Deploy and test an S/4HANA Cloud extension in Kyma

This document demonstrates the steps to deploy and test an S/4HANA Cloud extension in SAP Business Technology Platform, Kyma Runtime.

www.sap.com/contactsap

THE BEST RUN SAP®

Table of Contents

DISCLAIMER

The information shared in this document is confidential and proprietary to SAP and may not be disclosed without the permission of SAP. All functionality presented here is subject to change and may be changed by SAP at any time for any reason without notice.

OBJECTIVE

This document is aimed for cloud developers (beginners or seasoned) and DevOps professionals.

The objective of this exercise is to demonstrate the steps to deploy and test an S/4HANA Cloud extension in SAP Business Technology Platform, Kyma Runtime.

SCENARIO

The S/4HANA Cloud extension is a simple Business Partner Management app, managing exclusively records of category "Person" and using only S/4HANA master data as persistence (nothing stored on the extension side, thus dismissing any database for persistence). It simply leverages the S/4 Business Partner OData service to execute the CRUD operations directly in the S/4 system, so it's quite simple.

App components:

1. **Frontend UI**: ***SAP Fiori Elements HTML5 app*** based on the List Report/Object page templates. No coding on the UI – everything is interpreted from the backend service annotations by the Fiori Elements engine.
2. **Backend service**: ***Cloud Application Programming Model app*** with simple service exposing an OData entity
3. **Create and update operations**: ***Serverless Function*** deployed directly in Kyma and invoked either directly or through an event message via event subscription. The event publishing can occur either from inside or outside the cluster (in the second case it uses another serverless function exposed outside the cluster via API Rule). All of this is determined through and environment parameter set upon application deployment.

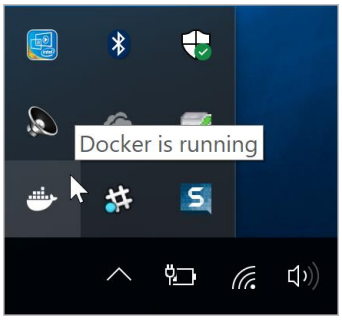
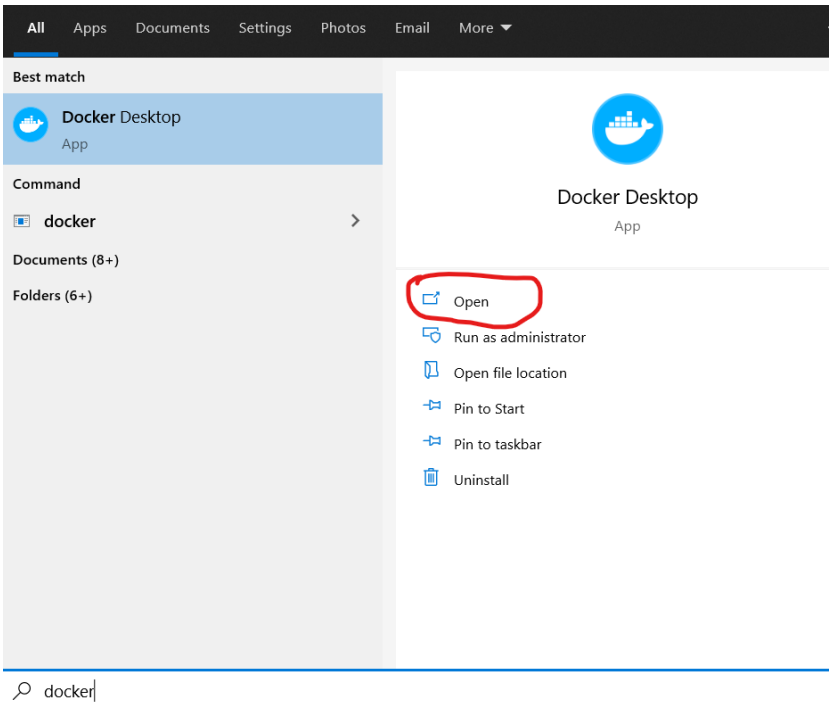
SUGGESTED PRE-READS

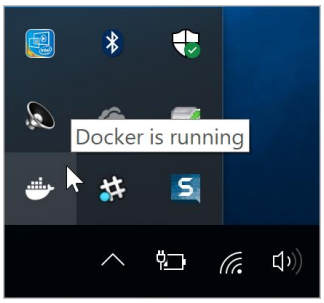
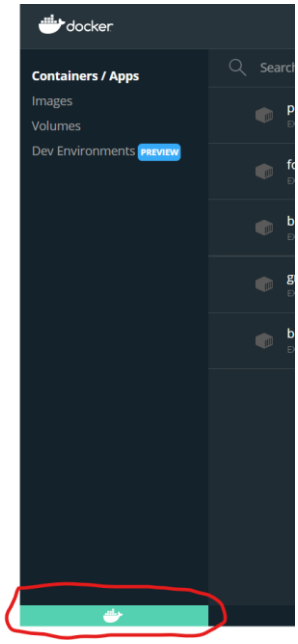
- Basic Concepts of Kyma Environment
- Getting Started with the SAP BTP, Kyma runtime

PREREQUISITES

- Have an **S/4HANA Cloud tenant** available with an active **communication user** set for inbound communication in a **communication arrangement** which is based on the communication scenario **SAP_COM_0008** (see: https://help.sap.com/viewer/0f69f8fb28ac4bf48d2b57b9637e81fa/2108.501/en-US/fab3fd449cf74c6384622b98831e989e.html?q=communication%20arrangement) – this is required to access the Business Partner OData Service v2.
    - **NOTE**: if you cannot comply to the above prerequisite, alternatively, you can build and deploy an **OData Mock Service** following the instructions in **this Git page**. In that case, you'll set your destination to point to the mock URL instead of S/4 and with no basic authentication (no credentials are required).
- Create an SAP BTP trial account: https://developers.sap.com/tutorials/hcp-create-trial-account.html
- Enable SAP BTP, Kyma Runtime: https://developers.sap.com/tutorials/cp-kyma-getting-started.html
- Set up local development environment using VS Code: https://developers.sap.com/tutorials/btp-app-set-up-local-development.html
- Install kubectl: https://kubernetes.io/docs/tasks/tools/ (test the kubectl installation)
- Install Helm Charts: https://helm.sh/docs/intro/install/

- Install Make tool (Windows users):
    - Install Chocolatey: https://chocolatey.org/install
    - Run: **choco install make**.
- Install Docker Desktop: https://www.docker.com/products/docker-desktop/
- Install Docker extension in VS Code: https://code.visualstudio.com/docs/containers/overview
- Create an account on Docker Hub: https://hub.docker.com/
- Make sure Docker Desktop is running locally successfully.

