Week 3: Add Enterprise Qualities

Unit 4: Deploying for Production

Overview

CAP applications can be deployed to the Cloud Foundry Environment of SAP Cloud Platform.

Cloud Foundry (CF) is a platform-as-a-service environment for running cloud-native applications.

Use the 'cf' command line interface (CLI) to

- Deploy your application to CF
- Create instances of platform services, e.g. SAP HANA service, User Account & Authentication service
- Bind those service instances to your application

Deployment using cf push

Use the `cf push` command to deploy applications. It needs

- Application files to be deployed
- An optional manifest file

Services given in the manifest will be *bound* to the application

… but need to be created *manually* beforehand, e.g. using
`cf create-service <type> <plan> <name>`

Plugin create-service-push is an alternative to automize service creation

See openSAP course SAP Cloud Platform Essentials (week 1, unit 5) for more on pushing applications to Cloud Foundry

`manifest.yaml`

```
---
applications:
- name: bookshop-db
  services:
  - bookshop-db-hdi-container

- name: bookshop-srv
  memory: …
  services:
  - bookshop-db-hdi-container
```

Deployment using cf push – Demo

1. Configure your project for SAP HANA:
   ```
   cds add hana
   ```
2. Create ready-to-deploy files, incl. table definitions, manifests, and more:
   ```
   cds build/all
   ```
3. Create the SAP HANA service, incl. a DB schema:
   ```
   cf create-service hana hdi-shared bookshop-db-hdi-container
   ```
4. Push the database deployer application:
   ```
   cf push -f gen/db
   ```
5. Push the actual service application:
   ```
   cf push -f gen/srv --random-route
   ```
6. Find and open the application URL:
   ```
   ...
   requested state: started
   routes: bookshop-srv-...hana.ondemand.com
   ```

gen/db/manifest.yaml

```
---
applications:
- name: bookshop-db
  path: gen/db
  services:
  - bookshop-db-hdi-container
```

Multitarget applications – Overview

On top of cf push, SAP offers a convenient deployment API for CF called multitarget applications (MTA).

Features:

- Building and packaging self-contained and versioned application archives
- Single descriptor for even advanced deployments
- Declarative MTA descriptor to avoid maintaining deployment scripts. Detailed CF deployment actions are automatically derived out of it.
- Configuration parameters to distinguish different deployments (e.g. test vs. production)

Open Source tools are available for building and packaging MTAs and for deploying MTAs with the CF CLI.

Multitarget applications – Build

- An MTA descriptor (`mta.yaml`) declares "modules" and "resources"
- Build your project using the MTA Build Tool (MBT). This creates a deployable artifact, the MTA archive.
- The MTA archive can then be deployed to CF with `cf deploy`
- CAP provides a tool to generate the `mta.yaml` file from existing configuration
- Enhance `mta.yaml` as needed, e.g., to add
  - More modules, e.g. UI modules
  - Specify creation parameters for platform services

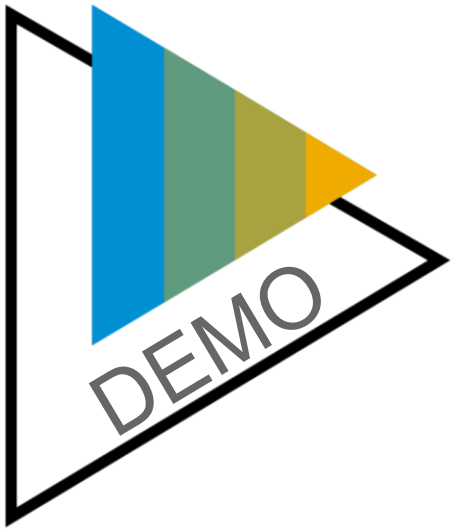
`mta.yaml`

```yaml
_schema-version: '3'
ID: bookshop
version: 1.0.0

modules:  # correspond to CF applications
   # server module
 - name: bookshop-srv
   type: nodejs
   path: gen/srv
   requires:
    - name: bookshop-db-hdi-container

   # SAP HANA deployer module
 - name: bookshop-db
   type: hdb
   path: gen/db
   requires:
    - name: bookshop-db-hdi-container

resources: # correspond to CF services
   # SAP HANA service
 - name: bookshop-db-hdi-container
   type: org.cloudfoundry.managed-service
   parameters:
     service: hanatrial  # or 'hana'
     service-plan: hdi-shared
```

