

Kubernetes üzerine WordPress kurulumu

Helm

Kubernetes üzerine wordpress kurulumu için helm'i tercih ettim.

Helm bizim için Kubernetes ortamında çalışan uygulamalarımızın kaynaklarını (deployment, statefulset, service, ingress vb.) kolayca yönetebilmemizi ve karmaşıklıklardan kurtulmamızı sağlar.

Paket yöneticisi gibi düşünebiliriz. Ubuntu'daki apt gibi.

Helm Kurulumu:

```
curl https://baltocdn.com/helm/signing.asc | gpg --dearmor | sudo tee /usr/share/keyrings/helm.gpg > /dev/null
sudo apt-get install apt-transport-https --yes
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/helm.gpg] https://baltocdn.com/helm/stable/debian/ all main" | sudo tee /etc/apt/sources.list.d/helm-stable-debian.list
sudo apt-get update
sudo apt-get install helm
```

Helm'i kurduktan sonra sunucu veya sisteminizde docker ve k3s olması gerekiyor. Yoksa aşağıdaki gibi bir hata alabilirsiniz.

```
root@wordpress:/home/selman# helm install my-release bitnami/wordpress
Error: INSTALLATION FAILED: Kubernetes cluster unreachable: Get "http://localhost:8080/version": dial tcp 127.0.0.1:8080: connect: connection refused
root@wordpress:/home/selman#
```

yine hata alıyorsanız şu kodu deneyebilirsiniz.

```
export KUBECONFIG=/etc/rancher/k3s/k3s.yaml
```

WordPress Kurulumu:

```
helm repo add bitnami https://charts.bitnami.com/bitnami
helm install my-release bitnami/wordpress
```

Kullanıcı adı,şifreyi ve bazı özellikleri biz kurmak istersek şöyle yapabiliriz.

```
helm install my-release \
  --set wordpressUsername=selman \
  --set wordpressPassword=parlak \
  --set wordpressEmail=selman@example.com \
  --set wordpressBlogName=SelmanBlog \
  --set mariadb.auth.rootPassword=secretpassword \
  bitnami/wordpress
```

Yukarıdaki komut, WordPress yönetici hesabı kullanıcı adını ve şifresini sırasıyla admin ve şifre olarak ayarlar. Ayrıca, MariaDB kök kullanıcı parolasını secretpassword olarak ayarlar.

Not:

```
root@wordpress:/home/selman# helm install my-release \
> --set wordpressUsername=selman \
> --set wordpressPassword=parlak \
> --set wordpressBlogName=Selmans Blog \
> --set mariadb.auth.rootPassword=secretpassword \
> bitnami/wordpress
Error: INSTALLATION FAILED: expected at most two arguments, unexpected arguments: bitnami/wordpress
```

Selman's Blog'unda boşluk olduğu için 2 argüman olarak alır ve hata verir. yaml dosyasında yapmak her zaman daha garantili olabilir.

yaml dosyasıyla da wordpress oluşturabiliriz.

wordpress.yaml

```
image:
  registry: docker.io
  repository: bitnami/wordpress
  tag: 5.3.2-debian-10-r32
  pullPolicy: IfNotPresent
  debug: false
wordpressUsername: selman
wordpressPassword: parlak
wordpressEmail: selman@example.com
wordpressFirstName: selman
wordpressLastName: parlak
wordpressBlogName: Selman's Blog!
```

```
wordpressTablePrefix: wp_
wordpressScheme: http
wordpressSkipInstall: false
service:
  type: LoadBalancer
  port: 80
  httpsPort: 443
  httpsTargetPort: https
  metricsPort: 9117
  nodePorts:
    http: ""
    https: ""
    metrics: ""
  externalTrafficPolicy: Cluster
  annotations: {}
  loadBalancerSourceRanges: []
```

```
helm install my-release -f wordpress.yaml bitnami/wordpress
```

Helm parametreleri :

`image.registry` : Wordpress image dosyası kaydını nereden alacağımızı söyler.Biz kaynak olarak docker'ı gösterdik.

`image.repository` : Docker image yolu

`image.tag` : image versiyonu gibi düşünebiliriz.

`image.debug` : Hata ayıklama yapıp yapmadığını bool değerlerle kontrol edebiliriz. varsayılan false gelir.

`wordpressUsername` : WordPress kullanıcı adı

`wordpressPassword` : WordPress kullanıcı şifresi

`wordpressEmail` : WordPress kullanıcı e-postası

`wordpressFirstName` : WordPress kullanıcı adı

`wordpressLastName` : WordPress kullanıcı soyadı

`wordpressBlogName` : Blog adı

`wordpressTablePrefix` : WordPress veritabanı tabloları için kullanılacak önek

`wordpressScheme` : WordPress URL'leri oluşturmak için kullanılacak şema. Varsayılan http

`wordpressSkipInstall` : Kurulum sihirbazını atlar. varsayılan false gelir.