Explanation	Screenshot
Ensure docker symbol in windows taskbar. If not already running, please follow the steps here below.	
Search docker desktop and click Open	

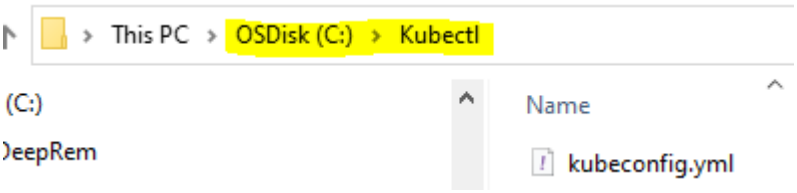
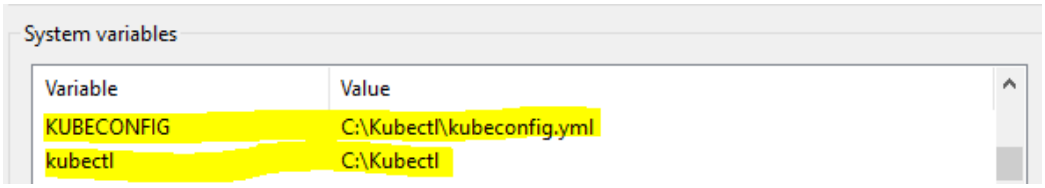
Explanation	Screenshot
Click docker icon and check green status	 

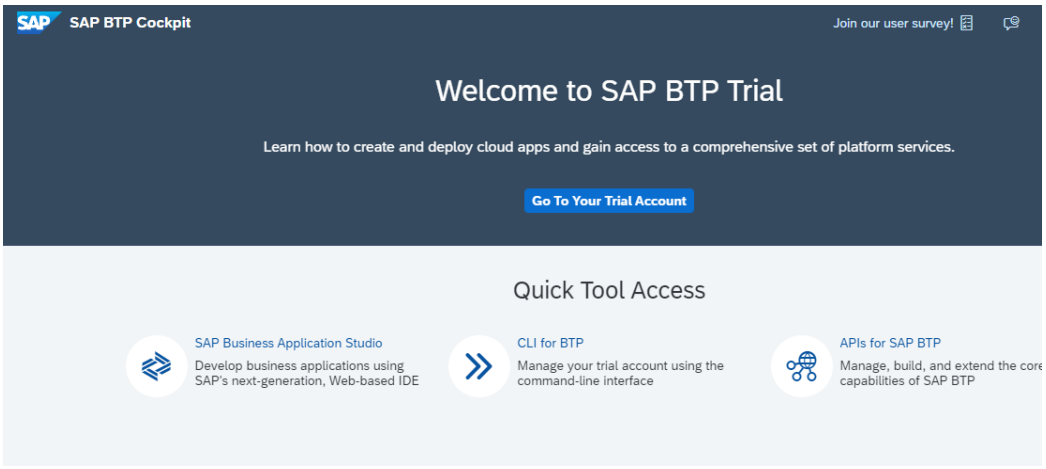
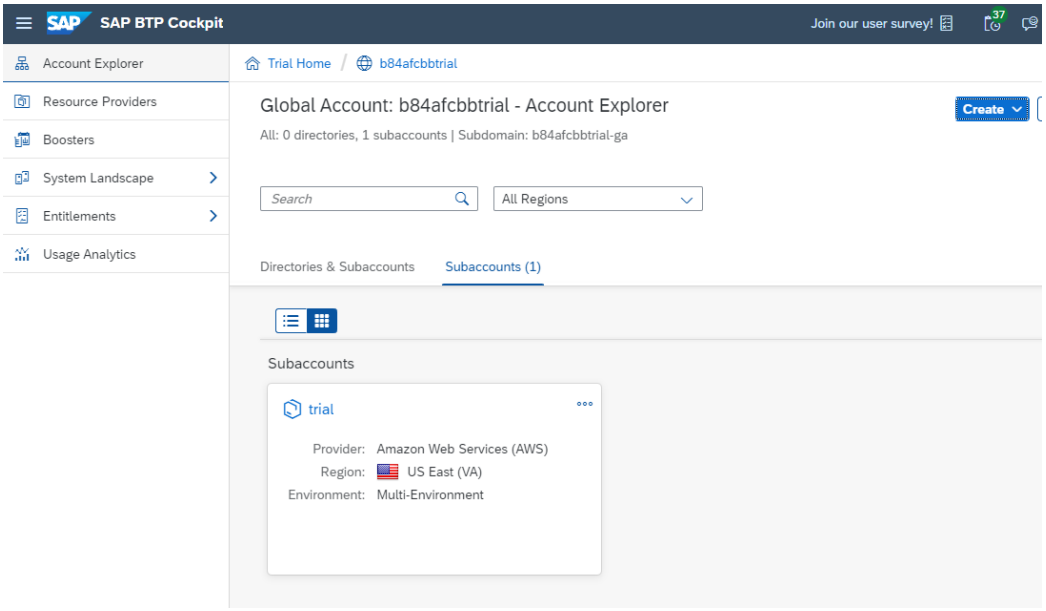
WARNING

