SIMPLE SAP INVENTORY MANAGER

(https://inventorymanager-demo.cfapps.eu10.hana.ondemand.com)

Note: I use CloundFoundry Trial and will expire about 30 days

1. DOCUMENTATION

This application consists of two parts namely CRUD inventory goods and overview inventory goods using the chart. I use NodeJS and Express to expose the API to the service and deploy it using CloudFoundry. In Front End, i use Fiori principle (https://experience.sap.com/fiori-design-web/) using OpenUI5 (https://openui5.org/).

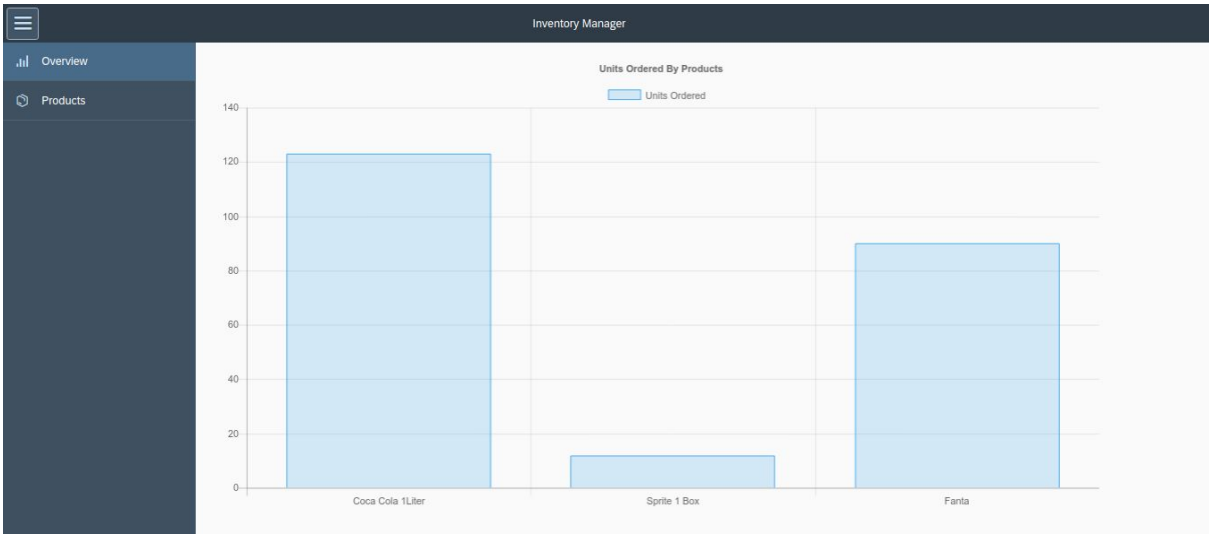


Image of Inventory Overview Page (a simple bar chart show units ordered for each products)

