

Huawei Kubernetes CCE Bootcamp Lab-1

Uygulama Raporu

Konu: Kubernetes Container Orchestration with CCE

Özet: CCE tabanlı Kubernetes işlemlerinde Huawei Console ortamında temel Kubernetes işlemleri , Deployment ve DaemonSet orkestrasyonları denenmiş olup adımlar ve çıktıları raporda açıklanmıştır

Hazırlayan
Mustafa ESEN

2025-ANKARA

1) Giriş

Rapor; Deployment oluşturma, ölçeklendirme, silme, DaemonSet oluşturma, DaemonSet doğrulama ve pod yeniden oluşturulma testlerini içermektedir.Tüm adımlar komut çıktıları ve ekran görüntüleriyle kanıtlanmıştır.

2) Donanım ve Çevresel Bilgiler

Platform: Huawei Cloud • KooLabs Sandbox

OS: CentOS (ECS üzerinde)

Cluster: cce01

Namespace: default / kube-system

CLI: kubectl 1.27+

User: root@ecs-k8s

3) Lab Ortamına Giriş ve Aktifleştirme

KooLabs'te açılan sanal masaüstü ortamında tarayıcıdan cloud hesabına sistem tarafından belirlenmiş SandBox kullanıcı bilgileri ile IAM User tipi ile giriş yapıldı.Daha sonra pratikler yapılırken gerekli olacak cluster,node,ecs temini için aşağıdaki adımlar uygulandı

→Yönergede de belirtildiği üzere yine yönergede belirtildiği gibi **cce01** adında bir cluster oluşturuldu.

→Bu **cluster'a** yine yönergede belirtilen özelliklerde **node-cce01** adında bir node oluşturuldu.

→Yönergede belirtildiği gibi **ecs-k8s** adında bir de elastic cloud server oluşturuldu.

Bu aşamadan sonra yönergede belirtildiği gibi masaüstündeki terminal üzerinden bu makineye bağlanmaya çalışırken time-out hatası alındı.

Bu hata ise **cluster more →manage group → configure security group →add rule** adımları üzerinden aşağıdaki şekilde tanımlanan 22 portunu erişime açan kural ile çözümlenmiştir.

Protocol: TCP

Port Range: 22

Source: 0.0.0.0/0

Description: SSH Access

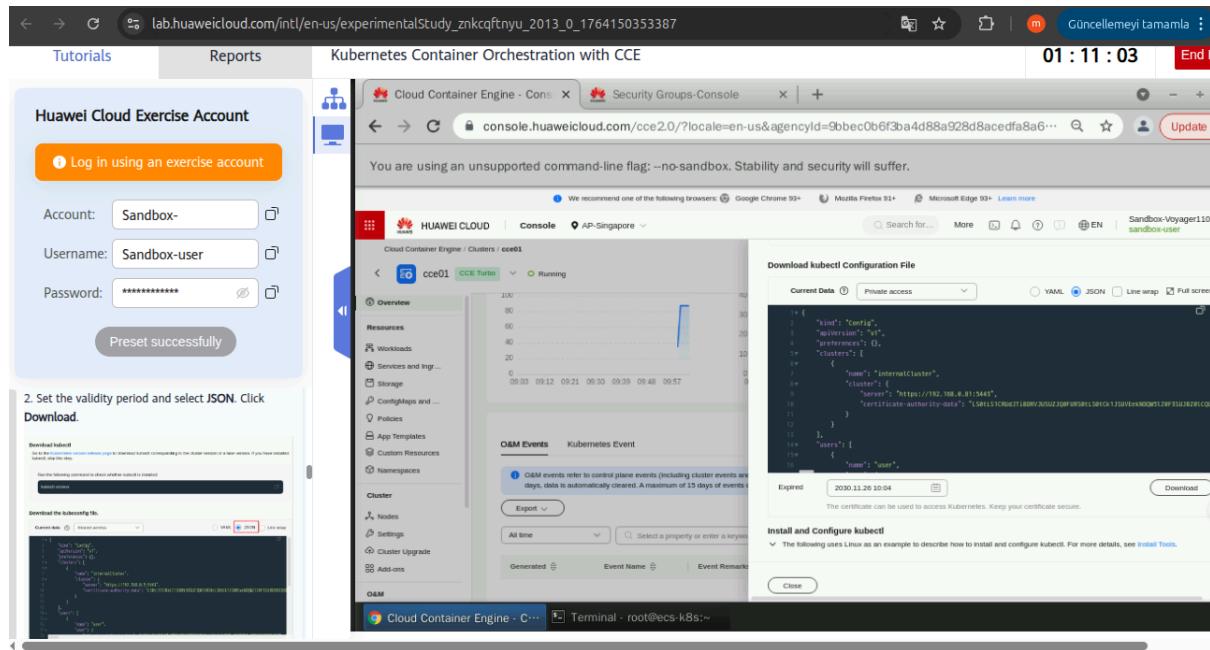
(Görsel 1.0 : TCP 22 portu kuralı eklenmesi.)

4) Makine Bağlantısı ve KubeConfig Dosyası import etme

LANG=en_us.UTF-8 ssh root@159.138.120.215 ile eip'si ile ve makine alırken oluşturulan parola bilgisi ile bağlantı gerçekleştirildi.

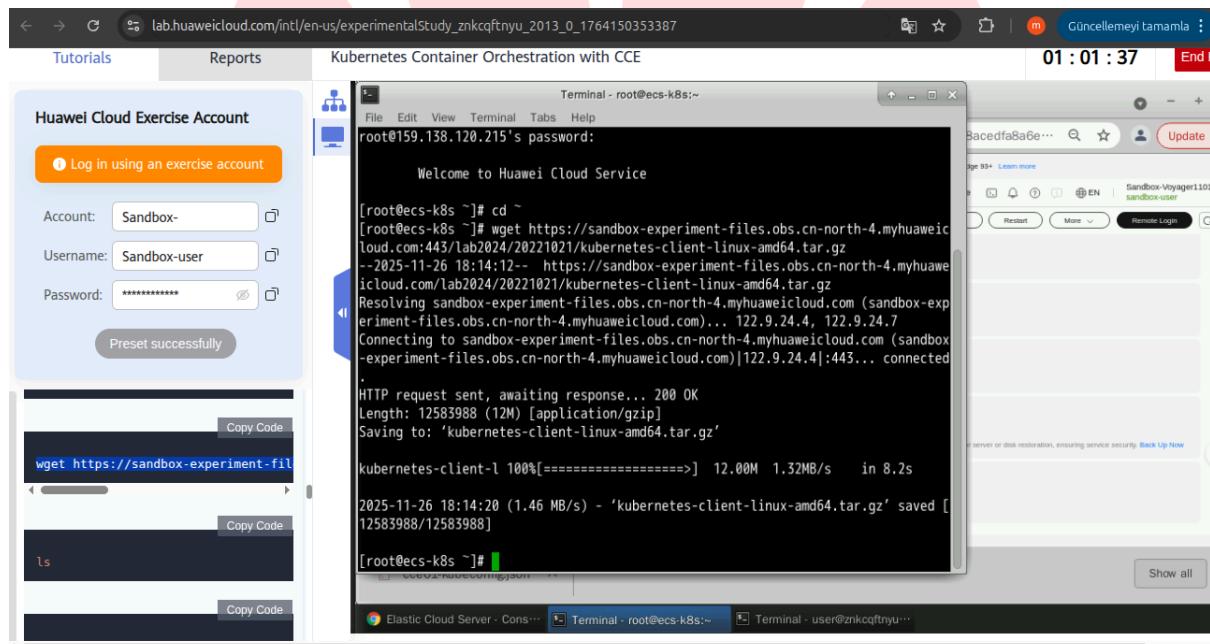
(Görsel 1.1 : EIP ve parola bilgileri ile makine bağlantısı.)

Daha sonra kubeconfig dosyası **cluster→connection information→kubectl→configure** adımları üzerinden JSON formatından lab ortamı masaüstüne indirildi.



(Görsel 1.2 : Kubeconfig Dosyası İndirme.)

Yönergede belirtildiği üzere kubectl komut satırı araçları terminal üzerinden indirildi.



(Görsel 1.3 : Kubectl Komut Araçları Terminal Üzerinden İndirimi.)

İndirilen paket içerisindeki kubectl binary dosyası ve console üzerinden indirilen JSON kubeconfig dosyası home dizinine kopyalandı bu sayede komut istemcisi üzerinden cluster'a bağlanabilecek

```

Terminal - root@ecs-k8s:/home
File Edit View Terminal Tabs Help
Connecting to sandbox-experiment-files.obs.cn-north-4.myhuaweicloud.com (sandbox-experiment-files.obs.cn-north-4.myhuaweicloud.com)|122.9.24.4|:443... connected.
.
HTTP request sent, awaiting response... 200 OK
Length: 12583988 (12M) [application/gzip]
Saving to: 'kubernetes-client-linux-amd64.tar.gz'

kubernetes-client-l 100%[=====] 12.00M 1.32MB/s in 8.2s
2025-11-26 18:14:20 (1.46 MB/s) - 'kubernetes-client-linux-amd64.tar.gz' saved [12583988/12583988]

[root@ecs-k8s ~]# ls
cce01-kubeconfig.json  kubernetes-client-linux-amd64.tar.gz
[root@ecs-k8s ~]# tar -zxf kubernetes-client-linux-amd64.tar.gz
kubernetes/
kubernetes/client/
kubernetes/client/bin/
kubernetes/client/bin/kubectl
[root@ecs-k8s ~]# cp /root/kubernetes/client/bin/kubectl /home/
[root@ecs-k8s ~]# cp cce01-kubeconfig.json /home/
[root@ecs-k8s ~]# cd /home
[root@ecs-k8s home]#

```

(Görsel 1.4 : Kubectl Binary ve Kubeconfig Dosyaları Home Dizinine Kopyalama.)

