[Hands-on] 15. Helm

자주 사용되는 Helm 명령어들을 실습해 보겠습니다.

먼저 어떤 명령어들이 있는지 살펴볼까요?

```
ubuntu@ip-10-0-1-161:~$ helm --help
The Kubernetes package manager

Common actions for Helm:

- helm search:    search for charts
- helm pull:      download a chart to your local directory to view
- helm install:   upload the chart to Kubernetes
- helm list:      list releases of charts

Environment variables:

| Name                                | Description                                                                                |
|-------------------------------------|--------------------------------------------------------------------------------------------|
| $HELM_CACHE_HOME                    | set an alternative location for storing cached files.                                      |
| $HELM_CONFIG_HOME                   | set an alternative location for storing Helm configuration.                                |
| $HELM_DATA_HOME                     | set an alternative location for storing Helm data.                                         |
| $HELM_DEBUG                         | indicate whether or not Helm is running in Debug mode                                      |
| $HELM_DRIVER                        | set the backend storage driver. Values are: configmap, secret, memory, sql.                |
| $HELM_DRIVER_SQL_CONNECTION_STRING | set the connection string the SQL storage driver should use.                               |
| $HELM_MAX_HISTORY                   | set the maximum number of helm release history.                                            |
| $HELM_NAMESPACE                     | set the namespace used for the helm operations.                                            |
| $HELM_NO_PLUGINS                    | disable plugins. Set HELM_NO_PLUGINS=1 to disable plugins.                                 |
| $HELM_PLUGINS                       | set the path to the plugins directory                                                      |
...생략...
```

> **명령어 :** `helm --help`

이제 설치(install)를 한 번 진행해볼까요?

먼저 Repository를 add해줍니다.

```
ubuntu@ip-10-0-1-161:~$ helm repo add bitnami https://charts.bitnami.com/bitnami
"bitnami" has been added to your repositories
```

> **명령어 :** `helm repo add bitnami https://charts.bitnami.com/bitnami`

Repository 목록도 볼 수 있습니다.

```
ubuntu@ip-10-0-1-161:~$ helm repo list
NAME    URL
bitnami https://charts.bitnami.com/bitnami
```

> **명령어 :** `helm repo list`

검색도 가능하구요.

```
ubuntu@ip-10-0-1-161:~$ helm search repo bitnami
NAME                                    CHART VERSION   APP VERSION     DESCRIPTION
bitnami/airflow                         12.5.12         2.3.2           Apache Airflow is a tool to express and execute...
bitnami/apache                          9.1.13          2.4.54          Apache HTTP Server is an open-source HTTP serve...
bitnami/argo-cd                         3.4.4           2.4.3           Argo CD is a continuous delivery tool for Kuber...
bitnami/argo-workflows                  2.3.5           3.3.8           Argo Workflows is meant to orchestrate Kubernet...
bitnami/aspnet-core                     3.4.10          6.0.6           ASP.NET Core is an open-source framework for we...
bitnami/cassandra                       9.2.7           4.0.4           Apache Cassandra is an open source distributed ...
bitnami/cert-manager                    0.7.1           1.8.2           Cert Manager is a Kubernetes add-on to automate...
bitnami/common                          1.16.0          1.16.0          A Library Helm Chart for grouping common logic ...
bitnami/concourse                       1.3.7           7.8.1           Concourse is an automation system written in Go...
bitnami/consul                          10.7.3          1.12.2          HashiCorp Consul is a tool for discovering and ...
bitnami/contour                         8.0.4           1.21.1          Contour is an open source Kubernetes ingress co...
bitnami/contour-operator                1.2.1           1.20.1          The Contour Operator extends the Kubernetes API...
...생략...
```

> **명령어 :** `helm search repo bitnami`

Wordpress를 한 번 찾아볼까요?

```
ubuntu@ip-10-0-1-161:~$ helm search repo wordpress
NAME                    CHART VERSION   APP VERSION     DESCRIPTION
bitnami/wordpress       15.0.7          6.0.0           WordPress is the world's most popular blogging ...
bitnami/wordpress-intel 2.0.7           6.0.0           WordPress for Intel is the most popular bloggin...
```

> **명령어 :** `helm search repo wordpress`

이제 설치를 진행해 보겠습니다.

```
ubuntu@ip-10-0-1-161:~$ helm repo update
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "bitnami" chart repository
Update Complete. ⎈Happy Helming!⎈
```