`service.type` : WordPress servis tipi varsayılan LoadBalancer.

`service.ports.http`:WordPress servis HTTP bağlantı noktası. varsayılan port 80 gelir.

`service.ports.https`: WordPress servis HTTPS bağlantı noktası. varsayılan 443 gelir.

`service.httpsTargetPort`: HTTPS için hedef bağlantı noktası

`service.externalTrafficPolicy`: WordPress servis dış trafik politikası. varsayılan cluster gelir.

Yüklendikten sonra çıktı bu şekilde olacaktır.

```
NAME: my-release
LAST DEPLOYED: Sun Sep  4 11:42:34 2022
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
CHART NAME: wordpress
CHART VERSION: 15.1.5
APP VERSION: 6.0.2

** Please be patient while the chart is being deployed **

Your WordPress site can be accessed through the following DNS name from within your cluster:

    my-release-wordpress.default.svc.cluster.local (port 80)

To access your WordPress site from outside the cluster follow the steps below:

1. Get the WordPress URL by running these commands:

    NOTE: It may take a few minutes for the LoadBalancer IP to be available.
    Watch the status with: 'kubectl get svc --namespace default -w my-release-wordpress'

    export SERVICE_IP=$(kubectl get svc --namespace default my-release-wordpress --include "{{ range (index .status.loadBalancer.ingress 0) }}{{ . }}{{ end }}" )
    echo "WordPress URL: http://$SERVICE_IP/"
    echo "WordPress Admin URL: http://$SERVICE_IP/admin"

2. Open a browser and access WordPress using the obtained URL.

3. Login with the following credentials below to see your blog:

    echo Username: selman
    echo Password: $(kubectl get secret --namespace default my-release-wordpress -o jsonpath="{.data.wordpress-password}" | base64 -d)
```

Wordpress bloğumuza bağlanabilmek için:

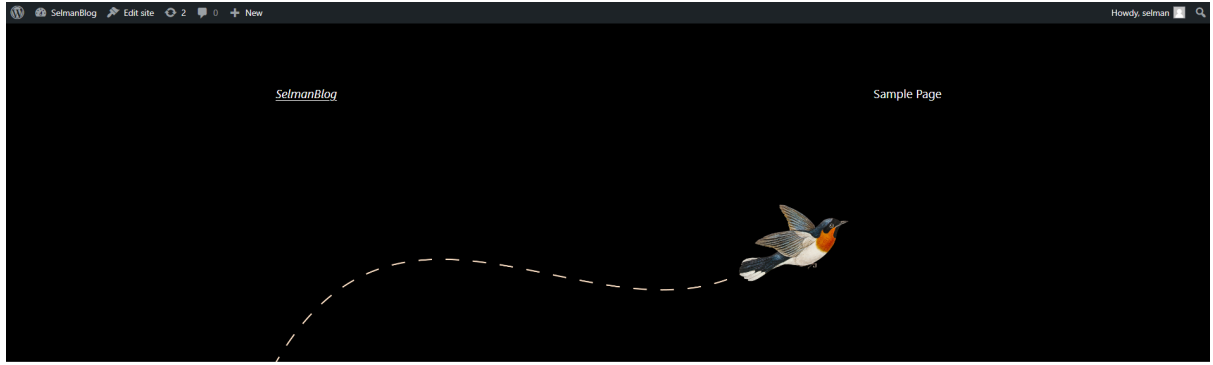
wordpressden oluşan servislerimize bakıyoruz.

`service.type` varsayılan olarak Loadbalancer oluşturur.

```
kubectl get svc
```

```
root@wordpress:/home/selman# kubectl get svc
NAME                TYPE                CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
kubernetes          ClusterIP           10.43.0.1        <none>            443/TCP          85m
my-release-mariadb   ClusterIP           10.43.29.102     <none>            3306/TCP          83m
my-release-wordpress LoadBalancer        10.43.121.122    <pending>        80:30451/TCP,443:31223/TCP 83m
```

localhost:port ile bağlanabilirsiniz. benim portum 80:30451 80 burada localhostu ifade eder.