5) Kubectl'in Sisteme Tanıtımı ve Kubeconfig Aktifleştirimi

Kubectl dosyasına çalışma izni verildi ve **usr/local/bin** içerisinde alınarak global scope yapıldı. Ardından **\$HOME/.kube** klasörü oluşturularak dosya buraya kopyalandı ve **kubectl config use-context internal** komutu ile cluster bağlantısı için aktif edildi.

```

Terminal - root@ecs-k8s:/home
File Edit View Terminal Tabs Help
Saving to: 'kubernetes-client-linux-amd64.tar.gz'

kubernetes-client-l 100%[=====] 12.00M 1.32MB/s in 8.2s
2025-11-26 18:14:20 (1.46 MB/s) - 'kubernetes-client-linux-amd64.tar.gz' saved [12583988/12583988]

[root@ecs-k8s ~]# ls
cce01-kubeconfig.json  kubernetes-client-linux-amd64.tar.gz
[root@ecs-k8s ~]# tar -zxf kubernetes-client-linux-amd64.tar.gz
kubernetes/
kubernetes/client/
kubernetes/client/bin/
kubernetes/client/bin/kubectl
[root@ecs-k8s ~]# cp /root/kubernetes/client/bin/kubectl /home/
[root@ecs-k8s ~]# cp cce01-kubeconfig.json /home/
[root@ecs-k8s ~]# cd /home
[root@ecs-k8s home]# chmod +x kubectl
[root@ecs-k8s home]# mv kubectl /usr/local/bin
[root@ecs-k8s home]# mkdir $HOME/.kube
[root@ecs-k8s home]# cp cce01-kubeconfig.json $HOME/.kube/config
[root@ecs-k8s home]# kubectl config use-context internal
Switched to context "internal".
[root@ecs-k8s home]#

```

(Görsel 1.5 : Sistem Tanıtımı ve Kubeconfig Aktifleştirimi.)

6) Kontrol ve Kubectl Komutları

6.1) kubectl cluster-info

Bu aşamada bağlantının sağlıklı gerçekleşip gerçekleşmediğini anlamak için **kubectl cluster-info** komutu kullanıldı ve cluster'in sağlıklı çalıştığı gözlemlendi.

The screenshot shows a browser window with a terminal session titled "Terminal - root@ecs-k8s:/home". The terminal displays the following command and its output:

```
[root@ecs-k8s ~]# ls
cce01-kubeconfig.json  kubernetes-client-linux-amd64.tar.gz
[root@ecs-k8s ~]# tar -zxf kubernetes-client-linux-amd64.tar.gz
kubernetes/
kubernetes/client/
kubernetes/client/bin/
kubernetes/client/bin/kubectl
[root@ecs-k8s ~]# cp /root/kubernetes/client/bin/kubectl /home/
[root@ecs-k8s ~]# cp cce01-kubeconfig.json /home/
[root@ecs-k8s ~]# cd /home
[root@ecs-k8s home]# chmod +x kubectl
[root@ecs-k8s home]# mv kubectl /usr/local/bin
[root@ecs-k8s home]# mkdir $HOME/.kube
[root@ecs-k8s home]# cp cce01-kubeconfig.json $HOME/.kube/config
[root@ecs-k8s home]# kubectl config use-context internal
Switched to context "internal".
[root@ecs-k8s home]# kubectl cluster-info
Kubernetes master is running at https://192.168.0.81:5443
CoreDNS is running at https://192.168.0.81:5443/api/v1/namespaces/kube-system/services/coredns:dns/proxy
To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
[root@ecs-k8s home]#
```

(Görsel 1.6 : kubectl cluster-info komutu.)

6.2) kubectl –help

Kubectl'in alt komutları ve açıklamalarını görüntülemek üzere **kubectl –help** komutu çalıştırıldı.

The screenshot shows a browser window with a terminal session titled "Terminal - root@ecs-k8s:/home". The terminal displays the following command and its output:

```
Deploy Commands:
  rollout   Manage the rollout of a resource
  scale    Set a new size for a Deployment, ReplicaSet or ReplicationController
  autoscale Auto-scale a Deployment, ReplicaSet, or ReplicationController

Cluster Management Commands:
  certificate  Modify certificate resources.
  cluster-info Display cluster info
  top          Display Resource (CPU/Memory/Storage) usage.
  cordon      Mark node as unschedulable
  uncordon    Mark node as schedulable
  drain        Drain node in preparation for maintenance
  taint       Update the taints on one or more nodes

Troubleshooting and Debugging Commands:
  describe   Show details of a specific resource or group of resources
  logs       Print the logs for a container in a pod
  attach     Attach to a running container
  exec       Execute a command in a container
  port-forward Forward one or more local ports to a pod
  proxy     Run a proxy to the Kubernetes API server
  cp         Copy files and directories to and from containers.
  auth      Inspect authorization
```

(Görsel 1.7 : kubectl –help komutu.)

6.3) kubectl get nodes

Cluster'in aktif durumda olan node'larını görüntüleyebilmek için **kubectl get nodes** komutu çalıştırıldı.

The screenshot shows the Kubernetes Container Orchestration with CCE interface. On the left, there's a sidebar with 'Huawei Cloud Exercise Account' settings (Account: Sandbox-, Username: Sandbox-user, Password: *****) and a '2. kubectl Exercises' section with '1. List nodes.' and a 'Copy Code' button. Below it is a terminal window titled 'Terminal - root@ecs-k8s:/home'. The terminal displays the help documentation for 'kubectl get' and then the command 'kubectl get node'. The output shows one node: '192.168.0.167 Ready <none> 22m v1.32.6-r0-32.0.10'. To the right of the terminal is a browser window showing a dashboard with a user profile and various service icons.

```
specification. The field specification is expressed as a JSONPath expression (e.g. '(.metadata.name)'). The field in the API resource specified by this JSONPath expression must be an integer or a string.  
--template='': Template string or path to template file to use when  
-o=go-template, --o=go-template-file. The template format is golang templates  
[http://golang.org/pkg/text/template/#pkg-overview].  
-w, --watch=false: After listing/getting the requested object, watch for  
changes. Uninitialized objects are excluded if no object name is provided.  
--watch-only=false: Watch for changes to the requested object(s), without  
listing/getting first.  
  
Usage:  
kubectl get  
[(-o|--output=json|yaml|wide|custom-columns=...|custom-columns-file=...|go-temp-  
late=...|go-template-file=...|jsonpath=...|jsonpath-file=...]  
(TYPE[.VERSION][.GROUP] [NAME | -l label] | TYPE[.VERSION][.GROUP]/NAME ...)  
[flags] [options]  
  
Use "kubectl options" for a list of global command-line options (applies to all  
commands).  
[root@ecs-k8s home]# kubectl get node  
NAME STATUS ROLES AGE VERSION  
192.168.0.167 Ready <none> 22m v1.32.6-r0-32.0.10  
[root@ecs-k8s home]#
```

(Görsel 1.8 : kubectl get nodes komutu.)

6.4) kubectl get namespace

Cluster içerisindeki kaynakları ayıran namespace'leri görüntüleyebilmek için **kubectl get namespace** komutu çalıştırıldı. **default**, **kube-system**, **kube-public**, **kube-node-lease** gibi varsayılan namespace'lerin aktif olduğu doğrulandı.

The screenshot shows the Kubernetes Container Orchestration with CCE interface. On the left, there's a sidebar with 'Huawei Cloud Exercise Account' settings (Account: Sandbox-, Username: Sandbox-user, Password: *****) and a '2. List namespaces.' section with a 'Copy Code' button. Below it is a terminal window titled 'Terminal - root@ecs-k8s:/home'. The terminal displays the help documentation for 'kubectl get' and then the command 'kubectl get namespace'. The output shows several namespaces: 'default Active 26m', 'kube-node-lease Active 26m', 'kube-public Active 26m', 'kube-system Active 26m', and 'system Active 5m'. To the right of the terminal is a browser window showing a dashboard with a user profile and various service icons.

```
-w, --watch=false: After listing/getting the requested object, watch for  
changes. Uninitialized objects are excluded if no object name is provided.  
--watch-only=false: Watch for changes to the requested object(s), without  
listing/getting first.  
  
Usage:  
kubectl get  
[(-o|--output=json|yaml|wide|custom-columns=...|custom-columns-file=...|go-temp-  
late=...|go-template-file=...|jsonpath=...|jsonpath-file=...]  
(TYPE[.VERSION][.GROUP] [NAME | -l label] | TYPE[.VERSION][.GROUP]/NAME ...)  
[flags] [options]  
  
Use "kubectl options" for a list of global command-line options (applies to all  
commands).  
[root@ecs-k8s home]# kubectl get node  
NAME STATUS ROLES AGE VERSION  
192.168.0.167 Ready <none> 22m v1.32.6-r0-32.0.10  
[root@ecs-k8s home]# kubectl get namespace  
NAME STATUS AGE  
default Active 26m  
kube-node-lease Active 26m  
kube-public Active 26m  
kube-system Active 26m  
system Active 5m  
[root@ecs-k8s home]#
```

(Görsel 1.9 : kubectl get namespace komutu.)

