

Contents

- Helm

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.
...생략...
```

명령어 : `helm --help`

Docker & Kubernetes - [Hands-on] 15. Helm

Common actions for Helm 의 명령어들을 하나씩 해볼까요?

첫 번째는 `helm search`인데, 그 전에 `helm repository`를 먼저 추가(add)해줘야 합니다.

```
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`

Docker & Kubernetes - [Hands-on] 15. Helm

이제 검색(`helm search`) 가능합니다.

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?

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`

Docker & Kubernetes - [Hands-on] 15. Helm

다음은 `helm pull` 명령어 입니다.

Helm repository에 등록되어 있는 Helm chart를 다운로드(pull)하는 명령어 입니다.

```
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/extralist.yaml  
...생략...
```

명령어 : `tar -xvf wordpress-15.0.7.tgz`

Docker & Kubernetes - [Hands-on] 15. Helm

어떤 파일들이 있는지 한 번 살펴보겠습니다.

```
ubuntu@ip-10-0-1-161:~/wp$ 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`

- `tree` 명령어는 리눅스에서 디렉토리/파일구조를 볼 수 있는 명령어입니다. 혹시 안되면 아래와 같이 `tree`를 설치하고 해주세요.

명령어 : `sudo apt-get update`

명령어 : `sudo apt-get install tree`

Docker & Kubernetes - [Hands-on] 15. Helm

이제 설치(`helm install`)를 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:
```

Docker & Kubernetes - [Hands-on] 15. Helm

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 install my-wordpress bitnami/wordpress`

설치된 Helm chart는 Release라고 합니다.

Release의 목록은 `helm list` 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`

Docker & Kubernetes - [Hands-on] 15. Helm

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...

삭제도 한 번에 가능합니다.

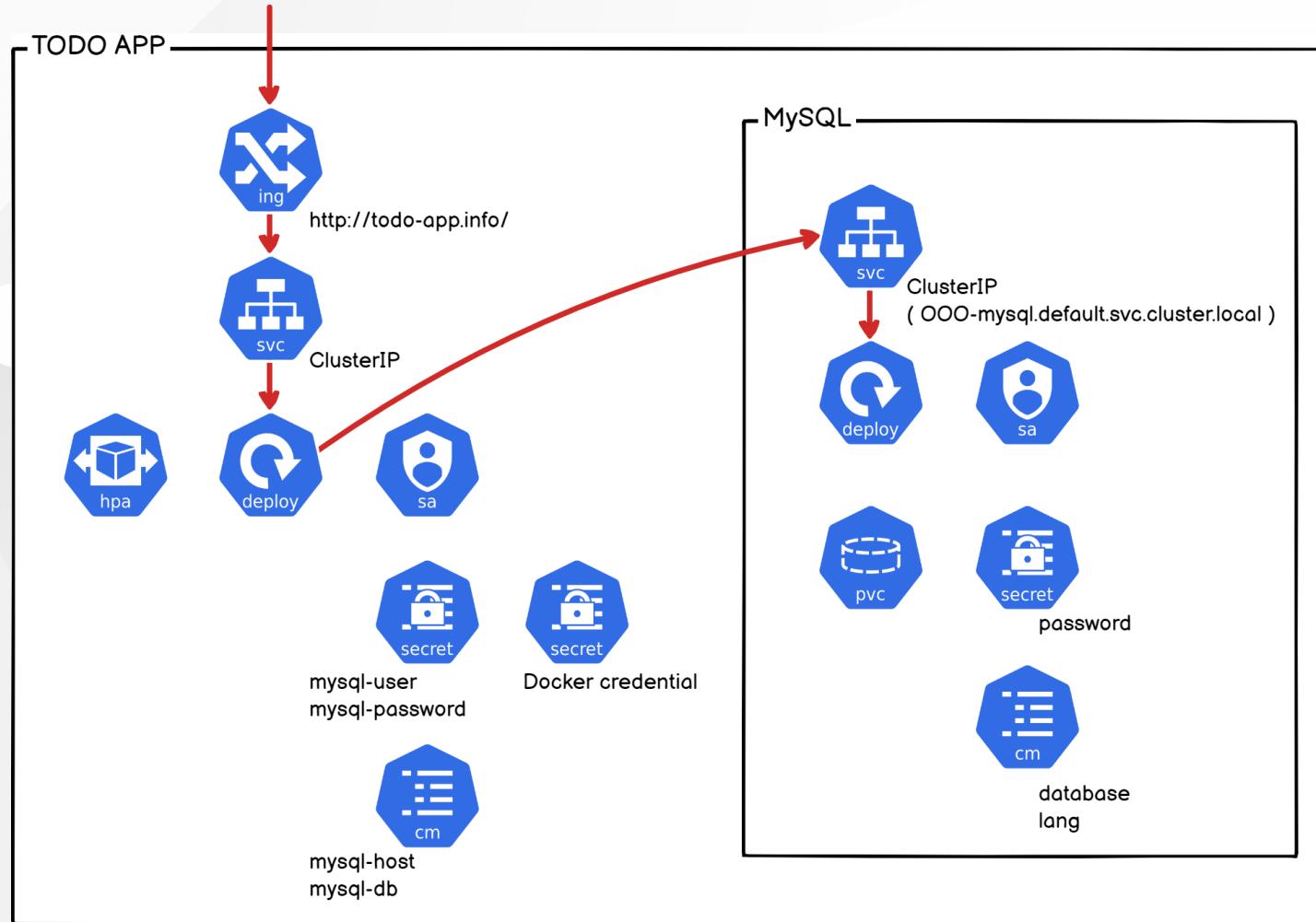
```
ubuntu@ip-10-0-1-161:~$ helm uninstall my-wordpress
release "my-wordpress" uninstalled
```