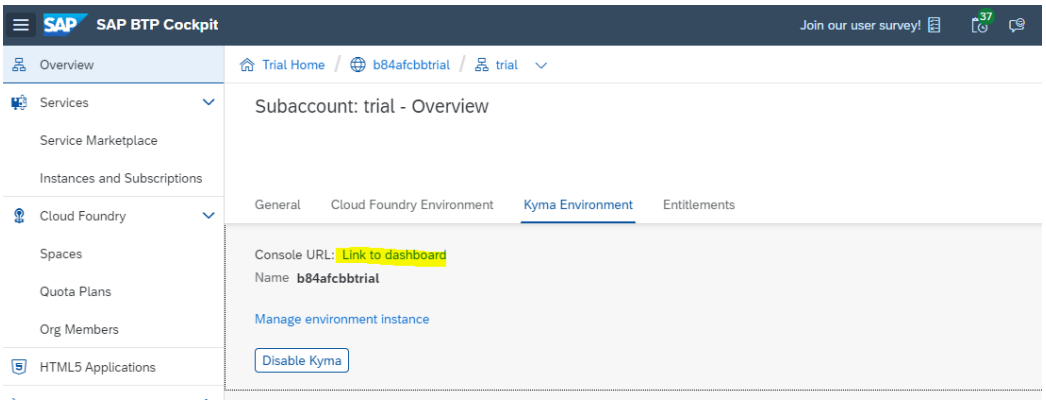
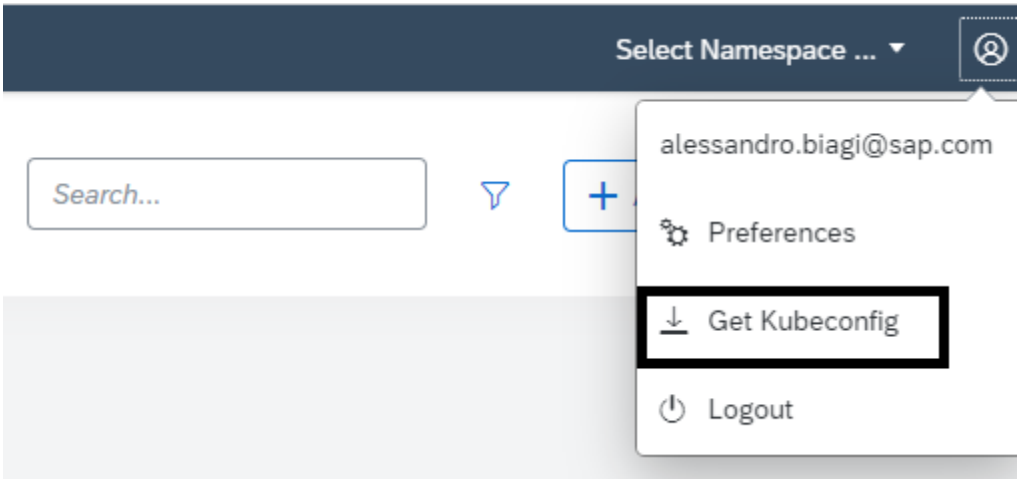
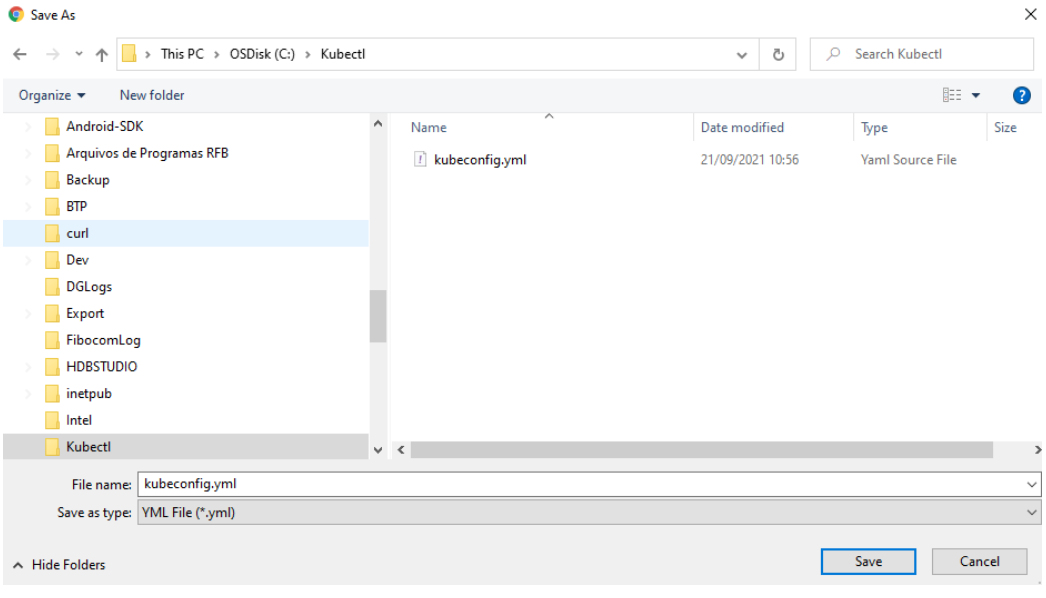
Failing to comply to any of the listed pre-requisites will prevent you from completing the exercise! Therefore, after completing each tutorial/document, make sure that everything you did is working properly!

EXERCISE

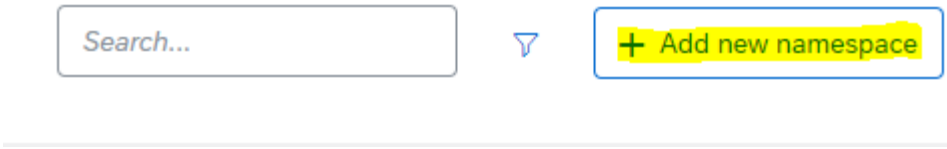
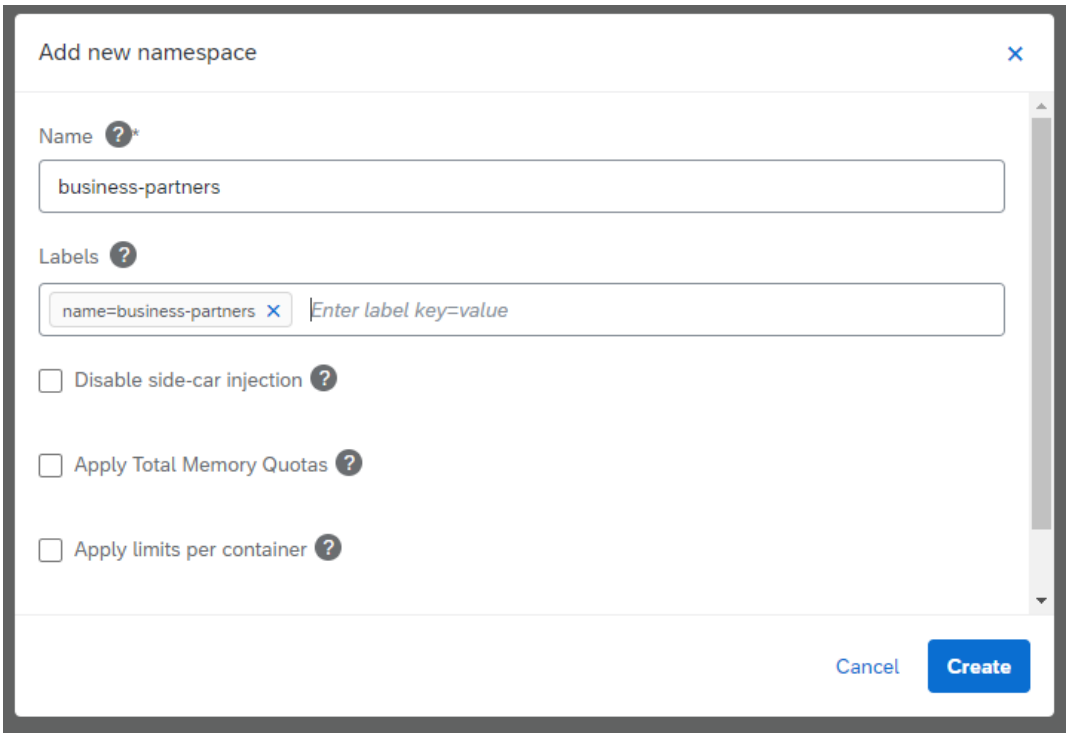
I. Set up Kubernetes context to point to the Kyma cluster