6.5) namespace Oluşturma

kubectl create namespace new-namespace komutu çalıştırılarak cluster içerisinde new-namespace adında yeni bir namespace oluşturuldu.

The screenshot shows the Kubernetes Container Orchestration with CCE interface. On the left, there's a sidebar with 'Huawei Cloud Exercise Account' settings (Account: Sandbox-, Username: Sandbox-user, Password: *****, Preset successfully). In the center, a terminal window titled 'Terminal - root@ecs-k8s:/home' is open, displaying the command 'kubectl create namespace new-namespace'. The output shows the namespace was created successfully. On the right, there's a status bar showing '00 : 54 : 58' and a browser tab for 'Sandbox-Voyager1101'.

```
--watch-only=false: Watch for changes to the requested object(s), without listing/getting first.

Usage:
  kubectl get
  [(-o|-ooutput=)json|yaml|wide|custom-columns=...|custom-columns-file=...|go-template=...|go-template-file=...|jsonpath=...|jsonpath-file=...]
  (TYPE[.VERSION][.GROUP] [NAME | -l label] | TYPE[.VERSION][.GROUP]/NAME ...)

  [flags] [options]

Use "kubectl options" for a list of global command-line options (applies to all commands).
[root@ecs-k8s home]# kubectl get node
NAME      STATUS   ROLES    AGE     VERSION
192.168.0.167 Ready    <none>   22m    v1.32.6-r0-32.0.10
[root@ecs-k8s home]# kubectl get namespace
NAME        STATUS   AGE
default     Active   26m
kube-node-lease Active  26m
kube-public  Active   26m
kube-system  Active   26m
[root@ecs-k8s home]# kubectl create namespace new-namespace
namespace/new-namespace created
[root@ecs-k8s home]#
```

(Görsel 2.0 : Yeni namespace Oluşturma.)

The screenshot shows the Kubernetes Container Orchestration with CCE interface. The setup is identical to the previous one. The terminal window now shows the result of the 'kubectl get namespace' command, which includes the newly created 'new-namespace'. The status bar shows '00 : 54 : 32'.

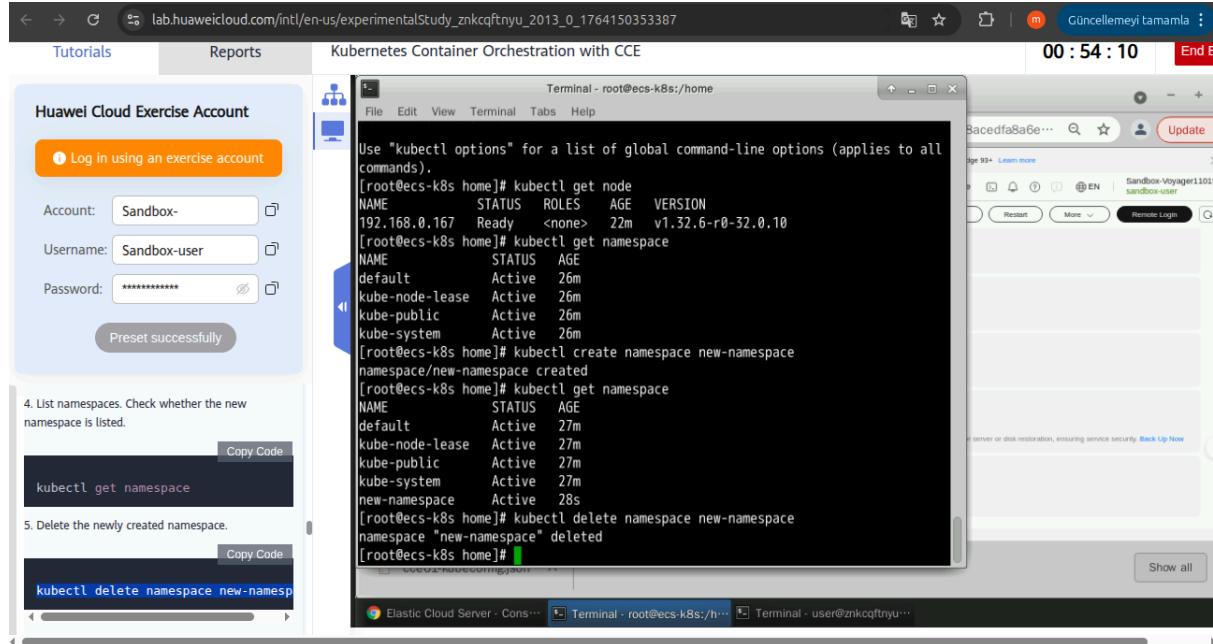
```
(TYPE[.VERSION][.GROUP] [NAME | -l label] | TYPE[.VERSION][.GROUP]/NAME ...)

Use "kubectl options" for a list of global command-line options (applies to all commands).
[root@ecs-k8s home]# kubectl get node
NAME      STATUS   ROLES    AGE     VERSION
192.168.0.167 Ready    <none>   22m    v1.32.6-r0-32.0.10
[root@ecs-k8s home]# kubectl get namespace
NAME        STATUS   AGE
default     Active   26m
kube-node-lease Active  26m
kube-public  Active   26m
kube-system  Active   26m
new-namespace Active  28s
[root@ecs-k8s home]#
```

(Görsel 2.1 : Oluşturulan namespace'in Kontrolü.)

6.6) namespace Silme

kubectl delete namespace new-namespace komutu oluşturulan namespace silindi.

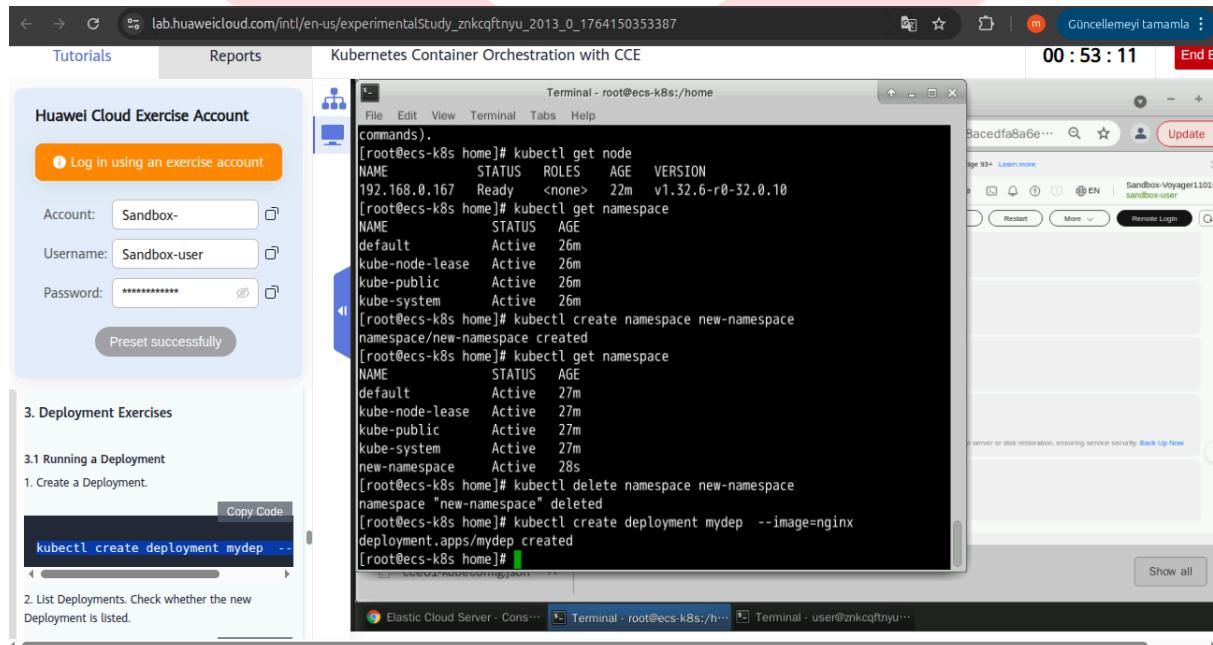


(Görsel 2.2 : namespace Silme.)

7) Deployment

7.1) Deployment Oluşturma

kubectl create deployment mydep --image=nginx komutu ile cluster'da **nginx** imajını kullanan podların istenilen sayıda ve sürümde çalışmasını sağlayan **deployment** oluşturuldu.



(Görsel 2.3 : Deployment Oluşturma.)

7.2) Deployment Kontrolü

kubectl get deploy komutu ile daha önce oluşturulan deployment'in kontrolü sağlandı **mydep** adında deployment oluşturulmuş pod 1/1 ise podun çalışır durumda olduğunu yani Kubernetes'in nginx pod'unu başarıyla ayağı kaldırdığını göstermektedir.

The screenshot shows the Huawei Cloud Exercise Account interface. On the left, there's a sidebar with 'Huawei Cloud Exercise Account' and fields for 'Account: Sandbox-', 'Username: Sandbox-user', and 'Password: *****'. Below these are three numbered steps:

1. Log in using an exercise account
2. List Deployments. Check whether the new Deployment is listed.
3. Display details about the new Deployment.