명령어 : `kubectl uninstall my-wordpress`

Docker & Kubernetes - [Hands-on] 15. Helm

우리가 익숙한 ToDo App을 이용해서 좀 더 자세히 볼게요.

차트는 아래와 같은 구조를 가지고 있습니다. 우리가 배운 여러가지가 다 들어있네요.



Docker & Kubernetes - [Hands-on] 15. Helm

설치는 간단합니다. 명령어 하나면 끝.

```
ubuntu@ip-10-0-1-161:~$ helm install my-todo-app https://github.com/JungSangup/mspt3/raw/main/hands_on_files/todo-app-1.0.0.tgz
NAME: my-todo-app
LAST DEPLOYED: Tue Feb 7 05:24:08 2023
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
1. Get the application URL by running these commands:
  http://todo-app.info/
```

명령어 : `helm install my-todo-app https://github.com/JungSangup/mspt3/raw/main/hands_on_files/todo-app-1.0.0.tgz`

위의 방법은 Helm chart 패키지 파일의 URL(깃헙에 올려놓은 파일)을 직접 지정해서 설치한 것입니다.

위의 방법 외에도 아래와 같은 다양한 방법으로 설치 가능합니다.

명령어 : `helm install my-todo-app ./todo-app-1.0.0.tgz` -> 로컬 경로의 tgz파일(패키징 된 Helm chart)

명령어 : `helm install my-todo-app ./todo-app` -> 로컬 경로의 차트 디렉토리

`hands_on_files` 아래에 위의 두 가지 방법을 위한 파일/디렉토리도 준비해 놓았습니다.

우선 이 Helm release는 Uninstall을 할게요. 뒤에 다른 방법으로 다시 설치하겠습니다.

```
ubuntu@ip-10-0-1-161:~$ helm uninstall my-todo-app
release "my-todo-app" uninstalled
```

명령어 : `helm uninstall my-todo-app`

Docker & Kubernetes - [Hands-on] 15. Helm

이번에는 구성을 조금 달리해서 설치하겠습니다.

여러분의 Docker private repository에 올려놓은 이미지를 사용하도록 하고, 이미지 pull을 위해서 자격증명을 사용하도록 할게요.

역시 아래와 같이 간단하게 실행할 수 있습니다.

```
ubuntu@ip-10-0-1-161:~$ helm install my-todo-app --set image.repository=rogallo/todo-app --set imageCredentials.create=true --set imageCredentials.username=rogallo --set imageCredentials.password=XXX https://github.com/JungSangup/mspt3/raw/main/hands_on_files/todo-app-1.0.0.tgz
NAME: my-todo-app
LAST DEPLOYED: Tue Feb 7 09:50:25 2023
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
1. Get the application URL by running these commands:
  http://todo-app.info/
```

명령어 :

```
helm install my-todo-app --set image.repository=[USER-NAME]/todo-app --set imageCredentials.create=true --set imageCredentials.username=[USER-NAME] --set imageCredentials.password=[PASSWORD] https://github.com/JungSangup/mspt3/raw/main/hands_on_files/todo-app-1.0.0.tgz
```

