

Creating a Helm chart (Part 1)

(Demo prep)

- Uninstall all microservice helm charts
- Ensure all microservices are deployed via standard yaml files
- Delete all images from ACR that should not be there
- Delete GitHub app from AAD and all its RBAC assignments

Start

Helm is a package manager tool that allows you to install applications into Kubernetes using what is known as a Helm chart. A helm chart contains all the resource definitions necessary to run an application, or in our case a microservice, in Kubernetes.

You can use the same Helm chart to install multiple microservices, while at the same time letting each microservice specify values that are specific to each of them. And since helm charts can be versioned, you can always rollback to a previous version of your microservice if things don't go as expected.

So, let's go ahead and convert the Kubernetes yaml files of our Identity microservice into a reusable Helm chart that later we can use to install all of our microservices.

In Identity repo

1. Create a new **helm** dir at the root
2. Create a **templates** dir under **helm** dir
3. Create **deployment.yaml**, **service.yaml** and **certificate.yaml** files into the **templates** dir
4. Copy the deployment definition from identity.yaml into **deployment.yaml**
5. Copy the service definition from identity.yaml into **service.yaml**
6. Copy the certificate definition from signing-cer.yaml into **certificate.yaml**
7. Create a **values.yaml** file in the **helm** dir
8. Move deployment.yaml values into values.yaml:

values.yaml

microserviceName: "identity"

image:

repository: playeconomy.azurecr.io/play.identity

```
tag: "1.0.10"
```

deployment.yaml

```
apiVersion: apps/v1
```

```
kind: Deployment
```

```
metadata:
```

```
  name: "{{ .Values.microserviceName }}-deployment"
```

```
spec:
```

```
  selector:
```

```
    matchLabels:
```

```
      app: {{ .Values.microserviceName }}
```

```
  template:
```

```
    metadata:
```

```
      labels:
```

```
        app: {{ .Values.microserviceName }}
```

```
        aadpodidbinding: {{ .Values.microserviceName }}
```

```
    spec:
```

```
      containers:
```

```
        - name: {{ .Values.microserviceName }}
```

```
          image: "{{ .Values.image.repository }}:{{ .Values.image.tag }}"
```

What do we do about these environment variables? Let's actually cover that in the next lesson, where you will learn several other techniques that you'll use to finish extracting all the values in this file and all your other yaml files.