On the right, a terminal window titled 'Terminal - root@ecs-k8s:/home' shows the following command execution:

```
192.168.0.167 Ready <none> 22m v1.32.6-r0-32.0.10
[root@ecs-k8s home]# kubectl get namespace
NAME STATUS AGE
default Active 26m
kube-node-lease Active 26m
kube-public Active 26m
kube-system Active 26m
[root@ecs-k8s home]# kubectl create namespace new-namespace
namespace/new-namespace created
[root@ecs-k8s home]# kubectl get namespace
NAME STATUS AGE
default Active 27m
kube-node-lease Active 27m
kube-public Active 27m
kube-system Active 27m
new-namespace Active 28s
[root@ecs-k8s home]# kubectl delete namespace new-namespace
namespace "new-namespace" deleted
[root@ecs-k8s home]# kubectl create deployment mydep --image=nginx
deployment.apps/mydep created
[root@ecs-k8s home]# kubectl get deploy
NAME READY UP-TO-DATE AVAILABLE AGE
mydep 1/1 1 1 26s
[root@ecs-k8s home]#
```

(Görsel 2.4 : Deployment Kontrolü.)

7.3) Deployment Detaylarını Görüntüleme

kubectl describe deployment mydep komutu ile deployment'in pod template'i, kullanılan container imajı, güncelleme stratejisi ve geçmiş olayları gözlemlendi. Deployment'in nginx imajını kullanan bir pod oluşturduğu doğrulandı. Rolling update stratejisi, replica set bilgisi ve durum (Available / Progressing) görüntülendi. Events bölümünde Kubernetes'in replica set'i başarıyla 0'dan 1'e ölçeklendiği görüldü.

The screenshot shows the Huawei Cloud Exercise Account interface. The sidebar and steps are identical to the previous screenshot. The terminal window now displays the output of the **kubectl describe deployment mydep** command:

```
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=mydep
  Containers:
    nginx:
      Image:  nginx
      Port:   <none>
      Host Port: <none>
      Environment: <none>
      Mounts:  <none>
      Volumes: <none>
  Conditions:
    Type Status Reason
    ---- -----
    Available True  MinimumReplicasAvailable
    Progressing True  NewReplicaSetAvailable
  OldReplicaSets: <none>
  NewReplicaSet:  mydep-78cc7df666 (1/1 replicas created)
Events:
  Type Reason Age From Message
  ---- ---- - - -
  Normal ScalingReplicaSet 73s  deployment-controller  Scaled up replica set
  mydep-78cc7df666 from 0 to 1
[root@ecs-k8s home]#
```

(Görsel 2.5 : Deployment Detayları.)

Daha sonra **console üzerinden →workloads→deployments** adımları sonunda oluşturulan deployment görüntülendi.

The screenshot shows the Huawei Cloud Exercise Account login page. Below it, a terminal window displays the creation of a deployment named 'mydep'. The deployment status is shown as 'Running' with one pod. The right side of the screen shows the Kubernetes Container Orchestration with CCE console, specifically the 'Deployments' section for the 'cce01' cluster. A deployment named 'mydep' is listed as 'Running' with the image 'nginx'.

(Görsel 2.6 : Console Üzerinden Deployment Kontrolü.)

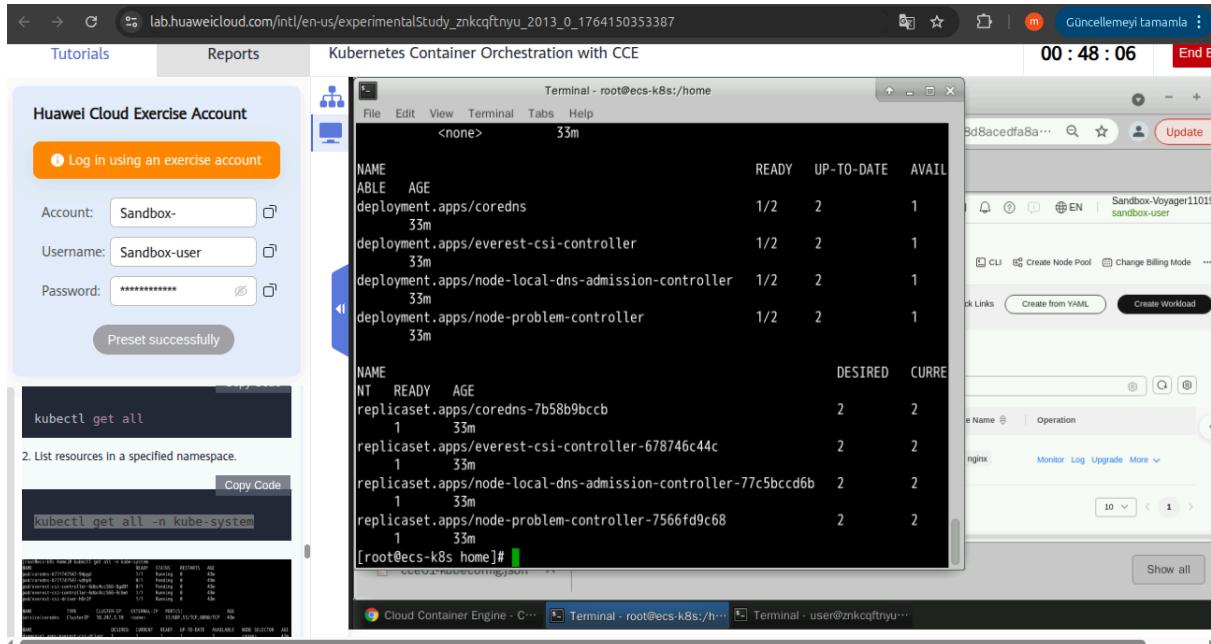
7.4) Tüm Kaynakları İnceleme

kubectl get all komutu ile pod,service,deployments ve replica sets gibi temel kaynaklar görüntülenerek cluster'in genel sağlığı hakkında izlenim elde edinildi.

The screenshot shows the Huawei Cloud Exercise Account login page. Below it, a terminal window runs the command `kubectl get all`. The output lists various resources: a deployment named 'mydep' with one pod running, a service named 'kubernetes' with a ClusterIP of 10.247.0.1, and a replicaset named 'mydep-78cc7df666' with one pod running. The right side of the screen shows the Kubernetes Container Orchestration with CCE console, specifically the 'Deployments' section for the 'cce01' cluster, where the 'mydep' deployment is visible.

(Görsel 2.7 : Temel Kaynakların İncelenmesi.)

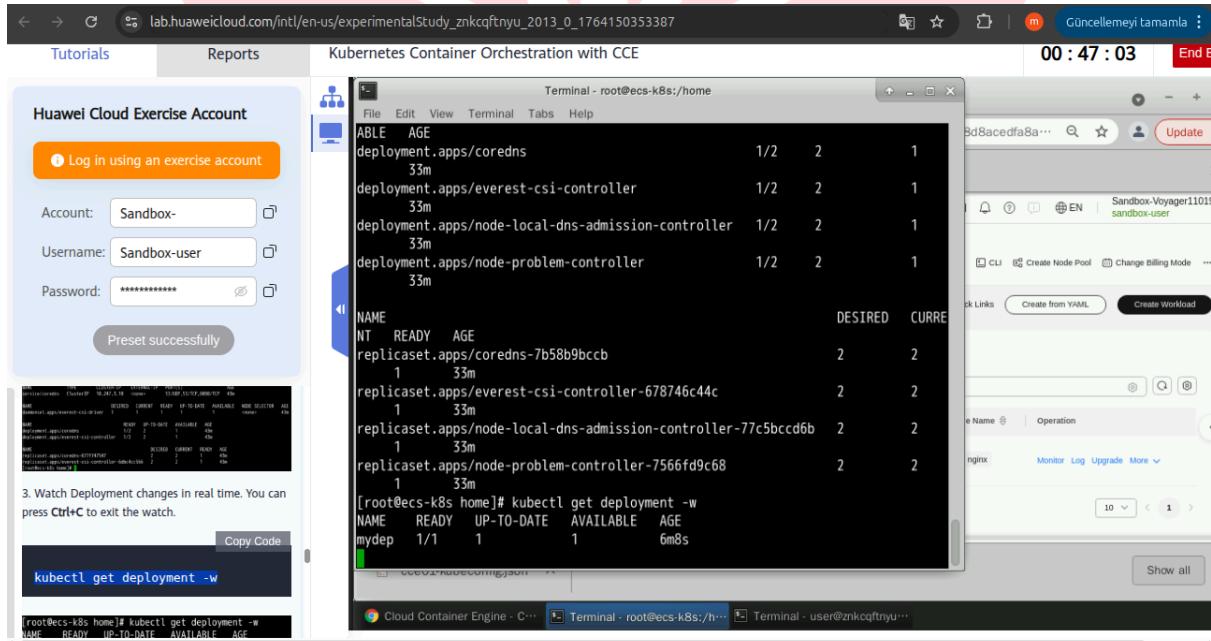
Daha sonra **kubectl get all -n kube-system** komutu ile kube-system namespace’ı içerisindeki tüm kaynaklar listelendi .



(Görsel 2.8 : kube-system namespace altındaki kaynaklar.)

8) Watch Mode

kubectl get deployment -w komutu ile deployment’ın durum değişiklikleri gerçek zamanlı(watch mode) olarak izlendi. Genellikle scaling rolling update durumlarında anlık değişiklikleri gözlemelemek için watch mode kullanılır.



(Görsel 2.9 : Deployment’ın Watch Mode İzlenmesi.)

9) Custom Deployment

YAML dosyası ile özel bir deployment oluşturulmuştur.

9.1) Çalışma Klasörü Oluşturma

YAML dosyasını yazabilmek için içerisinde deploylife adında bir klasör bulunan labfile adında bir klasör oluşturulmuştur. Bu sayede YAML dosyası için uygun alan oluşturulmuştur.