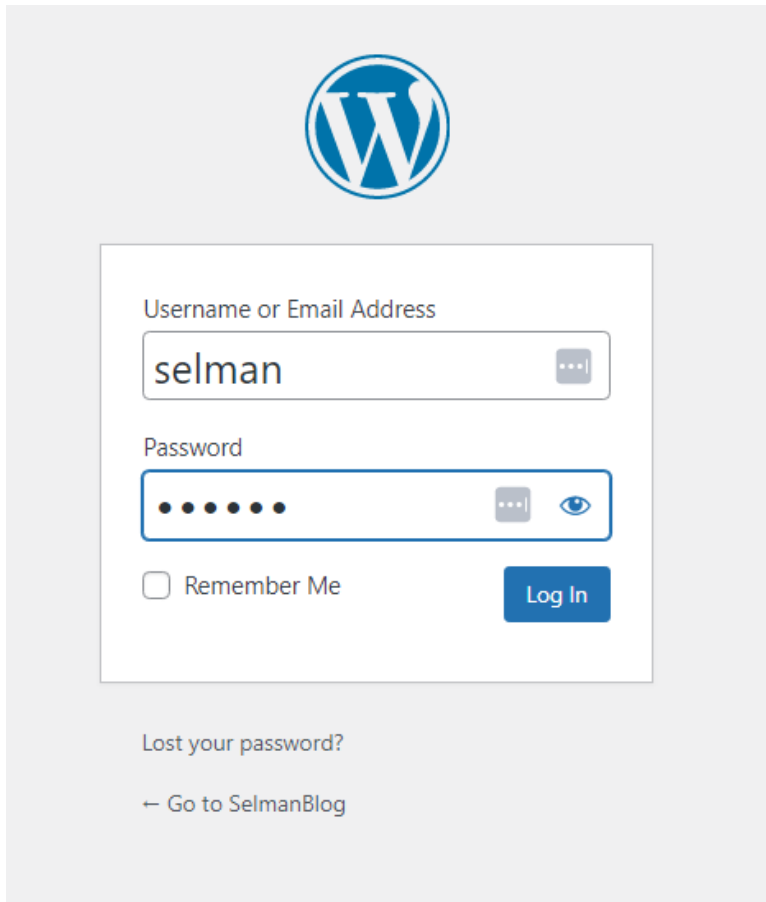
Welcome

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

September 4, 2022

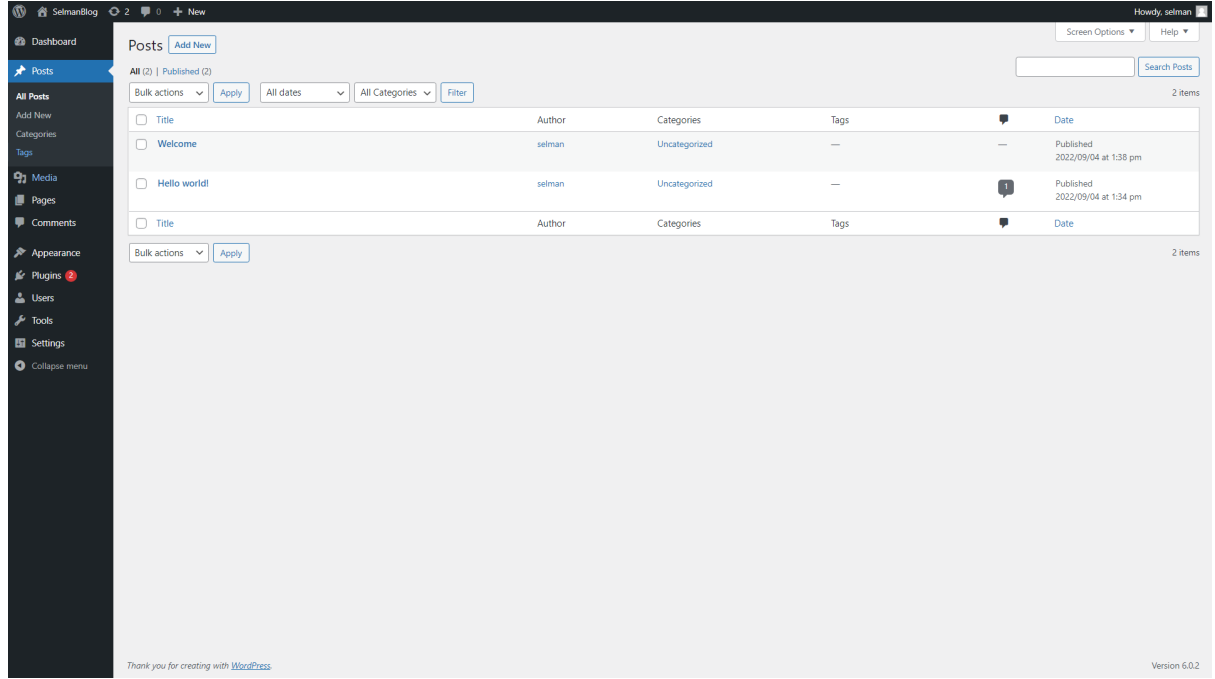
Hello world!

admin paneline ulaşmak isterseniz localhost:port/wp-admin



yukarıda anlattığım set ile veya yaml ile verdiğiniz `wordpressUsername` ve `wordpressPassword` ' u burada girebilirsiniz.

ve sonuç :



google kubernetes-cluster'da oluşturduğum wordpressten giriş yapmak isterseniz

<http://34.89.132.191/>