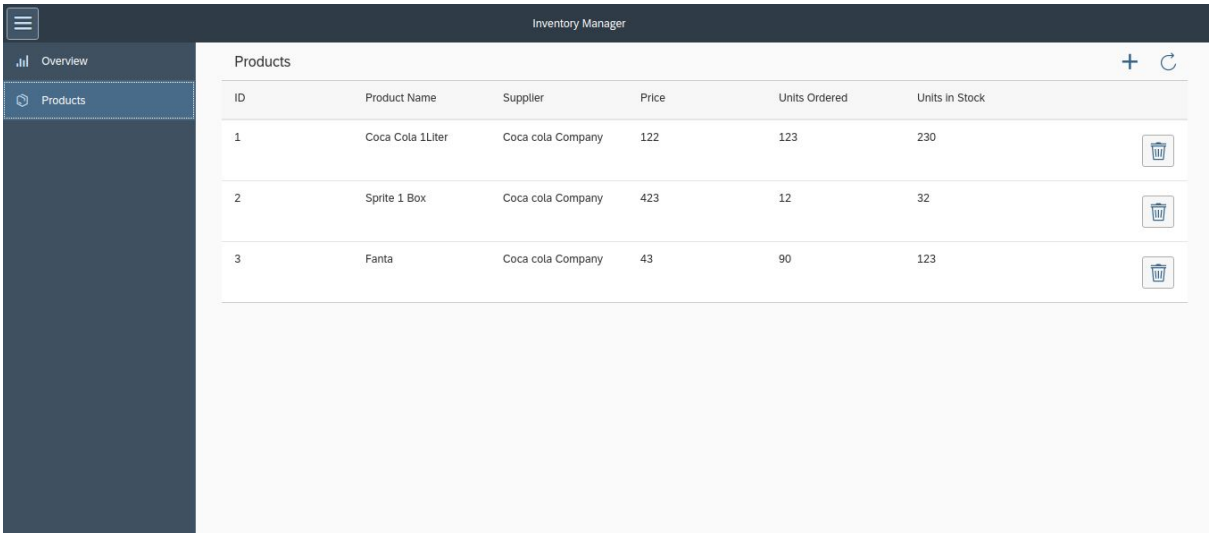


Image of CRUD List Inventory

The purpose of this app is to maintain your inventory and see the top product ordered.
In this project i use the following controller/library:

1. OpenUI5: Easy and responsiveness ready, mainly using the side navigation, and list

2. OpenUI5-ChartJS (https://github.com/StErMi/openui5-chartjs): Wrapper for popular chart library Chart.js, easy to understand and implement.
3. SCP CloudFoundry (https://cloudplatform.sap.com): For deploying node JS Server
4. MongoDB service: For persisting the data

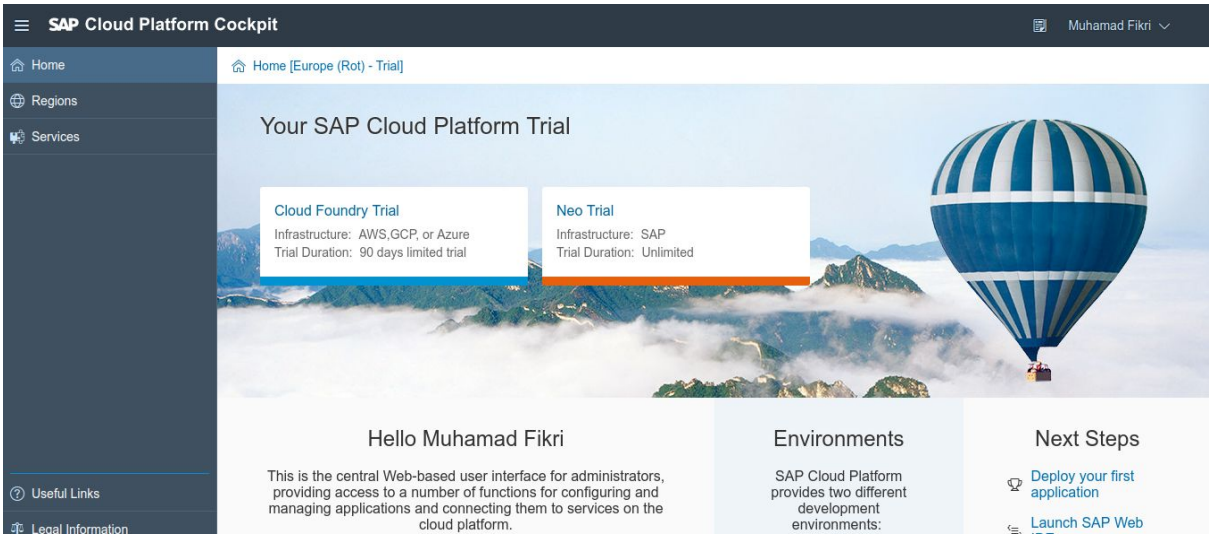
In the application root folder the folders “server” and “client” are created to separate the server/client logic from the beginning.