The screenshot shows the Huawei Cloud Container Engine interface. On the left, there's a sidebar for 'Huawei Cloud Exercise Account' with fields for Account (Sandbox), Username (Sandbox-user), and Password. Below this, a terminal window shows the command 'cd deployfile/' being run. On the right, a larger terminal window titled 'Terminal - root@ecs-k8s:/labfile/deployfile' displays the output of several 'kubectl get deployment' commands. The output shows four replicaset objects: coredns, everest-csi-controller, node-local-dns-admission-controller, and node-problem-controller, each with 2 desired and 2 current pods. The timestamp in the top right corner is 00:44:31.

(Görsel 3.0 : YAML İçin Uygun Klasör Yapısı Oluşturma.)

9.2) YAML Dosyasını Uygulama

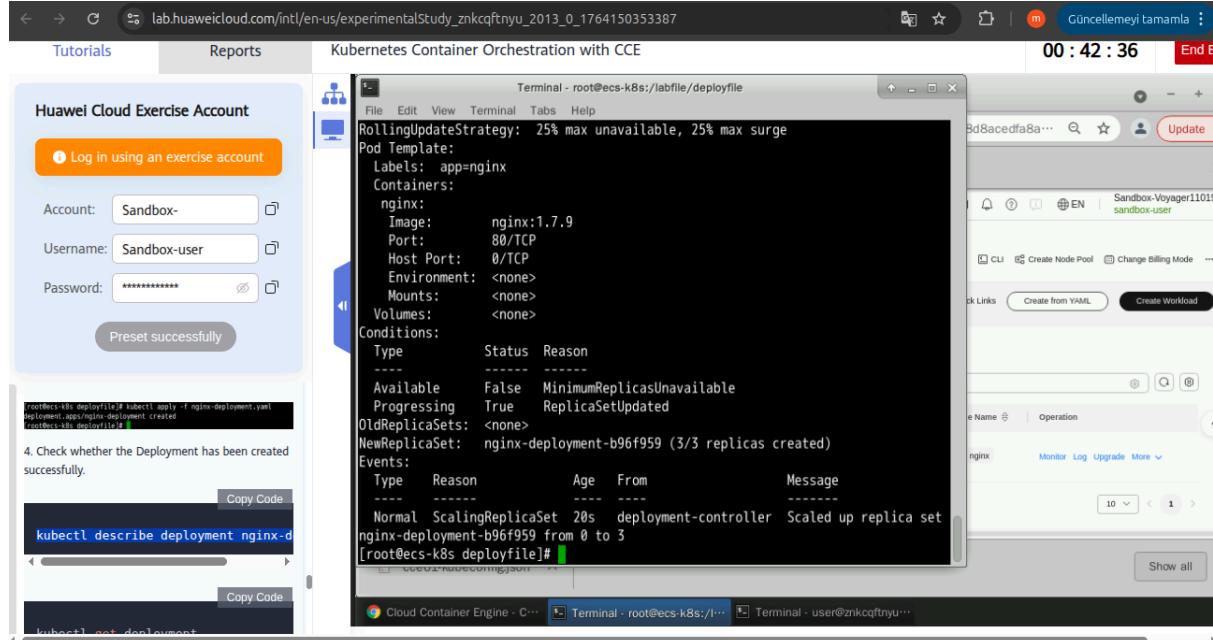
Vim ile belge oluşturulmuş daha sonra yönergedeki YAML kodları eklenerek belge kaydedilmiştir(esc + :wq) . Daha sonra **kubectl apply -f nginx-deployment.yaml** komutu ile **nginx-deployment.yaml** dosyası Kubernetes'e manuel olarak eklenerek yeni bir deployment oluşturulmuştur.

The screenshot shows the same interface as the previous one. The terminal window now shows the command 'kubectl apply -f nginx-deployment.yaml' being run. The output indicates that the deployment has been created successfully. The timestamp in the top right corner is 00:42:57.

(Görsel 3.1 : YAML Dosyasını Kubernetes'e Uygulama ve Deployment)

9.3) Oluşturulan Deployment'in Kontrolü

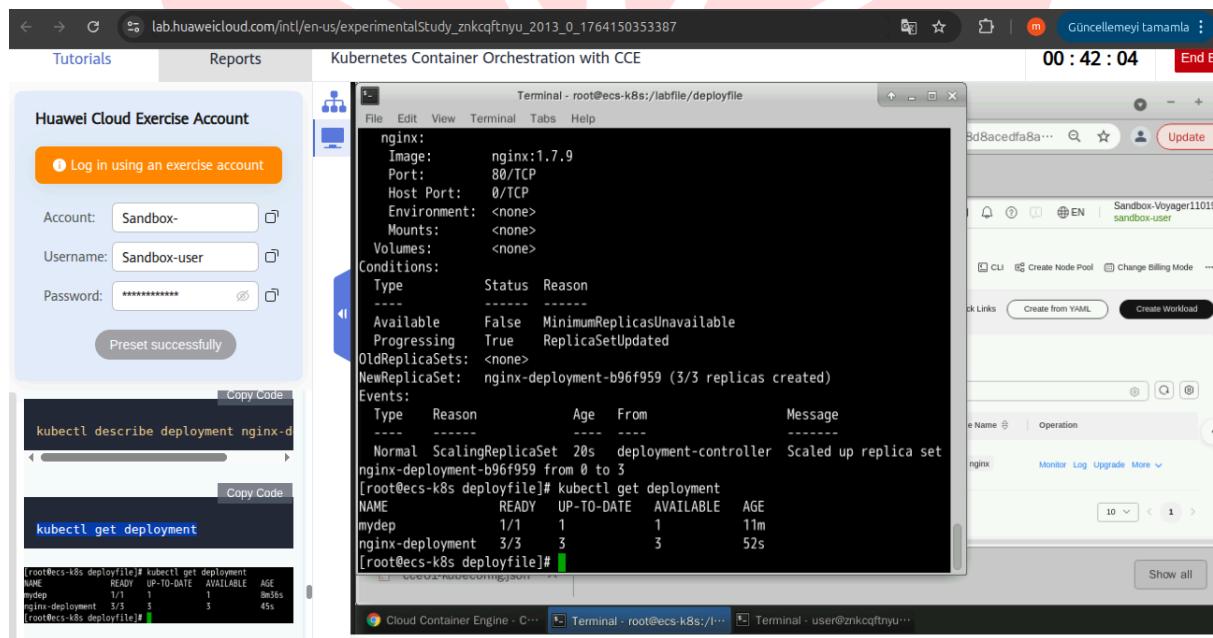
Oluşturulan deployment'ın detaylı özellikleri kubectl describe deployment nginx-deployment komutu ile incelendi . Bu deployment'in nginx:1.7.9 imajını kullandığı ve 0'dan 3 başarılı bir şekilde ölçüklendiği gözlemlendi.



The screenshot shows the Kubernetes Container Orchestration with CCE interface. On the left, there's a sidebar for 'Huawei Cloud Exercise Account' with fields for Account (Sandbox), Username (Sandbox-user), and Password. Below it, a terminal window shows the command 'kubectl apply -f nginx-deployment.yaml' being run, followed by the output: 'deployment.apps/nginx-deployment created'. A note says '4. Check whether the Deployment has been created successfully.' On the right, a larger terminal window titled 'Terminal - root@ecs-k8s:/labfile/deployfile' displays the output of 'kubectl describe deployment nginx-deployment'. It shows the RollingUpdateStrategy (25% max unavailable, 25% max surge), Pod Template (Labels: app=nginx), Containers (nginx: Image nginx:1.7.9, Port 80/TCP, Host Port 0/TCP), and Events (Normal ScalingReplicaSet 20s deployment-controller Scaled up replica set nginx-deployment-b96f959 from 0 to 3). The status shows 3/3 replicas available.

(Görsel 3.2 : Deployment'in Özellik Kontrolü.)

Daha sonra kubectl get deployments komutu ile tüm deployment'lar arasında görüntülendi.



The screenshot shows the Kubernetes Container Orchestration with CCE interface. The left sidebar and terminal window are identical to the previous screenshot. The right side shows a list of deployments in the 'Deployments' section of the interface. It lists 'mydep' and 'nginx-deployment' both with 1/1 ready pods, up-to-date, and available. The 'nginx-deployment' entry includes a 'Logs' button. The bottom terminal window shows the output of 'kubectl get deployment', listing 'mydep' and 'nginx-deployment' with their respective details.

(Görsel 3.3 : kubectl get deployments ile kontrol.)

9.4 Deployment'in Pod'larını Görüntüleme

kubectl get pod komutu ile oluşturulan bu deployment'in (**nginx-deployment**) oluşturduğu tüm pod'lar listelendi. **Replica Set** tarafından 3 pod otomatik şekilde oluşturulduğu ve hepsinin **running** durumunda olduğu gözlemlendi.

The screenshot shows the Kubernetes Container Orchestration with CCE interface. On the left, there's a sidebar for 'Huawei Cloud Exercise Account' with fields for Account (Sandbox), Username (Sandbox-user), and Password. Below it, two terminal windows show command outputs:

- kubectl get deployment**:
Output:

```
root@ecs-k8s deployfile]# kubectl get deployment
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
mydep     1/1     1           1           11m
nginx-deployment 3/3     3           3           45s
[root@ecs-k8s deployfile]#
```
- kubectl get pod**:
Output:

```
root@ecs-k8s deployfile]# kubectl get pod
NAME                  READY   STATUS    RESTARTS   AGE
mydep-78cc7df66-5s7vr 1/1     Running   0          11m
nginx-deployment-b96f959-897x9 1/1     Running   0          103s
nginx-deployment-b96f959-pkwst 1/1     Running   0          103s
nginx-deployment-b96f959-wk72p 1/1     Running   0          103s
[root@ecs-k8s deployfile]#
```

On the right, there's a browser window showing the Cloud Container Engine interface with a list of pods under the 'nginx' workload.

