

# Section 12: (2025 Updates) Helm Basics

- Helm: kubernetes package manager로, 복잡한 애플리케이션 배포를 간소화함
- `helm install wordpress` : 전체 앱을 설치, 세부 사항을 신경 쓸 필요 없이 필요한 모든 object를 자동으로 Kubernetes에 추가함
- `helm upgrade wordpress` : 전체 앱을 업그레이드, 각각의 object가 변경해야 하는 사항을 추적, 파악함
- `helm rollback wordpress` : 이전 리비전으로 롤백 가능

## 269. A quick note about Helm2 vs Helm3

- Helm 2와 비교해서 Helm 3는
  - Tiller가 제거됨
  - three-way strategic merge patch 기능이 추가됨
    - 기존 차트, 새 차트, 그리고 클러스터의 현재 상태를 비교하여 더 똑똑한 업그레이드/롤백 가능
  - 수동으로 변경된 내용도 감지하고 보존할 수 있게 됨

## 270. Helm Components

- Chart: 쿠버네티스 클러스터에서 필요한 object 를 생성하기 위한 모든 지침을 포함한 파일의 모음
- Release: 차트를 사용하여 설치된 애플리케이션의 인스턴스
- Revision: application의 스냅샷
- Repository: docker hub와 같이 차트를 저장하는 저장소
- Metadata: 설치한 release, 사용된 차트, 리비전 상태 등과 같이 클러스터에서 수행한 작업을 추적하기 위해 헬름이 이런 데이터를 저장할 장소

## 271. Helm Charts

- chart.yaml: 차트의 기본 정보 (api version, name, description 등)
- values.yaml: 사용자가 차트에 제공할 수 있는 매개변수와 그 기본값들을 정의하는 파일
- templates/: YAML 파일들을 담고 있는 directory. 이 위치의 파일들은 실제 쿠버네티스 리소스를 생성하기 위한 템플릿 역할을 함
- charts/: 차트의 의존성으로 추가된 다른 차트들이 위치하는 곳

## 272. Working with Helm: basics

- `helm search hub wordpress` : [artifacthub.io](https://artifacthub.io) 내에서 차트 검색
- `helm search repo wordpress` : 특정 레포지토리에서 검색

## 273. Customizing chart parameters

- `helm install my-release bitnami/wordpress --set wordpressBlogName="Helm tutorials" --set wordpressEmail="john@example.com"` : `--set` 옵션 사용하여 parameter value 지정
- `helm install my-release bitnami/wordpress --values customvalues.yaml` : 사용자 정의 values 파일 사용
- 차트 다운로드 후 직접 수정:

```
helm pull bitnami/wordpress --untar
# values.yaml 수정 후,
helm install my-release ./wordpress
```

## 275. Lifecycle management with Helm

- `helm list` 명령으로 현재 릴리스와 리비전 번호를 확인
- `helm history nginx-release` : 특정 릴리즈에 대한 자세한 내용 조회
- `helm rollback nginx-release 1` : 이전 리비전으로 롤백. 롤백 시 완전히 이전 리비전으로 돌아가는 것이 아니라, 이전 리비전과 동일한 구성을 가진 새 리비전이 생성됨