Helm

Terminology

1. A *Chart* is a Helm package. It contains all of the resource definitions necessary to run an application
2. A *Repository* is the place where charts can be collected and shared. Here's good reference for these: https://artifacthub.io/packages/search?kind=0
3. A Release is an instance of a chart running in a Kubernetes cluster. One chart can often be installed many times into the same cluster. And each time it is installed, a new release is created.

So... Helm installs *charts* into Kubernetes, creating a new *release* for each installation. And to find new charts, you can search Helm chart *repositories*.

What can it do?

Helm can do the following:

- Create new charts from scratch
- Charts are templates for kubernetes manifests
- Package charts into chart archive (tgz) files
- Interact with chart repositories where charts are stored
- Install and uninstall charts into an existing Kubernetes cluster
- Manage the release cycle of charts that have been installed with Helm

Helm cli

- We work with helm v3
- Installing helm: https://helm.sh/docs/intro/install/#from-script
- Renders templates
- Talks to kube-api-server

```
helm install <releasename> chartrepo/chartname
helm upgrade <releasename> --set replicas=2
helm uninstall <releasename>
```

Helm templating

```
port: {{ .Values.Port }}
```

```
port: {{ .Release.Name }}
```

A lot more tips and tricks:

- https://helm.sh/docs/howto/charts_tips_and_tricks/

Exercise 7 - Hackathon

Migrate the entire existing sunny bikes solution to a Helm chart

- Decide for yourself which variables should be configurable (e.g. the number of replicas)
- Make sure there are no hardcoded values (i.e. the PG_HOST); these should be calculated from variables

Bonus:

- Make persistence optional
- Add horizontal auto-scaling for sunnybikes
- Generate a random password for postgres on install (hint, checkout "randAlphaNum" and this fix for it.