(Görsel 3.4 : Deployment'in Pod'larını Görüntüleme.)

Daha sonra console üzerinden **nginx-deployment'a** tıklanıp pod'lar hesap üzerinden görüntüülendi.

The screenshot shows the Cloud Container Engine interface. On the left, there's a sidebar for 'Huawei Cloud Exercise Account' with fields for Account (Sandbox), Username (Sandbox-user), and Password. Below it, a message says "6. Refresh the CCE console to check the number of replicas and the pod status. Click nginx-deployment to go to its details page. You can view the pods in a list." A terminal window at the bottom shows command outputs.

In the center, a detailed view of the **nginx-deployment** pod is shown:

Pod Name	Status	Nodes	Created
nginx-deployment	Running	1/1	Nov 26, 2025 10:23:06 GMT+00:00

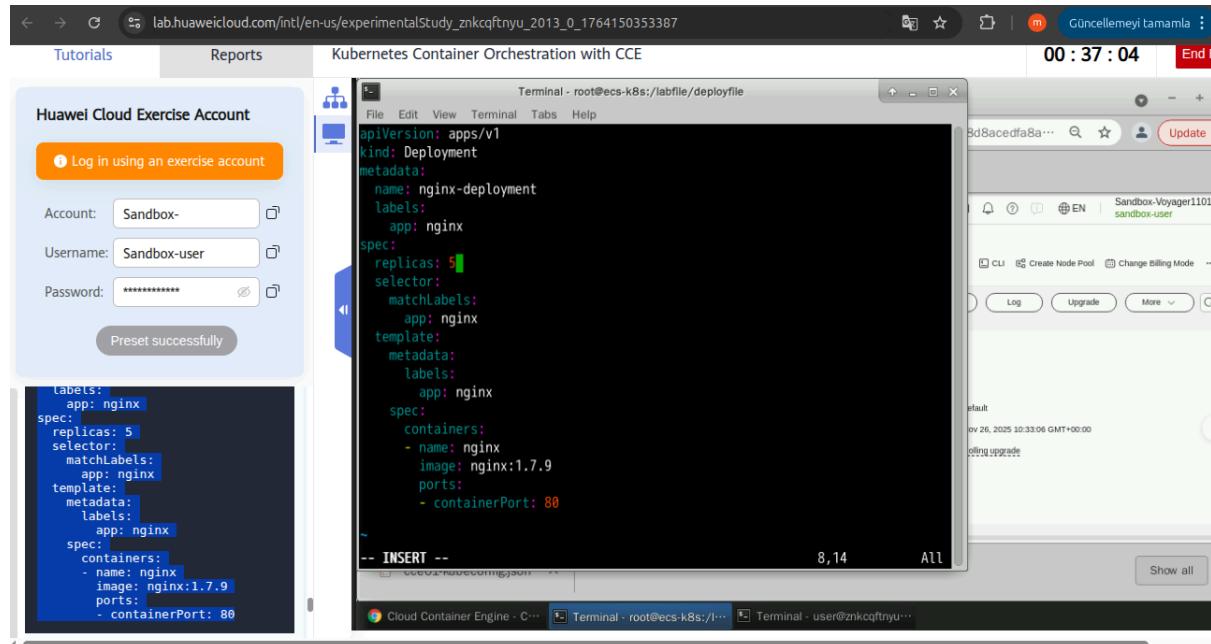
Below this, a table lists all pods:

Pod Name	Status	Nodes	Created
nginx-deployment-6999... (highlighted)	Running	1/1	Nov 26, 2025 10:23:06 GMT+00:00
nginx-deployment-6999... (highlighted)	Running	1/1	Nov 26, 2025 10:23:06 GMT+00:00

(Görsel 3.5 : Pod'ların Console Üzerinden Görüntülenmesi.)

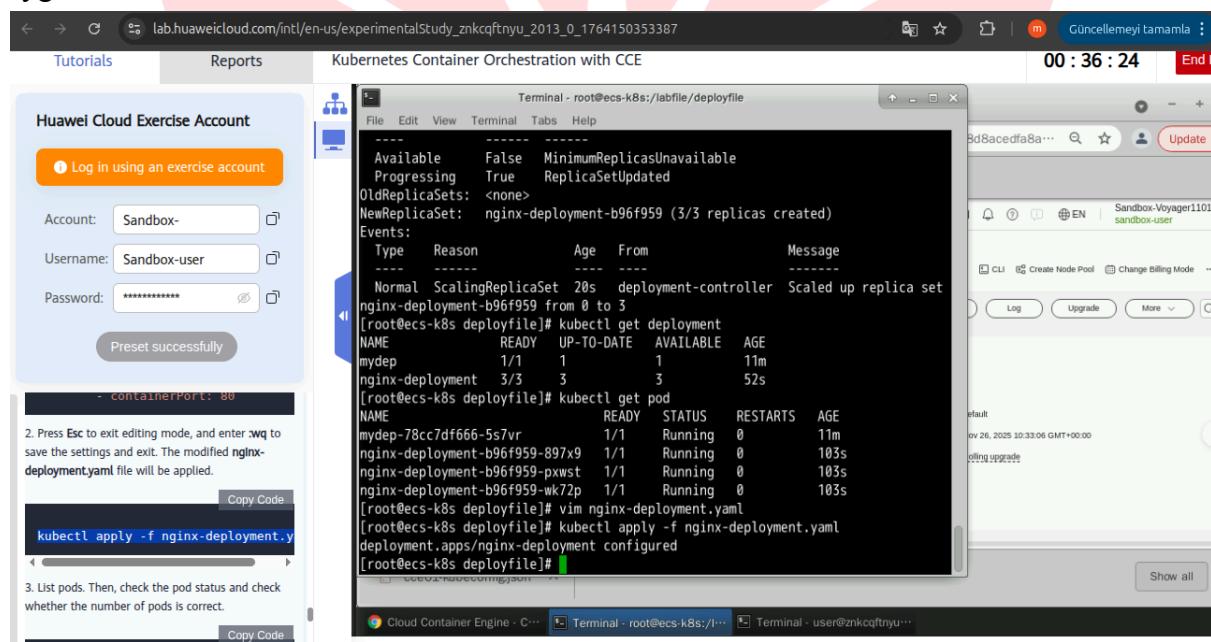
9.5) Deployment Güncelleme

YAML dosyası açılarak replica set sayısı 3'ten 5' güncellendi. Bu şekilde pod sayısı 5' çıkarılmış oldu.



(Görsel 3.6 : Replica Set Sayısını Güncelleme.)

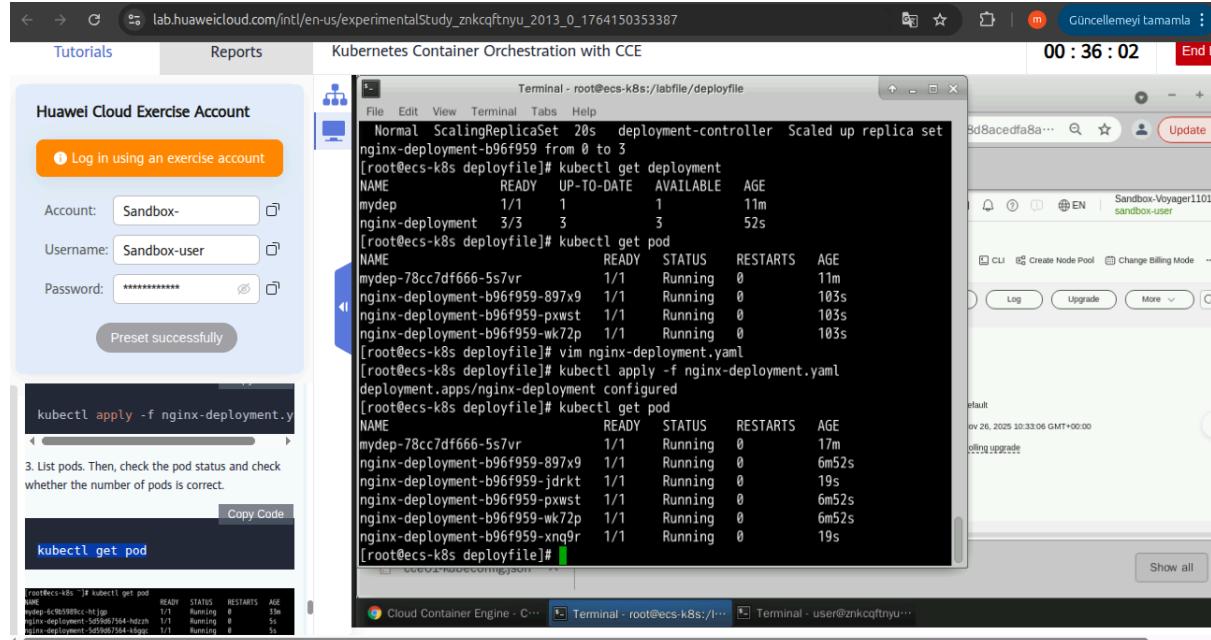
Daha sonra bu değişiklik **kubectl apply -f nginx-deployment.yaml** komutu ile cluster'a uygulandı.