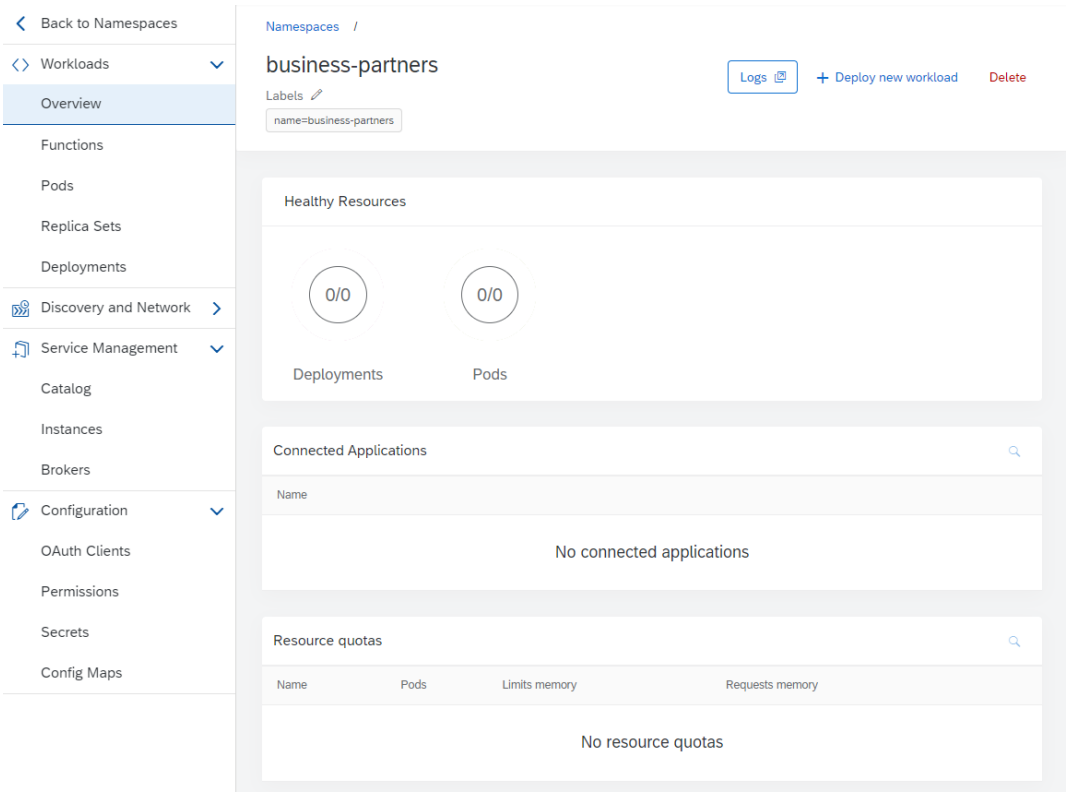
Explanation	Screenshot
Create a folder on your hard-drive to store the **kubeconfig.yml** file (there's an **example** of a folder named "C:\Kubectl" in the screenshot).	
Set two system environment variables (**examples** in the screenshot): • **kubectl=<your folder path>** • **KUBECONFIG=<your folder**	

Explanation	Screenshot
path>\kubeconfig.yml **Note**: variable names ara cese-sensitive!	
Open your BTP trial account cockpit and click **Go To Your Trial Account**	
Click on your subaccount	

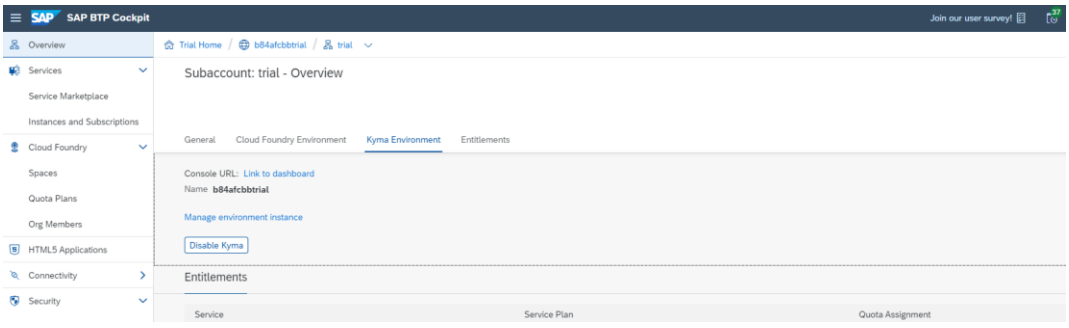
Explanation	Screenshot
Select the **Kyma Environment** tab and click the link: “**Link to dashboard**”	
In the Kyma Console UI click on the “little head” icon on the top right corner and select **Get Kubeconfig**	
Save the kubeconfig.yml file in the folder you created. **Note**: if the file is already there due to a previous access just **overwrite it**. **Do not change the file name!** These two steps to save the config file are required about 12 hours after each access to the cluster as the file token expires after that period..	

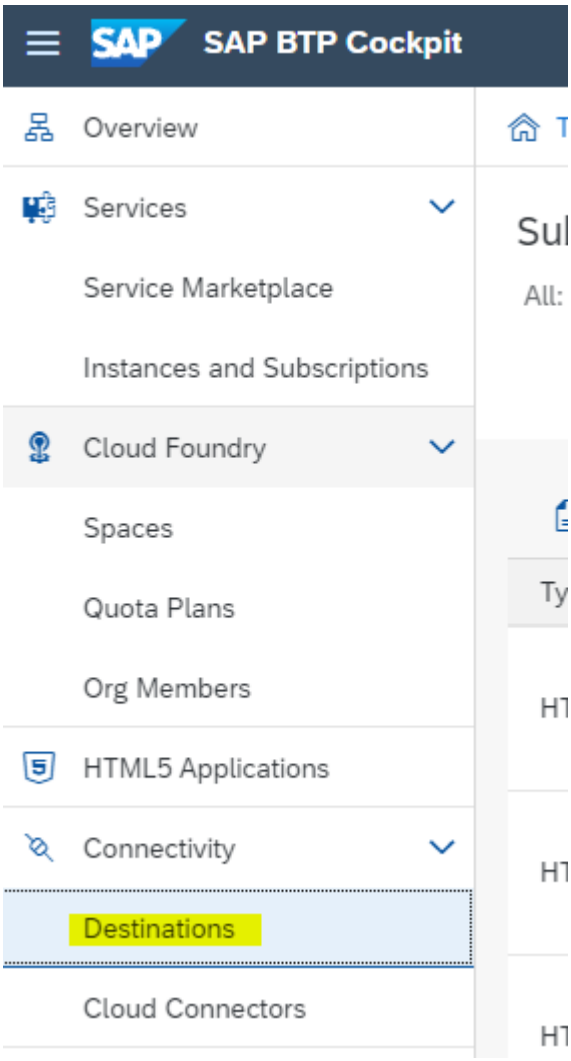
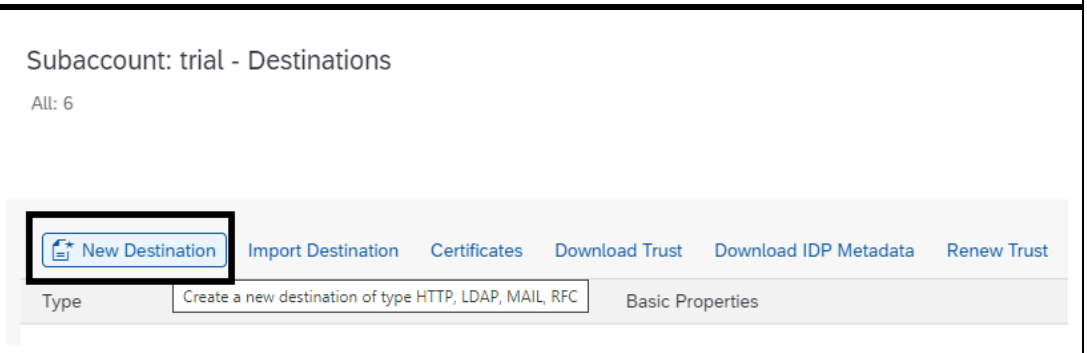
II. Create the application namespace