2. DEPLOYMENT

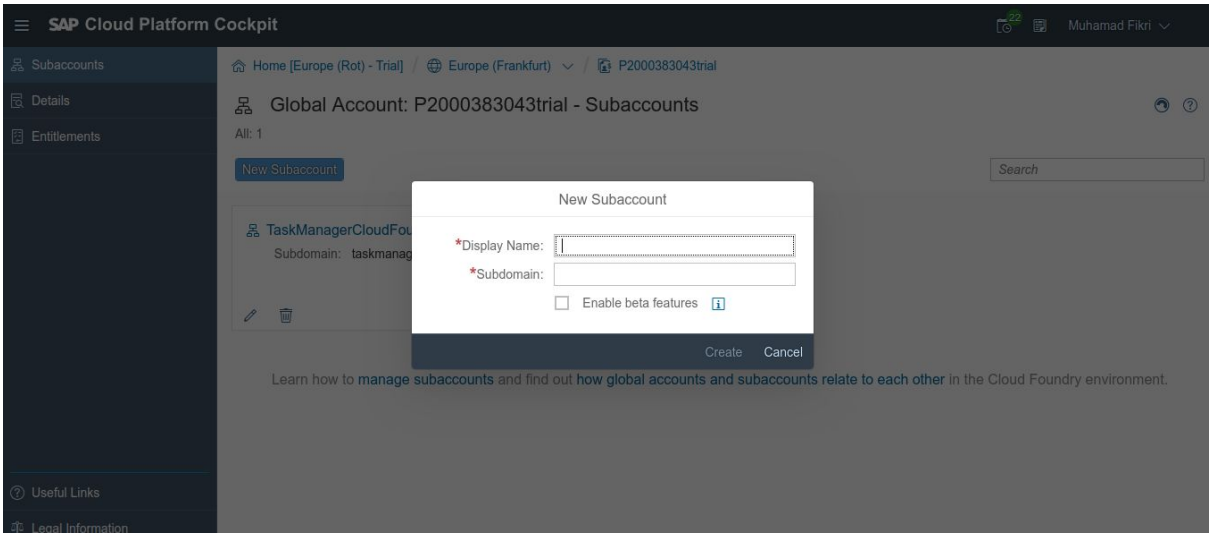
To be able to do the implementation steps some prerequisites have to be fullfilled.
First of all a SCP Cloud Foundry trial instance has to be created.

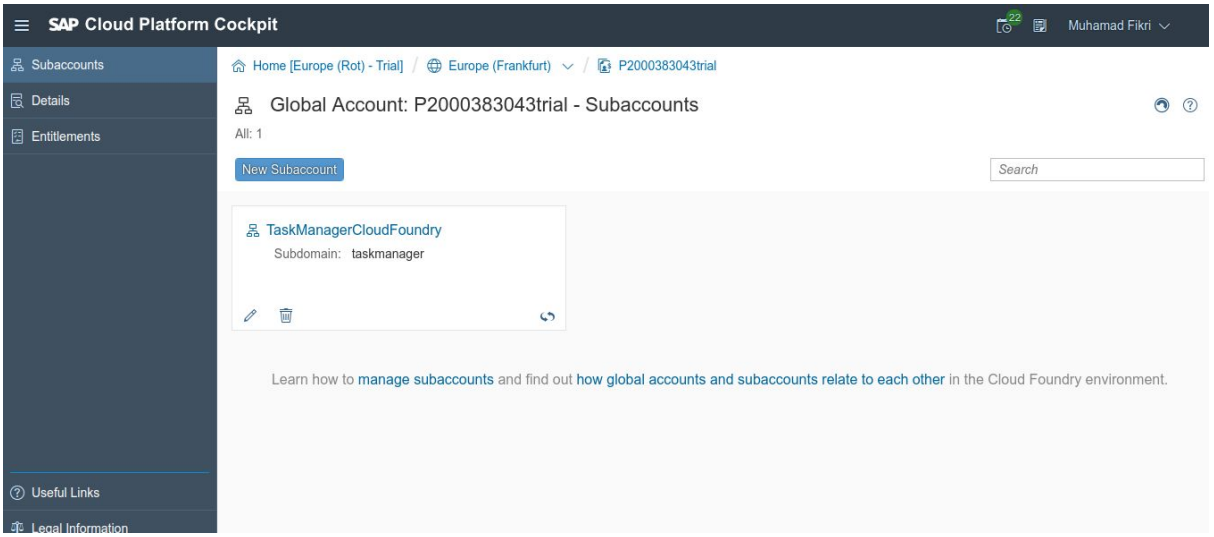
a. Prepare Deploy Environment

- Create a trial account in https://cloudplatform.sap.com
- Login and access https://account.hanatrial.ondemand.com/cockpit
- Click CloudFoundry Trial to setup