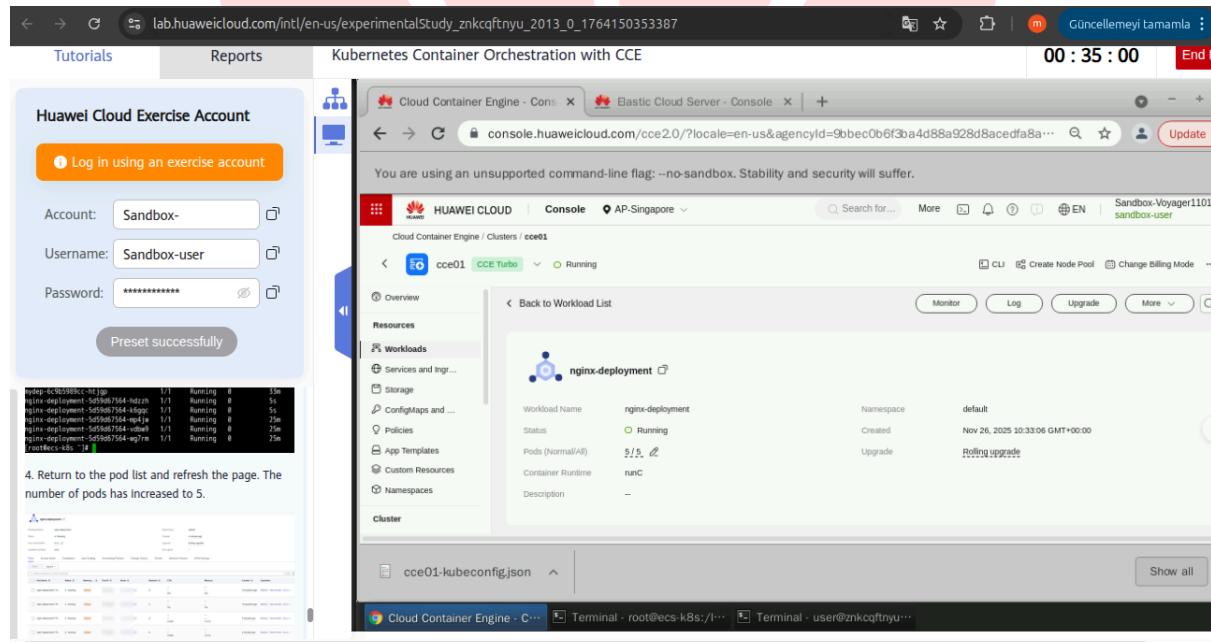
(Görsel 3.7 : Değişikliğin Cluster'a Uygulanması.)

9.6 Yapılan Güncellemenin Kontrolleri

Yapılan güncelleme **kubectl get pod** komutu ile önce terminal üzerinden daha sonra console üzerinden gözlemlenmiştir.



(Görsel 3.8 : Değişikliğin Kontrolü.)



(Görsel 3.9 : Değişikliğin Console Üzerinden Kontrolü.)

9.7 Deployment Silme

kubectl delete deployment nginx-deployment komutu ile nginx-deployment tamamen silinmiştir. Bu silinme ile deployment'in tüm pod'ları da beraberinde silinmiştir.

The screenshot shows the Huawei Cloud Exercise Account interface. On the left, there's a sidebar with 'Huawei Cloud Exercise Account' and a login button. Below it are fields for 'Account' (Sandbox), 'Username' (Sandbox-user), and 'Password'. A message says 'Preset successfully'. In the center, there's a terminal window titled 'Terminal - root@ecs-k8s:/labfile/deployfile'. It shows the command 'kubectl get deployment' followed by its output:

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
mydep	1/1	1	1	11m
nginx-deployment	3/3	3	3	52s

Then, the command 'kubectl get pod' is run, showing:

NAME	READY	STATUS	RESTARTS	AGE
mydep-78cc7df666-5s7yr	1/1	Running	0	11m
nginx-deployment-b96f959-897x9	1/1	Running	0	103s
nginx-deployment-b96f959-pwxst	1/1	Running	0	103s
nginx-deployment-b96f959-wk72p	1/1	Running	0	103s

Next, 'vim nginx-deployment.yaml' is run to edit the deployment configuration. Then, 'kubectl apply -f nginx-deployment.yaml' is run to apply the changes. Finally, 'kubectl get pod' is run again, showing the pods are now running. The terminal ends with 'kubectl delete deployment nginx-deployment' and 'deployment.apps "nginx-deployment" deleted'. The right side of the interface shows a browser window with the Kubernetes Container Orchestration with CCE dashboard.

(Görsel : 4.0 : Deployment Silme.)

10) DaemonSet Oluşturma

Daha önce oluşturulmuş olan lablife klasörü içerisinde YAML dosyasını yazabilmek için daemonlife adında bir klasör oluşturuldu. Vim ile YAML kodu eklendi.

The screenshot shows the Huawei Cloud Exercise Account interface. On the left, there's a sidebar with 'Huawei Cloud Exercise Account' and a login button. Below it are fields for 'Account' (Sandbox), 'Username' (Sandbox-user), and 'Password'. A message says 'Preset successfully'. In the center, there's a terminal window titled 'Terminal - root@ecs-k8s:/labfile/daemonfile'. It shows the command 'kubectl get pod' followed by its output:

NAME	READY	STATUS	RESTARTS	AGE
nginx-deployment	3/3	3	3	52s

Then, the command 'vim nginx-deployment.yaml' is run to edit the deployment configuration. Then, 'kubectl apply -f nginx-deployment.yaml' is run to apply the changes. Finally, 'kubectl get pod' is run again, showing the pods are now running. The terminal ends with 'kubectl delete deployment nginx-deployment' and 'deployment.apps "nginx-deployment" deleted'. The right side of the interface shows a browser window with the Kubernetes Container Orchestration with CCE dashboard.

(Görsel 4.1 : DaemonSet için YAML ve klasör hazırlıkları.)

10.1) YAML Dosyasını Uygulama

`kubectl apply -f DaemonSet.yaml` komutu ile hazırlanan DaemonSet YAML dosyası Kubernetes'e uygulandı. Worker Node'larda çalışacak nginx-daemonset isminde DaemonSet oluşturuldu.

The screenshot shows the Kubernetes Container Orchestration with CCE interface. On the left, there's a sidebar for 'Huawei Cloud Exercise Account' with fields for Account (Sandbox-), Username (Sandbox-user), and Password (*****). Below it, a terminal window titled 'Terminal - root@ecs-k8s:/labfile/daemonfile' shows the command history:

```
mydep-78cc7df666-5s7vr 1/1 Running 0 11m
nginx-deployment-b96f959-897x9 1/1 Running 0 103s
nginx-deployment-b96f959-pwxst 1/1 Running 0 103s
nginx-deployment-b96f959-wk72p 1/1 Running 0 103s
[root@ecs-k8s deployfile]# vim nginx-deployment.yaml
[root@ecs-k8s deployfile]# kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx-deployment configured
[root@ecs-k8s deployfile]# kubectl get pod
NAME READY STATUS RESTARTS AGE
mydep-78cc7df666-5s7vr 1/1 Running 0 17m
nginx-deployment-b96f959-897x9 1/1 Running 0 6m52s
nginx-deployment-b96f959-jdrkt 1/1 Running 0 19s
nginx-deployment-b96f959-pwxst 1/1 Running 0 6m52s
nginx-deployment-b96f959-wk72p 1/1 Running 0 6m52s
nginx-deployment-b96f959-xnq9r 1/1 Running 0 19s
[root@ecs-k8s deployfile]# kubectl delete deployment nginx-deployment
deployment.apps "nginx-deployment" deleted
[root@ecs-k8s deployfile]# mkdir /labfile/daemonfile
[root@ecs-k8s deployfile]# cd /labfile/daemonfile/
[root@ecs-k8s daemonfile]# vim DaemonSet.yaml
[root@ecs-k8s daemonfile]# kubectl apply -f DaemonSet.yaml
daemonset.apps/nginx-daemonset created
[root@ecs-k8s daemonfile]#
```

On the right, a browser window shows the Kubernetes dashboard with a log viewer for a pod named 'nginx-deployment-b96f959-897x9'. The log output includes:

```
default
Nov 26, 2025 10:33:06 GMT+00:00
nginx upgrade
```

(Görsel 4.2 : YAML Dosyasının Kubernetes'e uygulanması.)

10.2) DaemonSet Kontrolü

`kubectl get daemonset --namespace=kube-system` komutu ile DaemonSet'in kube-system namespace'i altında başarılı oluşturulup oluşturulmadığı kontrol edildi.