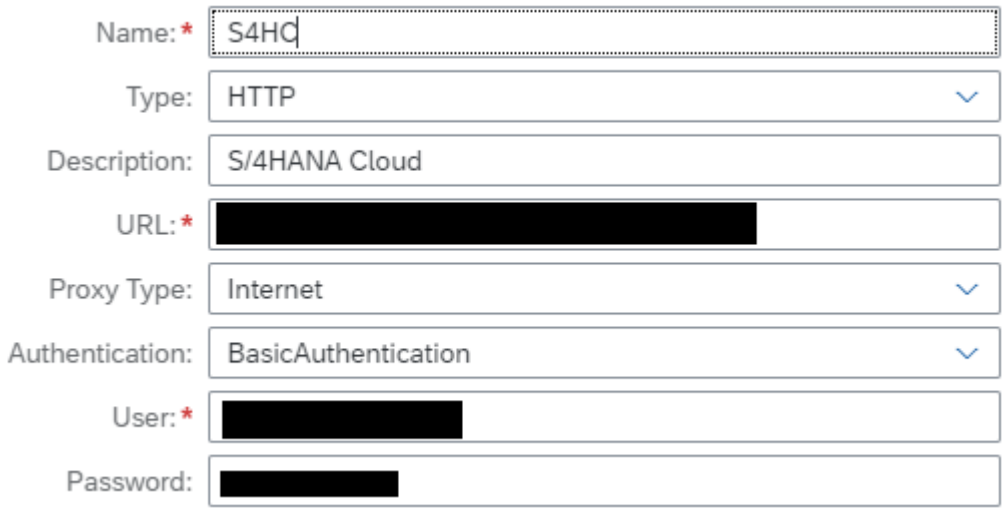
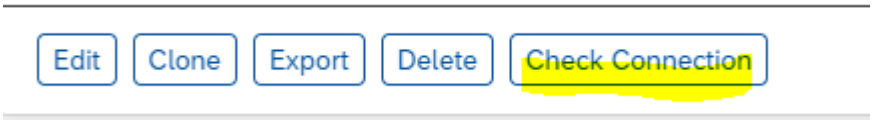
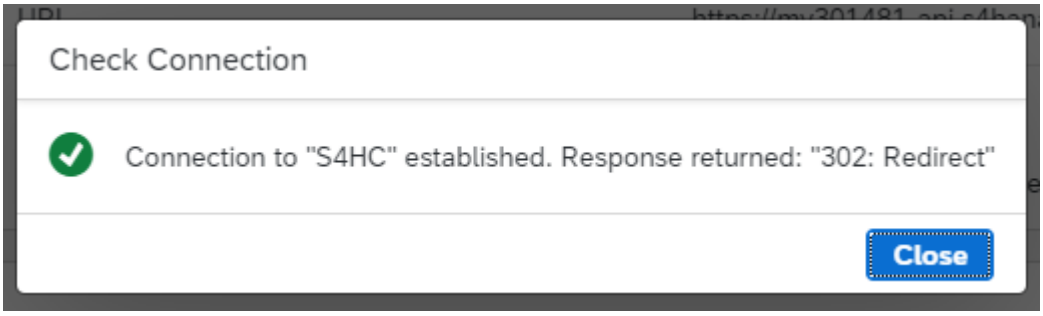
Explanation	Screenshot
In the Kyma Console UI click on **Add new namespace** in the top right corner	
In the dialog, set: Name: **business-partners** Label: **name=business-partners** Click **Create**.	

Explanation	Screenshot
You should see a page like the one in the screenshot.	

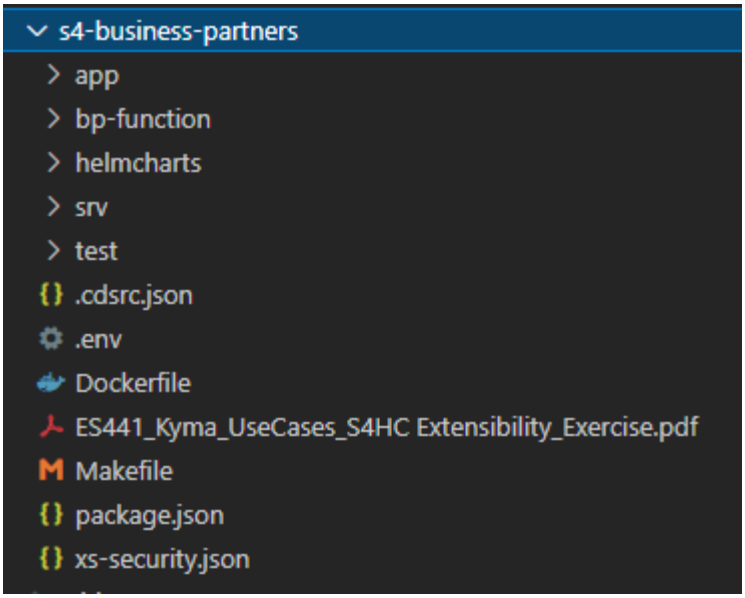
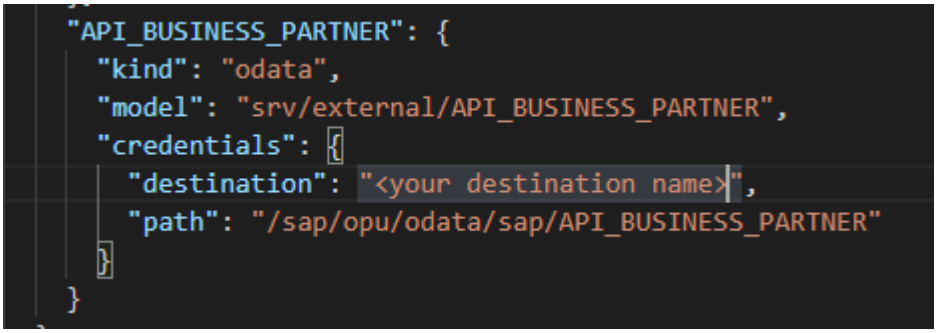
III. Create the destination to S/4HANA Cloud