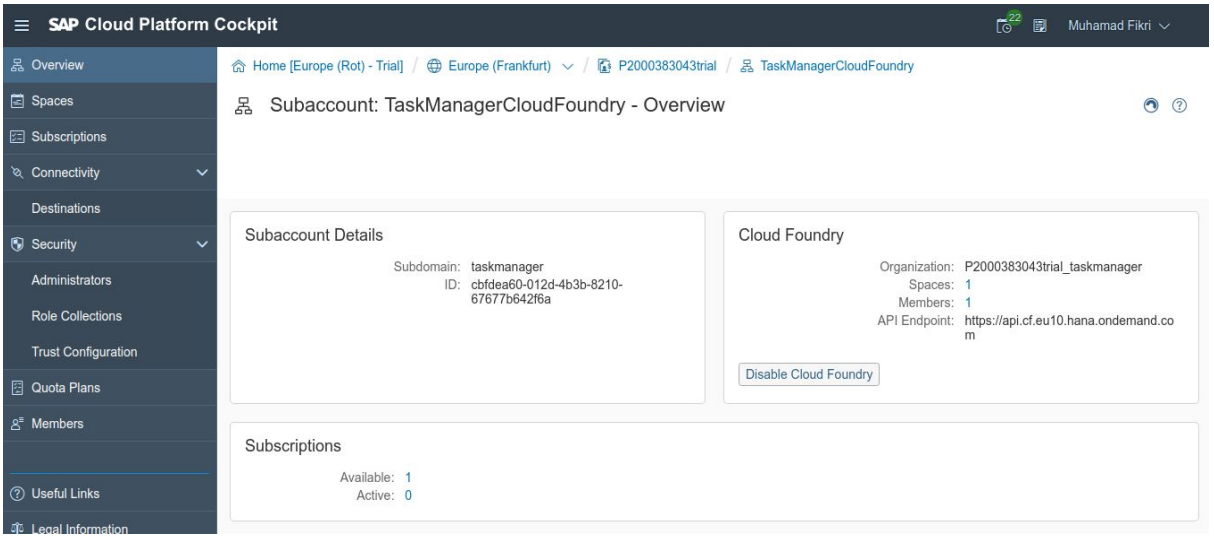


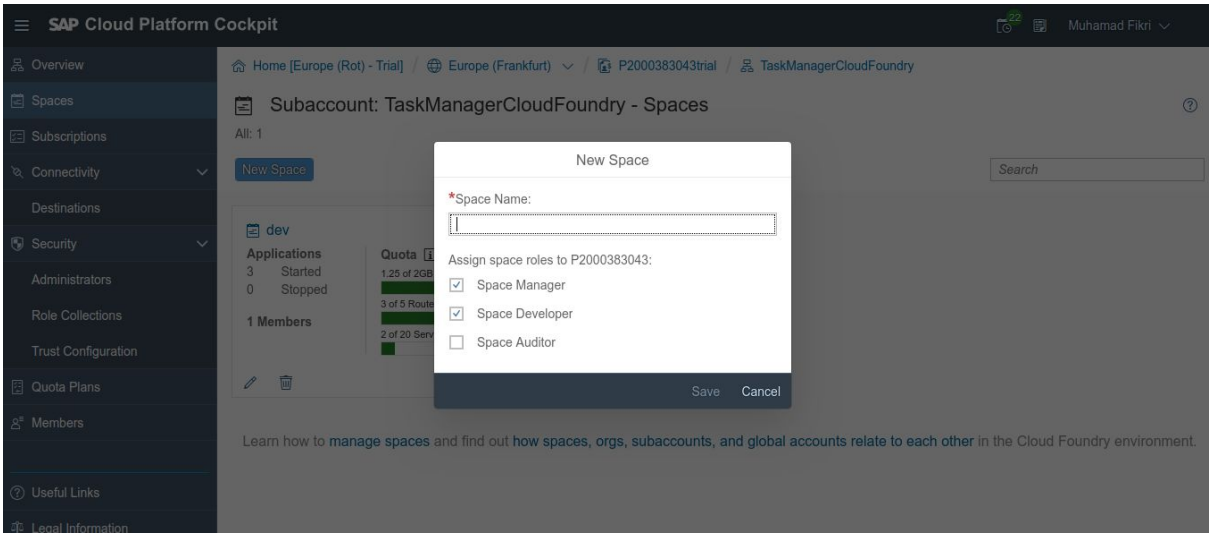
- Start create subaccount for CloudFoundry trial



