Using Kubernetes secrets

In Kubernetes, a Secret is an object that contains a small amount of sensitive data, such as a password, a token or a key. By using Kubernetes secrets, you avoid having to include confidential data in your application code and, since they can be created independently of the Pods that use them, there is less risk of the secret being exposed during the creation of the pod.

Let's see how to create a Kubernetes secret to hold the values needed by our Identity deployment.

In Play.Identity repo

1. Update the README:

````
## Create the Kubernetes namespace
```powershell
$namespace="identity"
kubectl create namespace $namespace
```

## Create the Kubernetes secrets
```powershell
kubectl create secret generic identity-secrets --from-literal=cosmosdb-connectionstring=$cosmosDbConnString --from-literal=servicebus-connectionstring=$serviceBusConnString --from-literal=admin-password=$adminPass -n $namespace
```
````

2. Run the commands

3. Add the secrets to the yaml file:

```
env:
 …
 - name: MongoDbSettings__ConnectionString
   valueFrom:
     secretKeyRef:
       name: identity-secrets
       key: cosmosdb-connectionstring
 - name: ServiceBusSettings__ConnectionString
   valueFrom:
     secretKeyRef:
       name: identity-secrets
       key: servicebus-connectionstring
 - name: IdentitySettings__AdminUserPassword
   valueFrom:
     secretKeyRef:
       name: identity-secrets
```

```
        key: admin-password
    resources:
    …
```

4. Commit and push.

In the next lesson you will deploy your first Kubernetes pod.