ddc94b 
-hsmth 
-7b88c7c4d7 
-22hhr 
-7b88c7c4d7 
-dfxfb 
-7476ddc94b 
-hsmth 
-7b88c7c4d7 
-dfxfb 
-7476ddc94b-c8w52 
-7476ddc94b-c8w52 
-7476ddc94b-c8w52 
-7476ddc94b-c8w52 
-7b88c7c4d7 
-22hhr 
AGE 
23s 
17m 
17m 
91m 
23s 
17m 
Is 
Is 
Is 
Is 
17m 

yeni konteynırlar oluşturup eskilerini terminate ediyor.

versionların ayrıntıl history si ni şöyle görebiirz:

kubectl rollout history deploy clarus-deploy 
kubectl rollout • history deploy • clarus -deploy 
-revision—I 
-revision—2 

bu komutla yapılan bütün deişilikleri görebilyoruz, çok önemli bir özellik.

• k get po 
clarus-dep10y-7476ddc94b-c8w52 
clarus-dep10y-7476ddc94b-hsmth 
nginx-pod 
• k get rs 
READY 
1/1 
1/1 
1/1 
STATUS 
Running 
Running 
Running 
RESTARTS 
AGE 
3m43s 
4m6s 
DESIRED 
2 
CURRENT 
2 
READY 
2 
clarus -deploy -7476ddc94b 
clarus -deploy-7b88c7c4d7 
AGE 
ami3s 
21m 

her bir revision rs demek, güncellenenlerin sayısını 2 ye çıkarıyor, eskilerini 0 a düşürüyor. güncellemyi geri alırsak tam tersini yapcak.

set komutun daha kolayı kubectl edit komutu:

kubectl edit deploy/clarus-deploy

ya da

kubectl edit deploy clarus-deploy

bize vi editörüyle yaml dosyamızı açtı:

Please edit the object below. Lines beginning with a '#' will be ignored, 
# and an empty file will abort the edit. If an error occurs while saving this file will be 
# reopened with the relevant failures . 
apiVersion: apps/vl 
kind: Deployment 
metadata : 
annotations : 
deployment . kubernetes . io/revision: "2" 
kubectl. kubernetes . io/last -applied-configuration: I 
{ " apiVersion" : " apps/vl " , "kind " • "Deployment" , "metadata" : {"annotations " : {"kubernetes . i 
rus-deploy is set as "labels" "contai 
eploy" , "namespace" : "default "spec " : {"replicas " : 2, "selector" : {"match Labels" : {"app" : "conte 
etadata" : { " labels " : { " app" : "container-info"}}, "spec " : {"containers " : " image" : "clarusway/cor 
ntainer-info" , "ports " : "containerPort" : 
kubernetes .io/change-cause: deploy/clarus-deploy is set as container-info=clarusway/cc 
"/tmp/kubect1-edit-4046344995. yaml" 72L, 2603B 

en alta indik:

app: container-info 
spec : 
containers : 
image: clarusway/container-info:2.Ø 
imagePu11P01icy: IfNotPresent 
name: container-info 
ports : 
containerPort : 80 
protocol: TCP 
resources: 

burad image i değiştrebilriz.

etadata " : { " labels " : { " app" : "container-info"}}, " spec " : {"containers" : "image" : "clarusway/container-info: 1.0" , " 
ntainer-info" , "ports " : "containerPort" : 
kubernetes .io/change-cause: deploy/clarus-deploy is set as 
creationTimestamp: "2023-03-29T2Ø: 18: 55Z" 
generation: 2 
labels : 
app: container-info 
name 

change-cause u da değiştirdik. kaydedip çıkıyoruz.

• kubectl edit deploy/clarus-deploy 
deployment . apps/clarus-deploy edited 

• ubuntu@kube-master : 
READY 
clarus -deploy 2/2 
• ubuntu@kube-master : 
k 
k 
clarus-dep10y-586659c7d5 
clarus -deploy -7476ddc94b 
clarus -deploy-7b88c7c4d7 
get deploy 
UP-TO-DATE 
2 
get rs 
DESIRED 
2 
1 
AVAILABLE 
2 
CURRENT 
2 
1 
AGE 
26m 
READY 
1 
1 
AGE 
33s 
9m24s 
26m 

eskilerini öldürüyor yenilerini kaldırıyor.

clarus-dep10y-7 
• k rollout history deploy clarus-deploy 
deployment . apps/ clarus -deploy 
REVISION CHANGE-CAUSE 
1 
2 
3 
deploy/clarus-deploy is set as 
deploy/clarus-deploy is set as 
deploy/clarus-deploy is set as container-info=c1arusway/container-info:3.Ø 

3 tane revision oluştu.

şimdi bizim hemen 1 e dönmemiz lazım dediler, 1. versiona geçiyoruz:

kubectl rollout undo deploy clarus-deploy --to-revision=1

tek bir komutla revison 1 e dönüyoruz.

• kubectl rollout undo deploy clarus-deploy 
deployment . apps/clarus-deploy rolled back 
-to-revision—I 

edit komutunun da daha kolay yöntemi yaml dosyasına gidip oradn değişklik yapmaktır. gidelim aml dosyasında image i nginx yapalım.

iki kere revison yaptığmız için 3 farklı rs görüyorz:

• k get rs 
1.0 
DESIRED 
clarus -deploy -586659c7d5 1 
clarus -deploy-7476ddc94b 
clarus -deploy- 7b88c7c4d7 2 
CURRENT 
1 
2 
READY 
1 
AGE 
10m 
18m 
36m 
@ k rollout history 
error: required resource not specified 
• k rollout history deploy clarus-deploy 
deployment . apps/ clarus -deploy 
REVISION CHANGE-CAUSE 
2 
3 
4 
deploy/ clarus-deploy 
deploy/ clarus-deploy 
deploy/ clarus-deploy 
IS 
IS 
IS 
set 
set 
set 
as 
as 
as 
container-info=clarusway/container-info: 
container-info=clarusway/ container-info . 
container-info=clarusway/container-info: 

deployment rs lerle version u kontrol ediyor, değişiklikleri yapabilyor

şu an stabil oldu:

• k get rs 
clarus- 
deploy-586659c7d5 
clarus-dep10y-7476ddc94b 
clarus -deploy-7b88c7c4d7 
DESIRED 
2 
CURRENT 
2 
READY 
2 
AGE 
11m 
20m 
37m 

şimdi yeni bir evison yaptık, yaml file da image değiştidik apply edelim:

• k apply -f mydeployment . yaml 
deployment . apps/ clarus-deploy configured 
• k get rs 
DESIRED CURRENT 
clarus-dep10y-586659c7d5 
clarus -deploy -7476ddc94b 
clarus -deploy-78d7b4f756 
clarus -deploy-7b88c7c4d7 
1 
2 
1 
2 
READY 
2 
AGE 
12m 
21m 
39m 

4.rs oluştu.

cluster ı durdurabiliriz yarın kaldığmız yerden devam edecez.