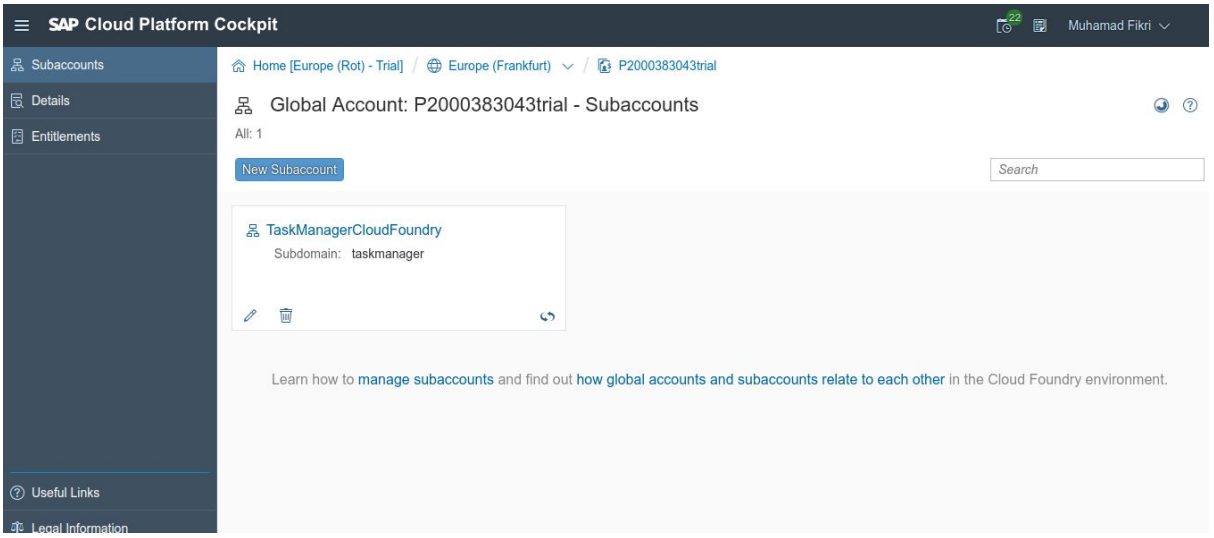


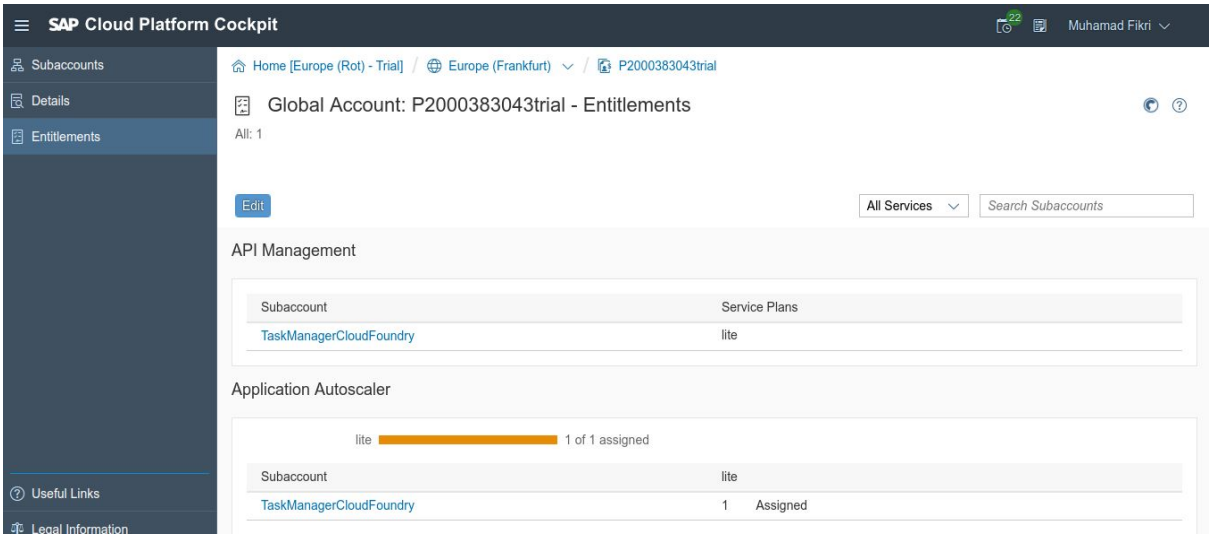
- Click The subaccount in this case TaskManagerCloudFoundry
- Click space to create the space for your app