Multitarget applications – Demo

Prerequisites:

- Install the MTA Build Tool, e.g. using `npm install -g mbt`



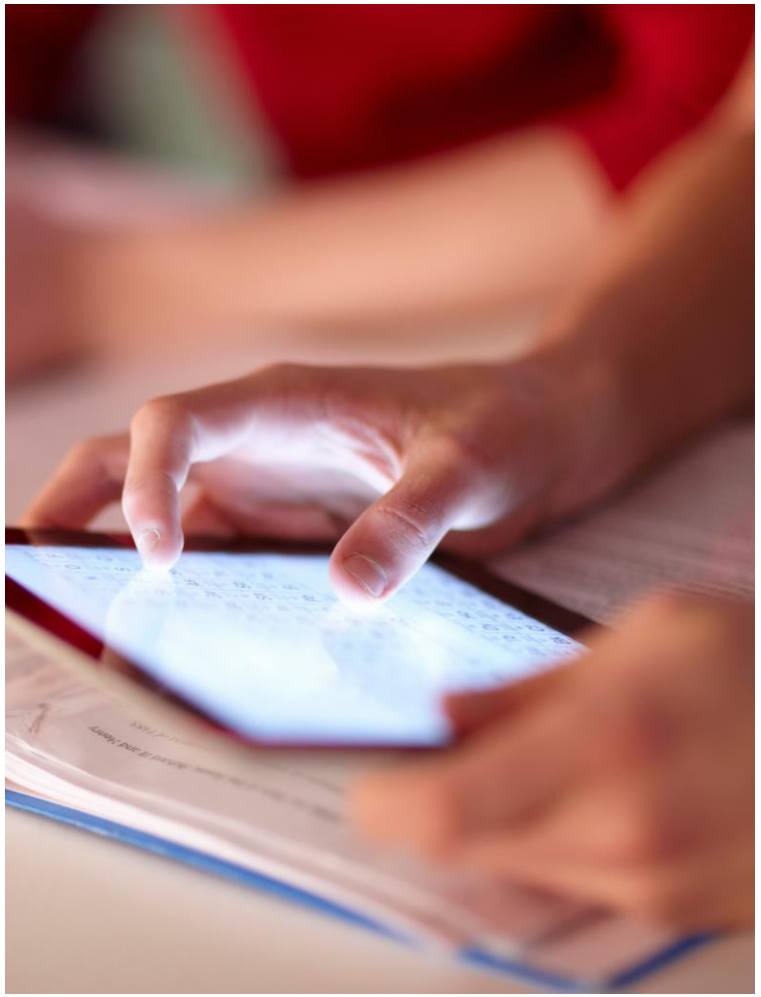
Procedure:

1. Configure the project for SAP HANA, adding e.g. the driver
   ```
   cds add hana
   ```
2. Generate MTA project descriptor file `mta.yaml`
   ```
   cds add mta
   ```
3. Build MTA archive
   ```
   mbt build
   ```
4. Log into your CF space. Then deploy MTA archive
   ```
   cf deploy mta_archives/….mtar
   ```
5. Identify the URL attached to the CF application "bookshop-srv" and open it in a browser

What you've learned in this unit

- How to build a deployment package for Node.js
- How to deploy applications using cf push
- How to deploy multitarget applications
- The pros and cons of both approaches

Also see the 'Continuous Integration and Deployment' unit in week 4 that takes deployments to the next level of automation.



Thank you.

Contact information:

open@sap.com

Follow all of SAP

www.sap.com/contactsap

THE BEST RUN SAP®