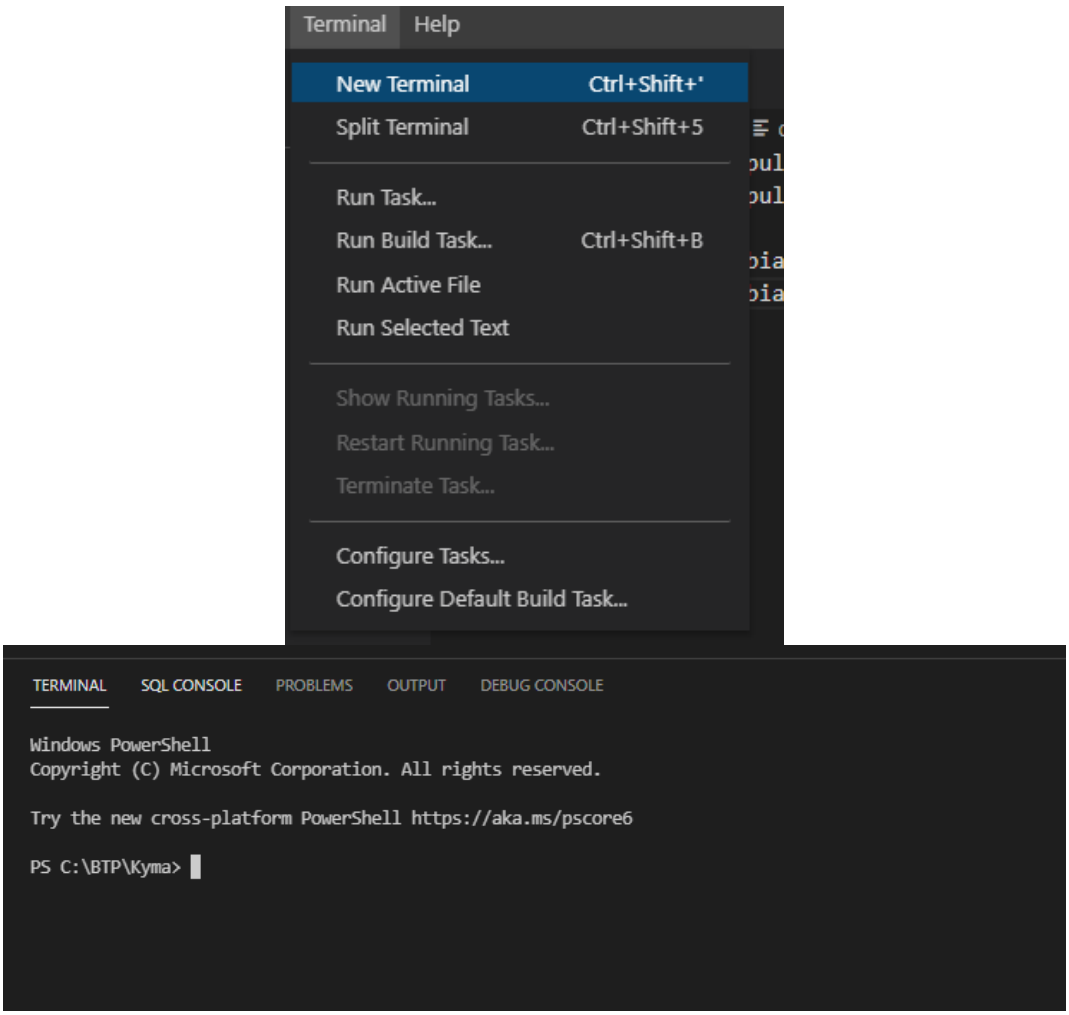
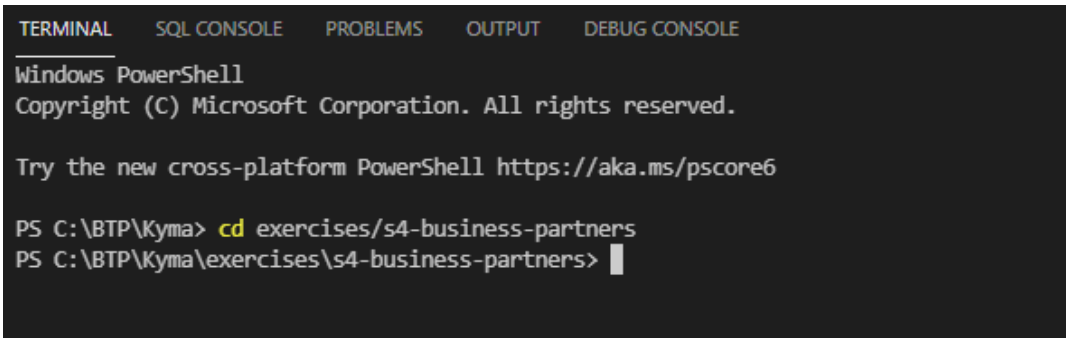
Explanation	Screenshot
Go back to your subaccount cockpit	

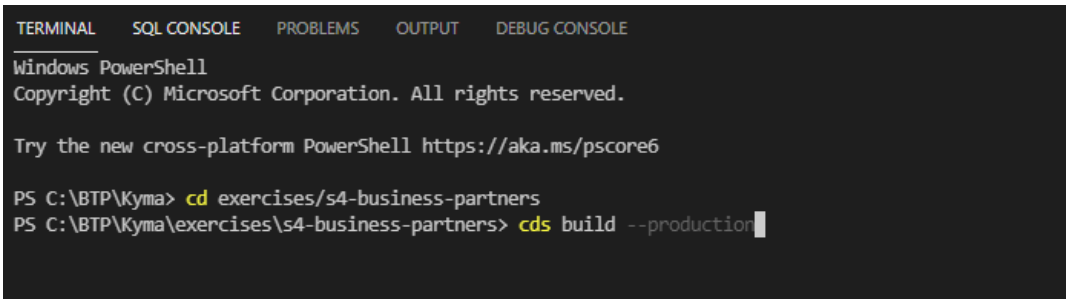
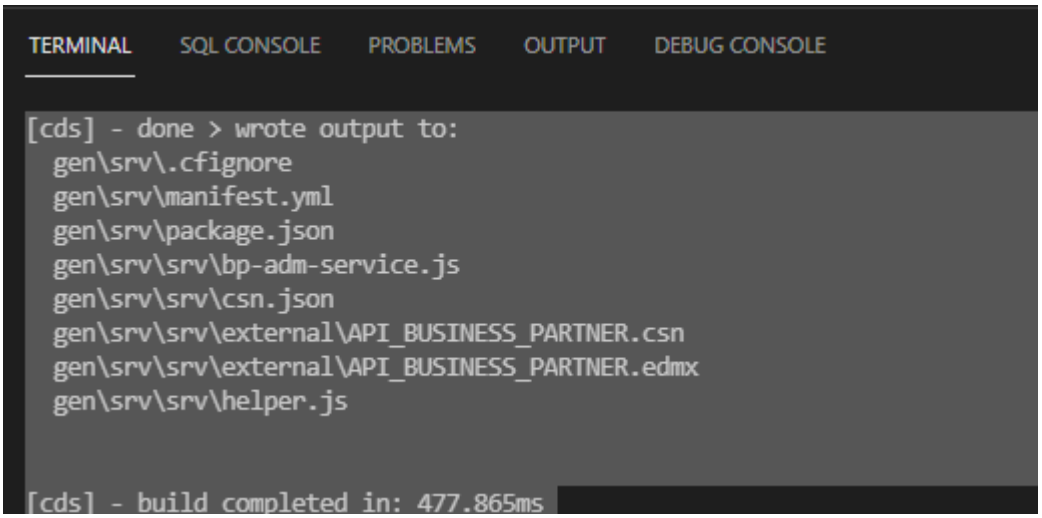
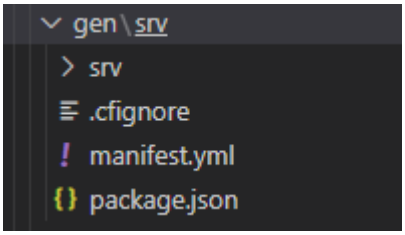
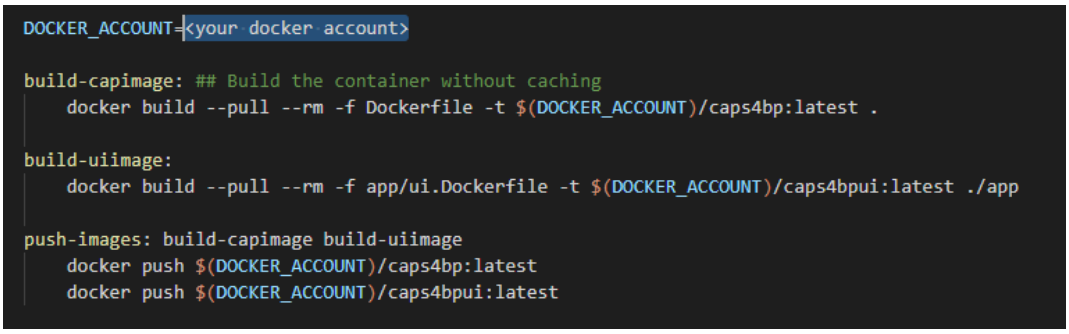
Explanation	Screenshot
In the left-hand side menu expand the **Connectivity** section and click on **Destinations**.	
Click on **New Destination**.	