- After space was created, back again to subaccount page and create entitlement for the space (Click Entitlement in the sidebar navigation)
- Assign to your space: API Management, Application Runtime, Connectivity, Destination, MongoDB, Portal Service

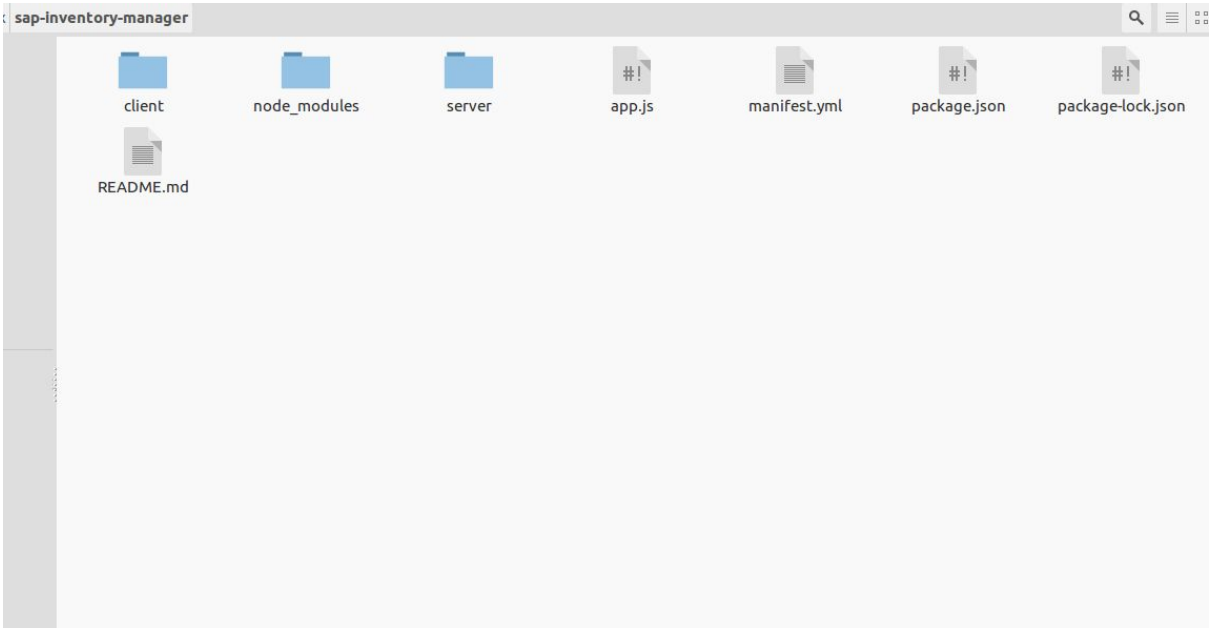




- At this point, your environment for deploying your apps is ready

b. Deploy Application

- Extract the app

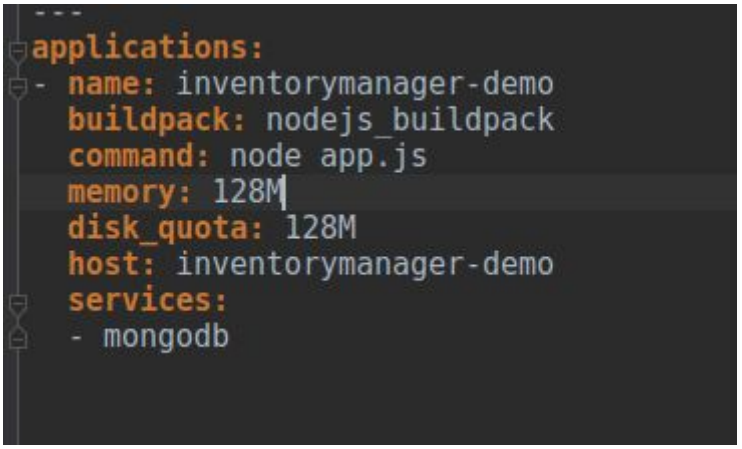


- To interact with the Cloud Foundry instance via command line the Cloud Foundry CLI has to be installed on the local machine and the connection to the instance has to be established using the commands “cf api” and “cf logon”. All details are described: https://help.sap.com/viewer/65de2977205c403bbc107264b8eccf4b/Cloud/en-US/4ef907afb1254e8286882a2bdef0edf4.html and https://help.sap.com/viewer/65de2977205c403bbc107264b8eccf4b/Cloud/en-US/7a37d66c2e7d401db4980db0cd74aa6b.html

```
bukalapak@fe-fikri:~$ cf api
api endpoint:   https://api.cf.eu10.hana.ondemand.com
api version:    2.112.0
bukalapak@fe-fikri:~$ cf login
API endpoint: https://api.cf.eu10.hana.ondemand.com

Email> mfikria@gmail.com
```

- In the application manifest file “manifest.yml”, information about the Node.js application and Cloud Foundry are described. The information is used by the “cf push” command for setting up the environment when the application is deployed to the Cloud Foundry instance. Later information will be added for the MongoDB service consumption.



Following information is provided in the file:

name: The name of the application.

buildpack: The name of the Node.js buildpack determined before with command “cf buildpacks”. It is also possible to reference the buildpack sources on GitHub. By default an auto determination of the buildpack is done if the buildpack information is missing in the application manifest. But from my point of view it is clearer to specify it in the application manifest.

