

HELM

CHART: is a helm package, it contains all of the resource definitions necessary to run an application, tool, service inside of a Kubernetes cluster. (Sistemin çalışmasını sağlayan tüm yaml dosyalarının ziplenmiş bir paket haline getirilmiş olanına denir.)

REPOSITORY: is the place where charts can be collected and shared. (chartların paylaşıldığı depo)

RELEASE: is an instance of a chart running in a Kubernetes cluster.
-One chart can often be installed many times into the same cluster
-and each time it is installed, a new release is created.(chartın clusterda hayata geçmiş/apply edilmiş haline denir.)

Helm:	Docker
-Chart	- Image
-Repository	- ECR
-Release	- Container

```
$ helm search hub
$ helm search hub wordpress
$ helm repo add bitnami https://charts.bitnami.com/bitnami -->(to
add repo-hangi repoların bulunduğunu görmek için kullanılan komut)
$ helm search repo bitnami -> bitnami reposundaki chartları görmek
için
$ helm repo ls -->localdeki helm repolarını görmek için
$ helm search repo bitnami | wc -l ->toplam chart sayısını görmek
için
$ helm pull bitnami/mysql ->bitnaminin reposundan mysql chartını
ziplenmiş olarak indirir.( zipi açmak için => tar -xvf mysql-*.tgz )
```

```
**bitnami reposundan indirdiğimiz charttan release oluşturmak için;
$ helm repo update
$ helm install mysql-release bitnami/mysql
# helm install komutunun özelliklerini documentation helm_install
sayfasından okuyabilirsiniz
***chartın install edilmesiyle bir release oluştu.
```

```
$ helm show chart bitnami/mysql
$ helm show all bitnami/mysql
```

==> - Whenever we install a chart, a new release is created. So one chart can be installed multiple times into the same cluster. And each can be independently managed and upgraded.

==>bitnami wordpress imajından/chartından bir release oluşturmak için;

```
$ helm install my-release \
  --set wordpressUsername=admin \
  --set wordpressPassword=password \
  --set mariadb.auth.rootPassword=secretpassword \
  bitnami/wordpress
```

```
$ helm list
```

==> uninstall a release

```
$ helm uninstall my-release
```

```
$ helm uninstall mysql-release
```

==> KENDİ HELM CHARTIMIZI OLUŞTURMA

```
$ helm create clarus-chart
```

#clarus-chart bizim verdiğimiz bir isim, başka bir isim de verebilirdik. Create komutu yeni bir chart oluşturur ve dosya yapısı bize hazır olarak gelir.

solumuzda görülen "chart" klasöründe bu chartın bağlı olduğu, ihtiyaç duyduğu başka chartlar bulunur.
##Template klasörü bizim manifest dosyalarımızı koyduğumuz yerdir.
#Chart yaml, chart ile ilgili genel bilgileri barındırır.
#Values yaml da terraformdaki values yaml gibi düşünülebilir.

```
$ ls clarus-chart
-configmap.yaml dosyası oluşturmak;
apiVersion: v1
kind: ConfigMap
metadata:
  name: clarus-chart-config
data:
  myvalue: "clarus-chart configmap example"
```

```
course: {{ .Values.course }}
```

```
$ helm install helm-demo clarus-chart
```

--> bu install komutuyla clarus-chart isimli helm chartımızdan helm-demo isimli bir release oluşturmuş olduk(chart 'ı çalışır hale getirmiş olduk).

```
$ helm install --debug --dry-run mydryrun clarus-chart
```

```
$ helm get manifest myvalue
```

==>>CHARTMUSEUM

ALT TAKİ KOMUTLARLA CHARTMUSEUMU KURUYORUZ !

```
```bash
```

```
curl -LO https://s3.amazonaws.com/chartmuseum/release/latest/bin/
linux/amd64/chartmuseum
```

```
chmod +x ./chartmuseum
```

```
sudo mv ./chartmuseum /usr/local/bin
```

```
chartmuseum --version
```

# ŞİMDİ BU ALT TAKİ KOMUTLA CHARTMUSEUMU ÇALIŞTIRIYORUM, 8080 PORTUNDAN YAYIN YAPSIN DİYORUM, LOCAL STORAGE KULLANSIN İSTİYORUM, DEĞİŞKENLERİ DE BULUNDUĞUM KLASÖR ALTINDA CHARTSTORAGE ALTINA KAYDETSİN DİYORUM