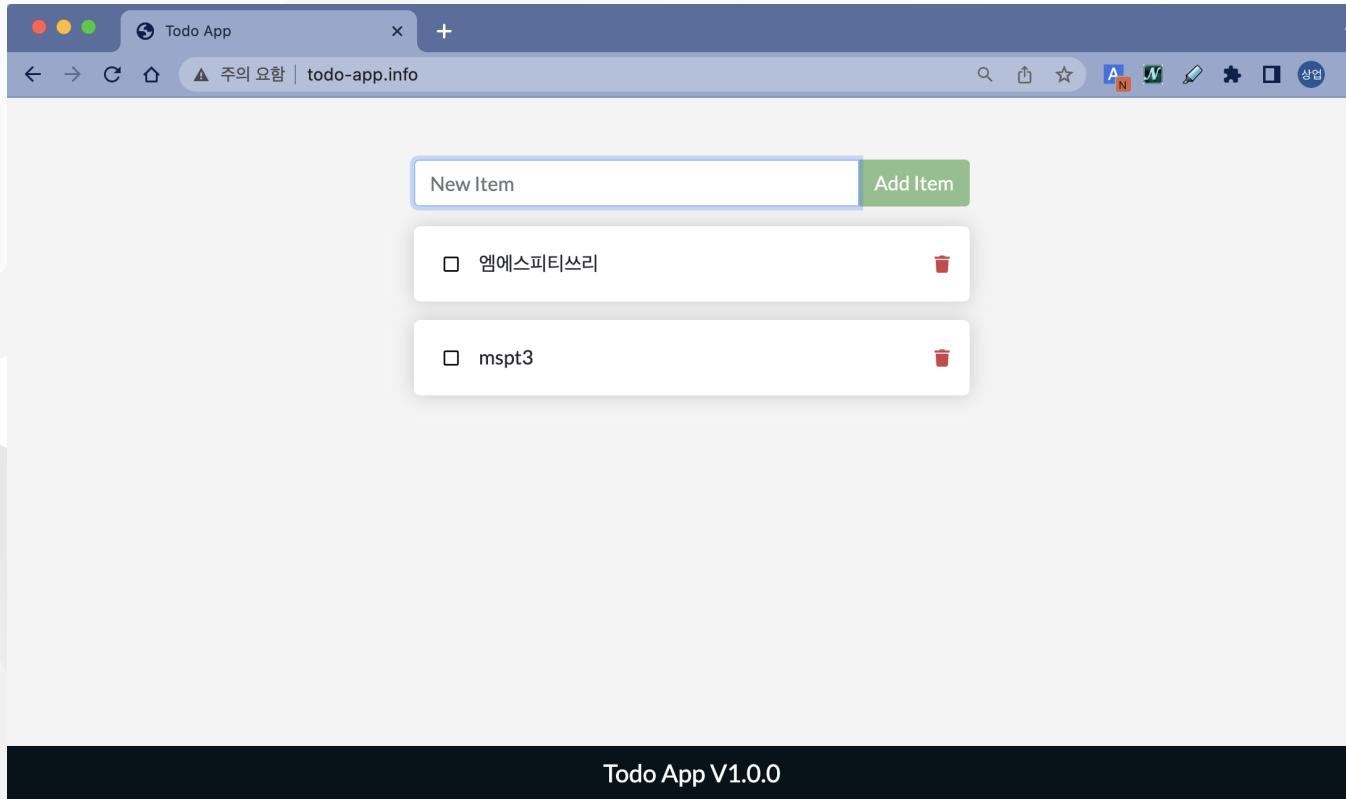
[USER-NAME]과 [PASSWORD]는 여러분의 정보로 채워넣어 주세요.

설치 시점에 아래 키-값 들을 변경해서 적용한 것입니다.

- image.repository=[USER-NAME]/todo-app
- imageCredentials.create=true
- imageCredentials.username=[USER-NAME]
- imageCredentials.password=[PASSWORD]

Docker & Kubernetes - [Hands-on] 15. Helm

브라우저에서 <http://todo-app.info/> 로 접속해서 테스트도 해보세요.



잘 되나요?