> **명령어 :** `helm repo update`

```
ubuntu@ip-10-0-1-161:~$ helm install my-wordpress bitnami/wordpress
NAME: my-wordpress
LAST DEPLOYED: Thu Jul  7 16:28:16 2022
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
CHART NAME: wordpress
CHART VERSION: 15.0.7
APP VERSION: 6.0.0

** 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-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:
```

앞장에서 계속

```
  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-wordpress'

   export SERVICE_IP=$(kubectl get svc --namespace default my-wordpress --include "{{ range (index .status.loadBalancer.ingress 0) }}{{ . }}{{ end }}")
   echo "WordPress URL: http://$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: user
  echo Password: $(kubectl get secret --namespace default my-wordpress -o jsonpath="{.data.wordpress-password}" | base64 -d)
```

> **명령어 :** `helm repo update` , `helm install my-wordpress bitnami/wordpress`

설치된 Helm chart는 **Release**라고 합니다.
Release의 목록은 다음 명령으로 조회할 수 있구요.

```
ubuntu@ip-10-0-1-161:~$ helm list
NAME            NAMESPACE       REVISION        UPDATED                                 STATUS          CHART                   APP VERSION
my-wordpress    default         1               2022-07-07 16:28:16.316510487 +0000 UTC deployed        wordpress-15.0.7        6.0.0
```

> **명령어 :** `helm list`

쿠버네티스 명령어로 어떤 리소스들이 생성됐나 볼까요?

```
ubuntu@ip-10-0-1-161:~$ kubectl get all
NAME                                 READY   STATUS    RESTARTS   AGE
pod/my-wordpress-56bff78c5d-szzbg    1/1     Running   0          3m40s
pod/my-wordpress-mariadb-0           1/1     Running   0          3m39s

NAME                            TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)                      AGE
service/kubernetes              ClusterIP      10.96.0.1       <none>        443/TCP                      3d5h
service/my-wordpress            LoadBalancer   10.103.84.122   <pending>     80:30798/TCP,443:31520/TCP   3m40s
service/my-wordpress-mariadb    ClusterIP      10.110.56.55    <none>        3306/TCP                     3m40s

NAME                            READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/my-wordpress    1/1     1            1           3m40s

NAME                                       DESIRED   CURRENT   READY   AGE
replicaset.apps/my-wordpress-56bff78c5d    1         1         1       3m40s

NAME                                     READY   AGE
statefulset.apps/my-wordpress-mariadb    1/1     3m40s
```

> **명령어** : `kubectl get all`

와우~ 뭔가 Wordpress 소프트웨어에 필요한 모든게 한 번에 설치가 된 것 같네요. 패키지로...

이게 바로 **Helm** 이랍니다.

chart를 다운로드(pull)도 해볼게요.

```
ubuntu@ip-10-0-1-161:~$ helm pull bitnami/wordpress --version 15.0.7
ubuntu@ip-10-0-1-161:~$ ls wordpress*
wordpress-15.0.7.tgz
```

> **명령어 :** `helm pull bitnami/wordpress --version 15.0.7`

tar 파일로 받아지네요.

압축도 풀어볼까요?

```
ubuntu@ip-10-0-1-161:~$ tar -xvf wordpress-15.0.7.tgz
wordpress/Chart.yaml
wordpress/Chart.lock
wordpress/values.yaml
wordpress/values.schema.json
wordpress/templates/NOTES.txt
wordpress/templates/_helpers.tpl
wordpress/templates/config-secret.yaml
wordpress/templates/deployment.yaml
wordpress/templates/externaldb-secrets.yaml
wordpress/templates/extra-list.yaml
...생략...
```

> **명령어 :** `tar -xvf wordpress-15.0.7.tgz`

어떤 파일들이 있는지 한 번 살펴보겠습니다.

```
ubuntu@ip-10-0-1-161:~$ tree ./wordpress
./wordpress
├── Chart.lock
├── Chart.yaml
├── README.md
├── charts
│   ├── common
│   │   ├── Chart.yaml
│   │   ├── README.md
│   │   ├── templates
│   │   │   ├── _affinities.tpl
│   │   │   ├── _capabilities.tpl
│   │   │   ├── _errors.tpl
│   │   │   ├── _images.tpl
│   │   │   ├── _ingress.tpl
│   │   │   ├── _labels.tpl
│   │   │   ├── _names.tpl
```

명령어 : `tree ./wordpress`

여기까지 Helm 에 대해 알아보았습니다.

수고하셨습니다. (〃・ᢆ‿・ᢆ)ゞ