command: Node.js applications needs a start command to start the application. In the example “node app.js” is called which executes the JS code in a file “app.js” which is described later. The command is executed automatically after the application is successfully deployed to the Cloud Foundry instance.

memory: Definition of the RAM available for the application. For the demo 128 MB are used.

disk_quota: Definition of the disk space available for the application. For the demo 128 MB are used.

host: Host information for the application which is used in the URL which makes the application accessible.

- MongoDB is available as service in the SCP CF environment. To display the available services the command

  ```
  cf marketplace
  ```

- To create an instance of the MongoDB service the command

  ```
  cf create-service mongodb v3.0-container mongodb-service
  ```

  needs to be executed. “mongodb-service” is the service instance name which is used to bind the service instance to the application.
- Executing command

  ```
  cf services
  ```

shows the created service instance. The service instance can also be seen in the SCP CF cockpit “Service Instances” area.

- To bind the service instance to the application, following “services” property using the name of the MongoDB service instance has to be added to the application manifest.

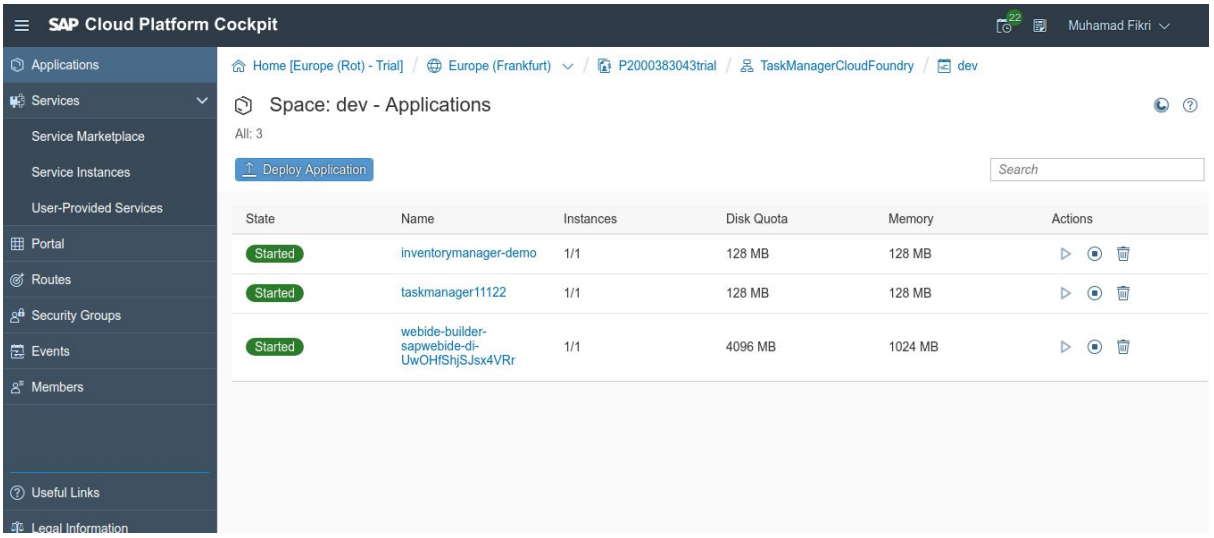
```
---
applications:
- name: inventorymanager-demo
  buildpack: nodejs_buildpack
  command: node app.js
  memory: 128M
  disk_quota: 128M
  host: inventorymanager-demo
  services:
  - mongodb-service
```