Docker & Kubernetes - [Hands-on] 15. Helm

생성된 K8s 리소스들도 확인해보세요.

```
ubuntu@ip-10-0-1-161:~$ kubectl get all
NAME                                         READY   STATUS    RESTARTS   AGE
pod/my-todo-app-6b8b4887d5-66d4b           1/1     Running   0          3m4s
pod/my-todo-app-mysql-7d8c985b5-6v251      1/1     Running   0          3m4s

NAME                TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes  ClusterIP  10.96.0.1   <none>       443/TCP   5d5h
service/my-todo-app  ClusterIP  10.101.188.213 <none>       3000/TCP  3m4s
service/my-todo-app-mysql  ClusterIP  10.101.187.212 <none>       3306/TCP  3m4s

NAME              READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/my-todo-app    1/1     1           1           3m4s
deployment.apps/my-todo-app-mysql  1/1     1           1           3m4s

NAME              DESIRED  CURRENT  READY   AGE
replicaset.apps/my-todo-app-6b8b4887d5  1        1        1      3m4s
replicaset.apps/my-todo-app-mysql-7d8c985b5  1        1        1      3m4s

NAME                  REFERENCE   TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
horizontalpodautoscaler.autoscaling/my-todo-app  Deployment/my-todo-app  <unknown>/80%  1          10         1          3m4s
```

명령어 : `kubectl get all`

Docker & Kubernetes - [Hands-on] 15. Helm

private repository의 이미지를 pull 하기 위해서 자격증명도 secret으로 생성했습니다.

```
ubuntu@ip-10-0-1-161:~$ kubectl describe secrets regcred
Name:          regcred
Namespace:     default
Labels:        app.kubernetes.io/managed-by=Helm
Annotations:   meta.helm.sh/release-name: my-todo-app
               meta.helm.sh/release-namespace: default
Type:         kubernetes.io/dockerconfigjson

Data
=====
.dockerconfigjson: 135 bytes
```

명령어 : `kubectl describe secrets regcred`

다른 리소스들 (ConfigMap, Secret, PVC, PV, Ingress) 도 한 번 확인해보세요.

Docker & Kubernetes - [Hands-on] 15. Helm

이제 Helm에서 업그레이드를 해볼게요.

여러가지 업그레이드가 있겠지만, 간단히 이미지의 Tag를 변경하는 경우만 해보겠습니다.

```
ubuntu@ip-10-0-1-161:~$ helm upgrade --set image.tag=2.0.0 my-todo-app https://github.com/JungSangup/mspt3/raw/main/hands_on_files/todo-app-1.0.0.tgz
Release "my-todo-app" has been upgraded. Happy Helming!
NAME: my-todo-app
LAST DEPLOYED: Tue Feb  7 10:14:33 2023
NAMESPACE: default
STATUS: deployed
REVISION: 2
TEST SUITE: None
NOTES:
1. Get the application URL by running these commands:
  http://todo-app.info/
```

명령어 : `helm upgrade --set image.tag=2.0.0 my-todo-app https://github.com/JungSangup/mspt3/raw/main/hands_on_files/todo-app-1.0.0.tgz`
image.tag만 변경해서 새로운 버전으로 업그레이드 합니다.