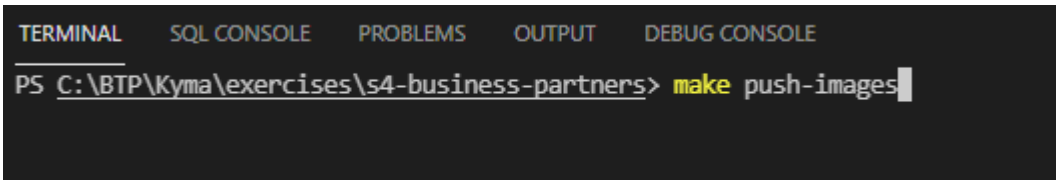
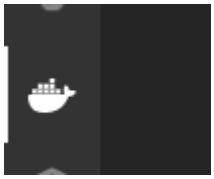
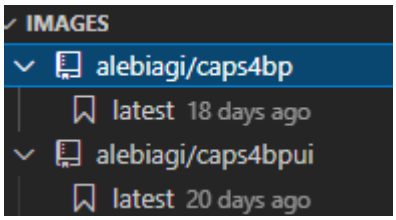
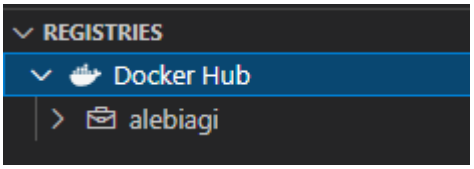
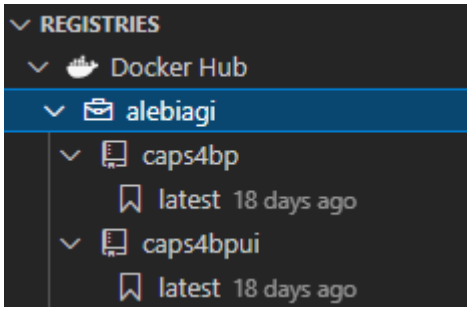
Explanation	Screenshot
Fill-in the destination info accordingly: Name: any name of your choice (i.e. S4HC – just a suggestion) Type: HTTP Description: S/4HANA Cloud URL: URL of your S/4HANA Cloud tenant Proxy Type: Internet Authentication: BasicAuthentication User: username of your communication user Password: password of your communication user Click **Save**.	
In the bottom bar click **Check Connection**.	
If successful, you should see a dialog like the one in the screenshot. Just click **Close**.	

IV. Build and push the Docker images

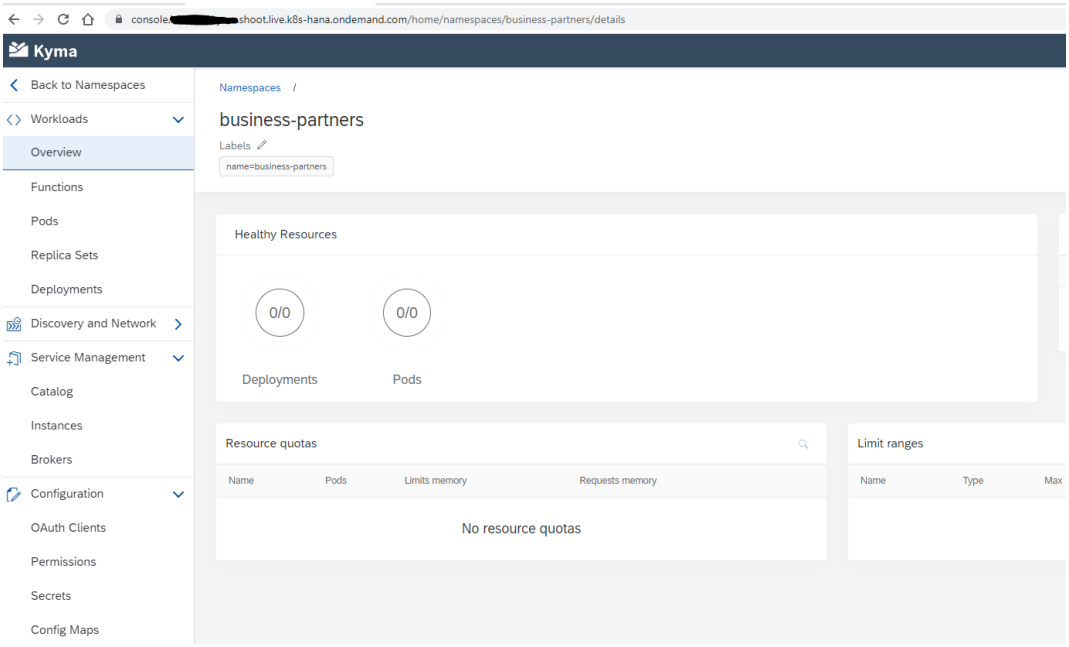
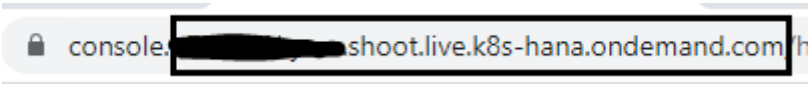
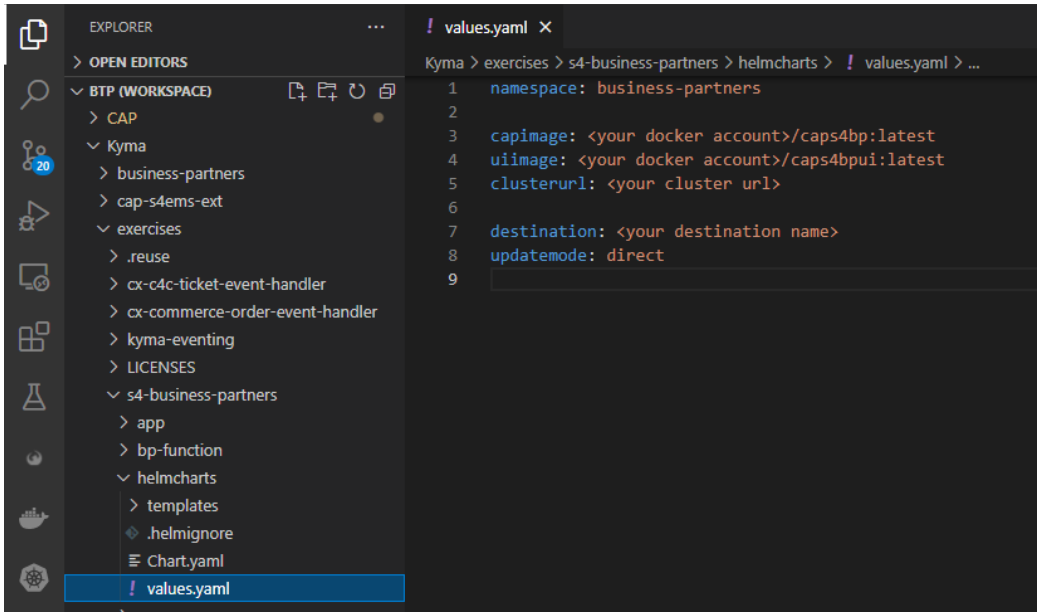
Explanation	Screenshot
Open the **s4-business-partners** folder from the cloned Git Repo in your **VS Code**.	
In the **File Explorer** on the left-hand pane, open the file **package.json** and replace the place holder **<your destination name>** with the name of the destination you created at the previous section (i.e. S4HC). **Save the file**.	