The screenshot shows the Kubernetes Container Orchestration with CCE interface. On the left, there's a sidebar for 'Huawei Cloud Exercise Account' with fields for Account (Sandbox-), Username (Sandbox-user), and Password (*****). Below it, a terminal window titled 'Terminal - root@ecs-k8s:/labfile/daemonfile' shows the command history:

```
nginx-deployment-b96f959-xnq9r 1/1 Running 0 19s
[root@ecs-k8s deployfile]# kubectl delete deployment nginx-deployment
deployment.apps "nginx-deployment" deleted
[root@ecs-k8s deployfile]# mkdir /labfile/daemonfile
[root@ecs-k8s deployfile]# cd /labfile/daemonfile/
[root@ecs-k8s daemonfile]# vim DaemonSet.yaml
[root@ecs-k8s daemonfile]# kubectl apply -f DaemonSet.yaml
daemonset.apps/nginx-daemonset created
[root@ecs-k8s daemonfile]# kubectl get daemonset --namespace=kube-system
NAME SELECTOR AGE DESIRED CURRENT READY UP-TO-DATE AVAILABLE NODE
cceaddon-npd <none> 50m 1 1 1 1 1 <none>
euler23-cceaddon-npd <none> 50m 0 0 0 0 0 <none>
everest-csi-driver <none> 50m 1 1 1 1 1 <none>
icagent <none> 50m 1 1 1 1 1 <none>
nginx-daemonset <none> 17s 1 1 1 1 1 <none>
node-local-dns <none> 50m 1 1 1 1 1 <none>
```

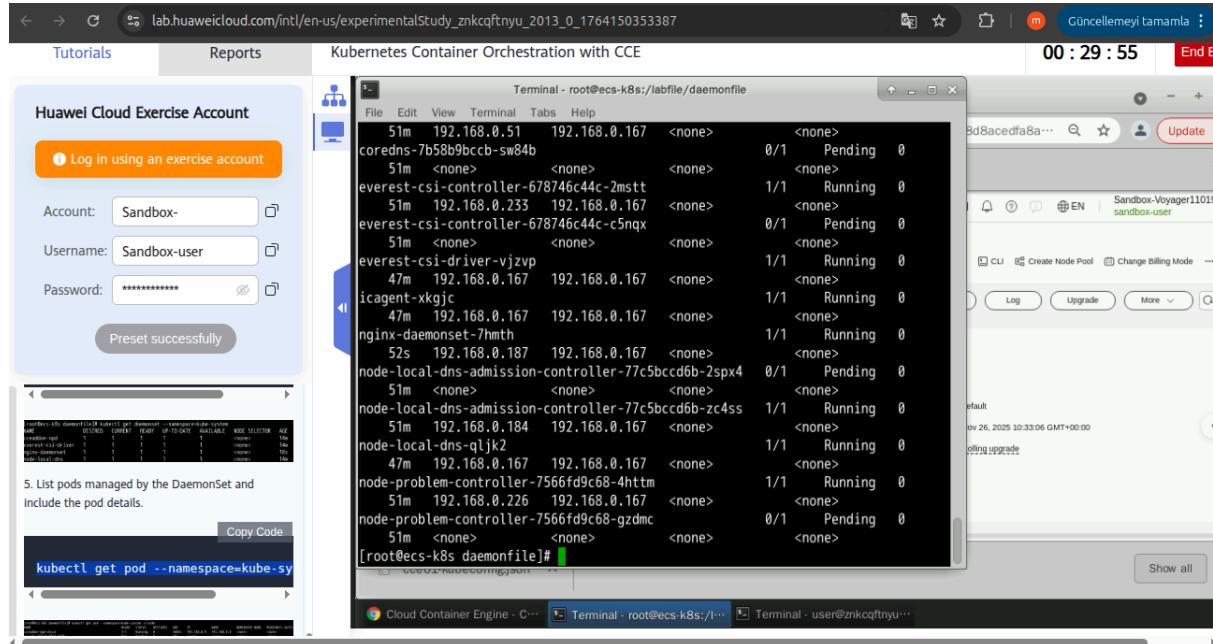
On the right, a browser window shows the Kubernetes dashboard with a log viewer for a pod named 'nginx-deployment-b96f959-897x9'. The log output includes:

```
default
Nov 26, 2025 10:33:06 GMT+00:00
nginx upgrade
```

(Görsel 4.3 : DaemonSet Kontrolü.)

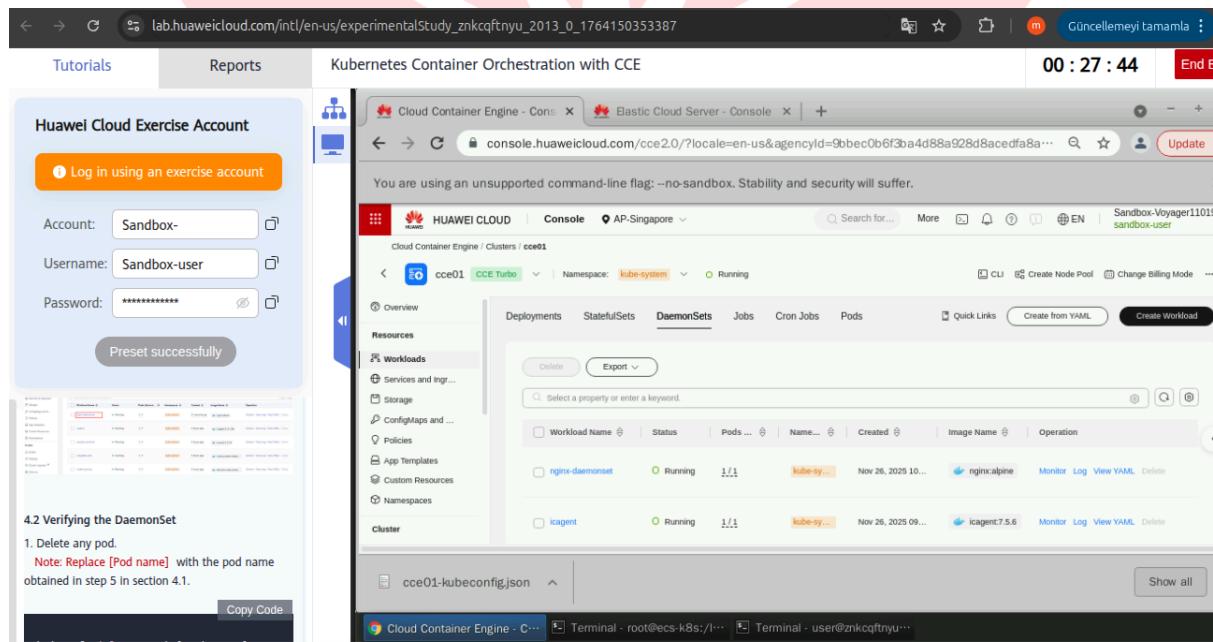
10.3) DaemonSet'in Pod'larını Görüntüleme

`kubectl get pod --namespace=kube-system -o wide` komutu ile kube-system namespace'indeki tüm pod'lar görüntülendi. Bu sayede DaemonSet'in her node üzerinde bir pod çalıştırıldığı gözlemlendi.



(Görsel 4.4 : DaemonSet Pod'ları Görüntüleme.)

Daha sonra **console** üzerinden **workloads** → **daemonset** üzerinden **namespcae:kubesystem** seçili olusturulan pod'lar görüntülenmiştir.

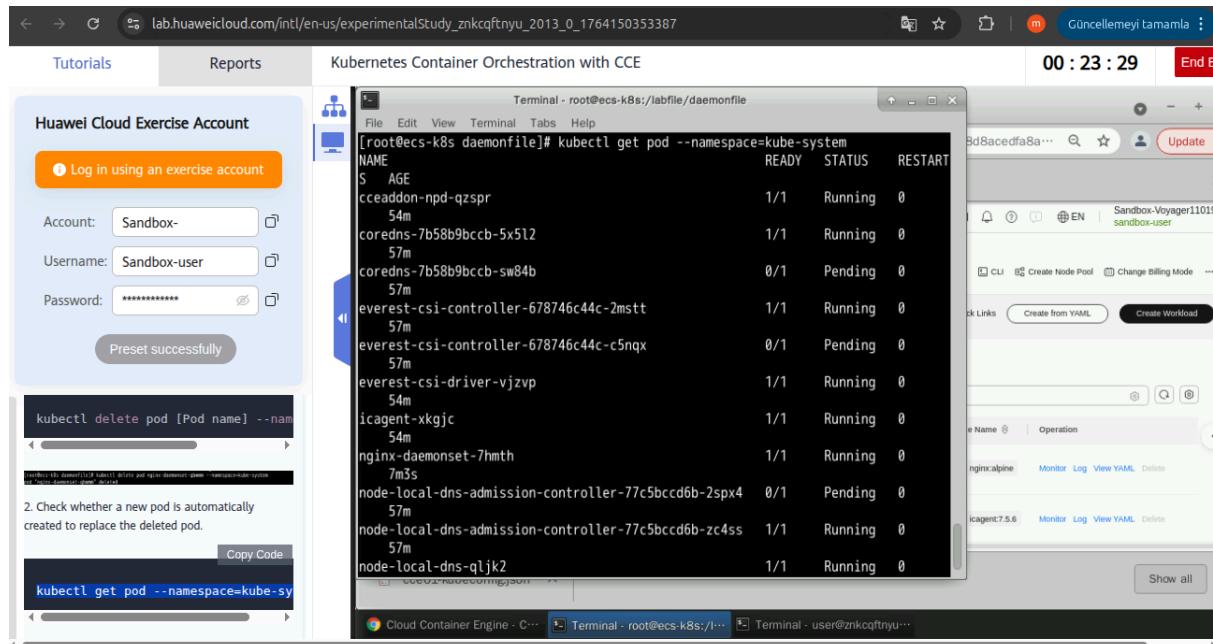


(Görsel 4.5 : Console üzerinden kontrol.)

10.4) Kubernetes Self-Healing Mekanizması

DaemonSet her node üzerinden bir pod çalıştırırmak zorundadır bu pod'lar silinse dahi Kubernetes yerlerine otomatik olarak yenilerini ekleyecektir.

kubectl delete pod nginx-daemonset-7hmth --namespace=kube-system komutu ile önce pod silinmiş daha sonra **kubectl get pod --namespace=kube-system** komutu ile kontrol edildiğinde yeni bir pod oluşturulduğu gözlemlenmiştir.



(Görsel 4.6 : Kubernetes Self-Healing Mekanizması.)