```
chartmuseum --debug --port=8080 \
```

```
--storage="local" \
```

```
--storage-local-rootdir="./chartstorage"
```

# ŞİMDİ MASTER EC2 İNSTANCEMİN PUBLIC İPSİNİN SONUNA 8080 KOYARAK (SECURITY GROUPSTA 8080. PORTU AÇIK OLMAYACAK ONU AÇARSIN) TARAYICIDA AÇIYORUM. Daha sonra vs coda geri döndüğünde terminali kapatma bırakalım çalışsın, yeni terminal aç oradan devam et.

# ŞİMDİ ALT TAKİ KOMUTU KULLANARAK REMOTEDEKİ REPOYU MYLOCALREPO YA ÇEKECEĞİZ.

```
$ helm repo add mylocalrepo http://<public-ip>:8080
```

```
$ helm repo ls
```

```
$ helm search repo mylocalrepo -> Find the names of the charts in
mylocalrepo
```

```
BU KOMUT CLARUS CHART DOSYASINI PAKETLER.
```

```
$ helm package clarus-chart
```

```
mkdir my-charts
```

```
mv clarus-chart-0.1.0.tgz my-charts
```

```
helm repo index my-charts --url http://<public-ip>:8080 # bu komut
chartların olduğu index fileyi oluşturuyor.
```

This is the index file of my-charts:

```
```yaml
```

```
apiVersion: v1
```

```
entries:
```

```
  clarus-chart:
```

```
    - apiVersion: v2
```

```
      appVersion: 1.16.0
```

```
      created: "2021-12-07T11:59:09.466396276+03:00"
```

```
      description: A Helm chart for Kubernetes
```

```
      digest:
```

```
712c46edcd85b167881bb644d8de4391eee9acd76aabb75fa2f6e53b  
edd1c872
```

```
      name: clarus-chart
```

```
      type: application
```

```
      urls:
```

```
        - http://<public ip>:8080/clarus-chart-0.1.0.tgz
```

```
      version: 0.1.0
```

```
generated: "2021-12-07T11:59:09.466104188+03:00"
```

```
# CHARTIMIZ HALA LOCALİMİZDE. ŞİMDİ BUNU REPOYA  
PUSHLAYACAĞIZ.
```

```
```bash
```

```
cd my-charts
```

```
curl --data-binary "@clarus-chart-0.1.0.tgz" http://<public ip>:8080/
api/charts
```

```
```
```

```
# Şimdi ip adresimiz:8080/api/charts a gidip kontrol edelim.
```

```
$ helm search repo mylocalrepo #Bu komut boş dönecektir çünkü bu  
bizim local depomuz bu. Update etmek lazım
```

```
$ helm repo update #bu komutla update yapıyoruz
```

\$ helm search repo mylocalrepo *#tekrar bunu çalıştırınca chartı göreceğiz.*

helm package clarus-chart

mv clarus-chart-0.1.1.tgz my-charts *# BU KOMUTU ÇALIŞTIRDIKTAN SONRA CD MY-CHARTS KLASÖRÜNE GİT VE LS YAP. Şimdi index.yaml fileyi güncellememiz lazım. Şimdi terminalde home directorye gidip alttaki kodu çalıştırıyoruz ve sonrasında index file güncellendi mi bakıyoruz.*

helm repo index my-charts --url http://<public-ip>:8080

ŞİMDİ ALTTAKİ KOMUTLAR İLE TGZ DOSYAMIZI DA REMOTEDEKİ CHARTS KLASÖRÜMÜZE ATIYORUZ.

```bash

cd my-charts

curl --data-binary "@clarus-chart-0.1.1.tgz" http://<public ip>:8080/api/charts

```

ŞİMDİ BÜTÜN REPOLARI GÜNCELLİYORUZ ALTTAKİ KOMUTLAR İLE

- Let's update all the repositories.

```bash

\$ helm search repo mylocalrepo

\$ helm repo update

\$ helm search repo mylocalrepo

\$ helm search repo mylocalrepo -l

*# BU KOMUTLAR İLE HELM PLUGINİNİ YÜKLÜYORUZ*

helm plugin ls

helm plugin install <https://github.com/chartmuseum/helm-push.git>

