

Helm Charts Deployments

Helm Charts is essentially a package manager for Kubernetes resources. Helm uses a packaging format called charts. A chart is a collection of files that describe a related set of Kubernetes resources. The Fastlane deployment pipeline relies on Helm Charts to manage deployments for wbd-streaming applications.

Github repo: [wbd-streaming/helm-charts](#)

Local Setup

Reference: [BOLT Onboarding Handbook | Add Artifactory Repo\(s\) to Helm](#)

In a terminal window, run the following to add Helm repo. Substitute repo name, URL, and your email address.

```
1 helm repo add wbd-streaming https://discovery.jfrog.io/artifactory/helm-charts/ --username your_email@discovery.c
```

It will prompt for password. Provide the identity token from [Artifactory](#). (The repo name in the screenshot below is outdated, but the idea remains the same)

```
helm repo add wbd-streaming-helm-snapshots https://discovery.jfrog.io/artifactory/helm-charts-helm-snapshots/ --username zihao_yu-wm@discovery.com
Password:
"wbd-streaming-helm-snapshots" has been added to your repositories
```

Update Helm to Fetch Available Charts

```
1 helm repo update
2
3 ...
4
5 ...Successfully got an update from the "wbd-streaming" chart repository
```

Working Locally

List your repos:

```
1 helm repo list
2 NAME          URL
3 wbd-streaming https://discovery.jfrog.io/artifactory/helm-charts/
```

Pull your chart:

```
1 helm pull wbd-streaming/microservice --version <chart version>
```

Render your resources with values file (replace with real values files):

```
1 helm template my-service wbd-streaming/microservice --version <chart version> -f ./helm/values-default.yaml -f ./
```

Test render your chart:

```
1 # render all
2 helm template test microservice-<chart version>.tgz
3
4 # render specific template
5 helm template test microservice-<chart version>.tgz -s templates/deployment.yaml
```

Cheat Sheet

Reference: [🌐 Helm Commands Cheat Sheet {Free Downloadable PDF}](#)

Install and Uninstall

Install an app:

```
1 helm install [app-name] [chart]
```

Install an app in a specific namespace:

```
1 helm install [app-name] [chart] --namespace [namespace]
```

Override the default values with those specified in a file of your choice:

```
1 helm install [app-name] [chart] --values [yaml-file/url]
```

Run a test installation to validate and verify the chart:

```
1 helm install [app-name] --dry-run --debug
```

Uninstall a release:

```
1 helm uninstall [release]
```

Upgrade and Rollback

Upgrade an app:

```
1 helm upgrade [release] [chart]
```

Instruct Helm to rollback changes if the upgrade fails:

```
1 helm upgrade [release] [chart] --atomic
```

Upgrade a release. If it does not exist on the system, install it:

```
1 helm upgrade [release] [chart] --install
```

Upgrade to a specified version:

```
1 helm upgrade [release] [chart] --version [version-number]
```

Roll back a release:

```
1 helm rollback [release] [revision]
```

Download Release Info

Download all the release information:

```
1 helm get all [release]
```

Download all hooks:

```
1 helm get hooks [release]
```

Download the manifest:

```
1 helm get manifest [release]
```

Download the notes:

```
1 helm get notes [release]
```

Download the values file:

```
1 helm get values [release]
```

Fetch release history:

```
1 helm history [release]
```

Manage Repositories

Add a repository from the internet:

```
1 helm repo add [repository-name] [url]
```

Remove a repository from your system:

```
1 helm repo remove [repository-name]
```

Update repositories:

```
1 helm repo update
```

List chart repositories:

```
1 helm repo list
```

Managing Releases

List all available releases in the current namespace:

```
1 helm list
```

List all available releases across all namespaces:

```
1 helm list --all-namespaces
```

List all releases in a specific namespace:

```
1 helm list --namespace [namespace]
```

List all releases in a specific output format:

```
1 helm list --output [format]
```

Apply a filter to the list of releases using regular expressions:

```
1 helm list --filter '[expression]'
```

See the status of a specific release:

```
1 helm status [release]
```

Display the release history:

```
1 helm history [release]
```

See information about the Helm client environment:

```
1 helm env
```

Chart Management

Run tests to examine a chart and identify possible issues:

```
1 helm lint [chart]
```

Inspect a chart and list its contents:

```
1 helm show all [chart]
```

Display the chart's definition:

```
1 helm show chart [chart]
```

Display the chart's values:

```
1 helm show values [chart]
```

Download a chart and extract the archive's contents into a directory:

```
1 helm pull [chart] --untar --untardir [directory]
```