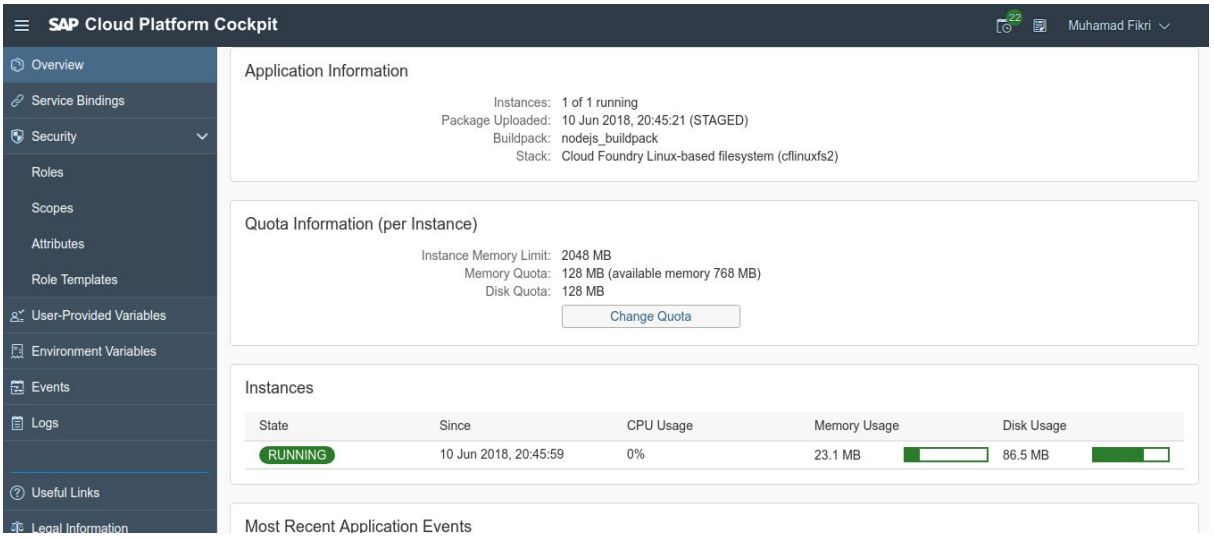
- So far the application was only tested on the local machine. Now it is time to deploy the application to the Cloud Foundry instance using command

  `cf push`

  executed in the application root folder where the application manifest file is available.

  The log shows different information regarding the deployment steps and finally reports that the application is successfully started and running.




SAP Cloud Platform Cockpit
22
Muhamad Fikri
Overview
Service Bindings
Security
Roles
Scopes
Attributes
Role Templates
User-Provided Variables
Environment Variables
Events
Logs
Useful Links
Legal Information
Application Information
Instances: 1 of 1 running
Package Uploaded: 10 Jun 2018, 20:45:21 (STAGED)
Buildpack: nodejs_buildpack
Stack: Cloud Foundry Linux-based filesystem (cflinuxfs2)
Quota Information (per Instance)
Instance Memory Limit: 2048 MB
Memory Quota: 128 MB (available memory 768 MB)
Disk Quota: 128 MB
Change Quota
Instances
State
Since
CPU Usage
Memory Usage
Disk Usage
RUNNING
10 Jun 2018, 20:45:59
0%
23.1 MB
86.5 MB
Most Recent Application Events