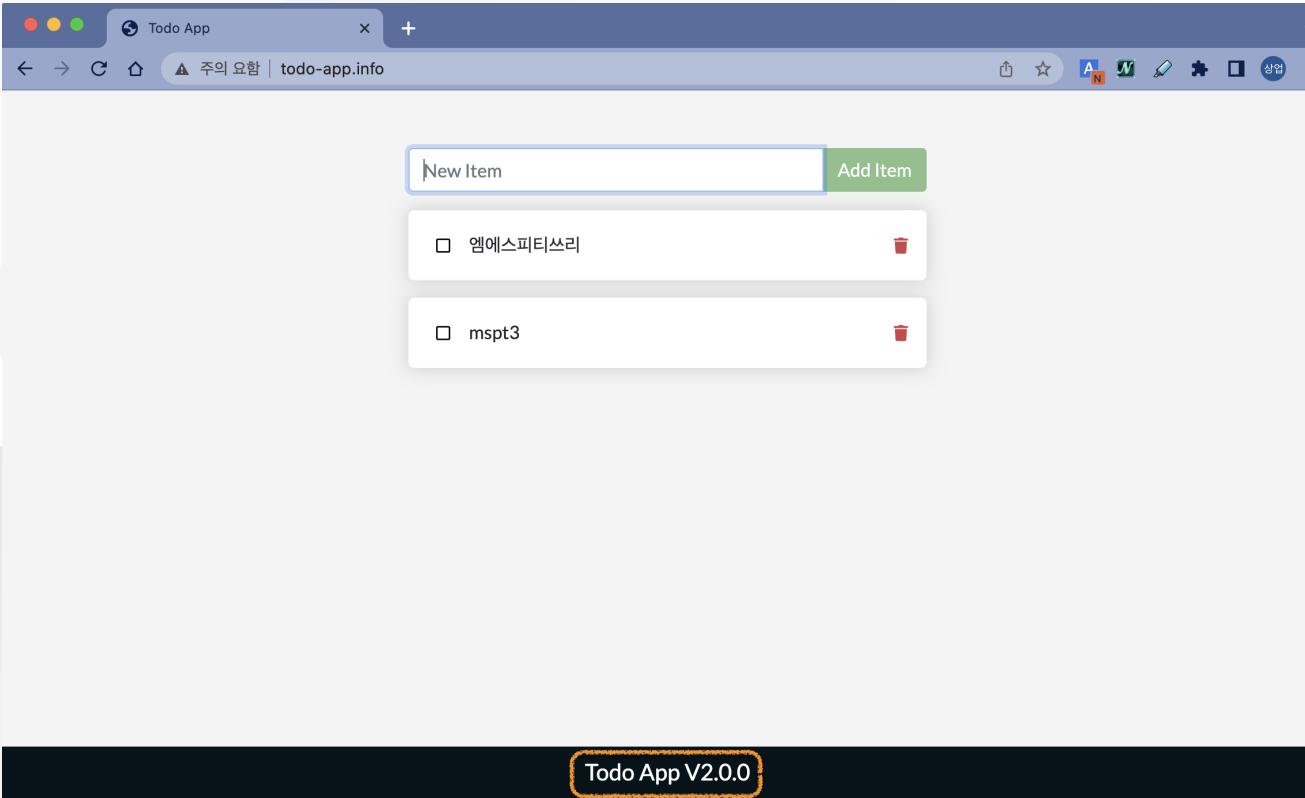
바뀐 Deployment도 확인해 보시구요.
이렇게요.

```
ubuntu@ip-10-0-1-161:~$ kubectl describe deployments my-todo-app | grep Image
  Image:      rogallo/todo-app:2.0.0
```

명령어 : `kubectl describe deployments my-todo-app | grep Image`

Docker & Kubernetes - [Hands-on] 15. Helm

브라우저에서 <http://todo-app.info/>로 접속해서 업그레이드 결과도 보세요.



구분하기 위해서 하단에 버전을 표시하도록 해 놓았습니다.

Docker & Kubernetes - [Hands-on] 15. Helm

롤백도 해볼까요?

간단히 History를 조회하고, 원하는 Revision으로 돌아가면 됩니다.

```
ubuntu@ip-10-0-1-161:~$ helm history my-todo-app
REVISION      UPDATED             STATUS        CHART          APP VERSION   DESCRIPTION
1            Tue Feb  7 09:50:25 2023  superseded  todo-app-1.0.0  1.0.0       Install complete
2            Tue Feb  7 10:14:33 2023  deployed    todo-app-1.0.0  1.0.0       Upgrade complete
ubuntu@ip-10-0-1-161:~$ helm rollback my-todo-app 1
Rollback was a success! Happy Helming!
ubuntu@ip-172-31-20-30:~/mspt3$ kubectl describe deployments my-todo-app | grep Image
  Image:      rogallo/todo-app:1.0.0
```

명령어 : `helm history my-todo-app`

명령어 : `helm rollback my-todo-app 1`

명령어 : `kubectl describe deployments my-todo-app | grep Image`

<http://todo-app.info/> 화면도 확인 해 보시구요.

Docker & Kubernetes - [Hands-on] 15. Helm

역시 마지막은 정리.

아래 명령어로 삭제(Uninstall) 해 주세요.

```
ubuntu@ip-10-0-1-161:~$ helm uninstall my-todo-app  
release "my-todo-app" uninstalled
```

명령어 : `helm uninstall my-todo-app`

한꺼번에 설치(install), 업데이트(update), 롤백(rollback), 삭제(uninstall)되니 편하네요.

여기까지 Helm 에 대해 알아보았습니다.

수고하셨습니다. (〃・_・〃) ♪