Explanation	Screenshot
From the menu **Terminal** select **New Terminal** to open the command terminal.	
CD into the source code folder of the project **cd <your cloned root>/s4-business-partners**	

Explanation	Screenshot
Build the CAP application with the command: **cds build --production**	
Check the **logs in the console**, as well as the **file explorer** to verify the contents of the folder **gen\srv** has been properly generated.	 
In the **File Explorer** on the left-hand pane, open the **Makefile** and replace the place holder **<your docker account>** with the name of the Docker account you must have created as per the 9th prerequisite of this document. **Save the file**.	

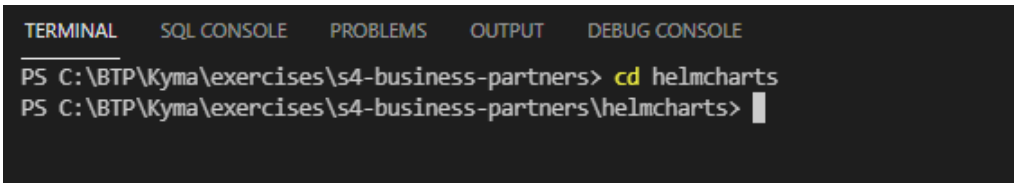
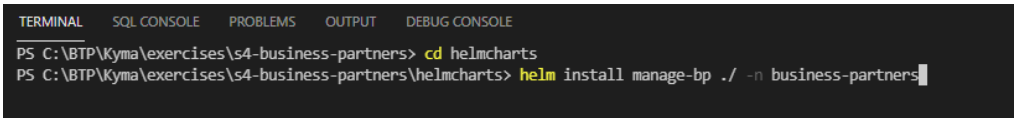
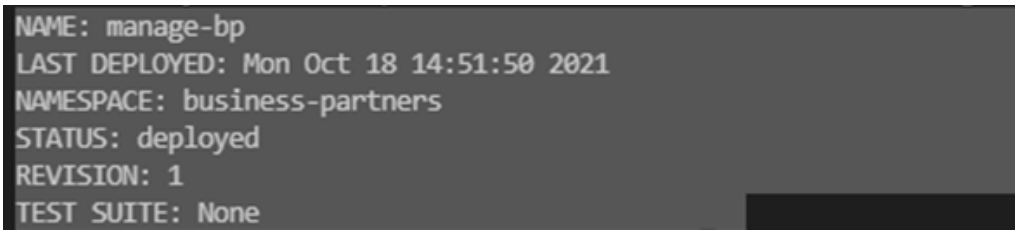
Explanation	Screenshot
Build and push the Docker images with the command: **make push-images**	
When the process is completed access the VS Code Docker extension the left-hand panel clicking on the icon in the screenshot.	
Expand the **Images** node and check whether the two images have been created.	
Expand the **Registries** node and the **Docker Hub** node. You should see your **Docker account** listed like in the screenshot.	
Expand your **Docker account** and check whether the two images have been pushed	

V. Adjust the Helm Charts values

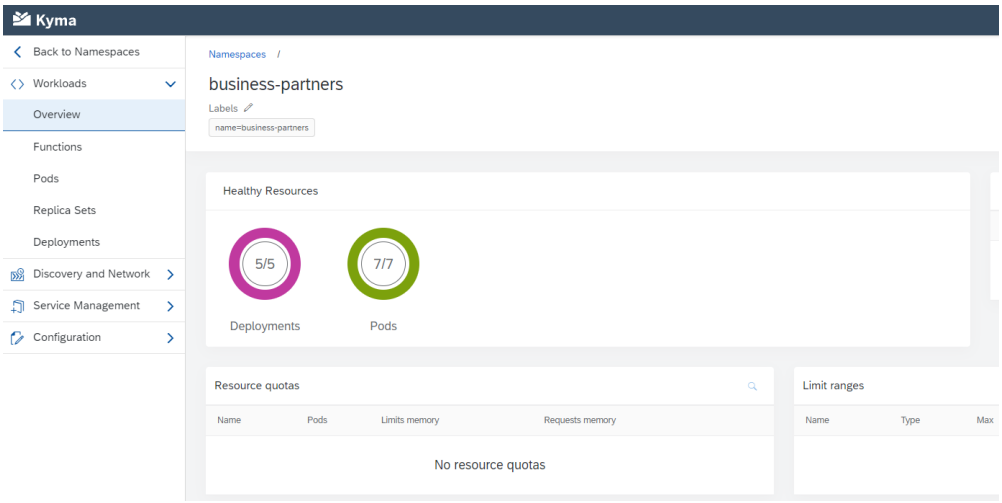
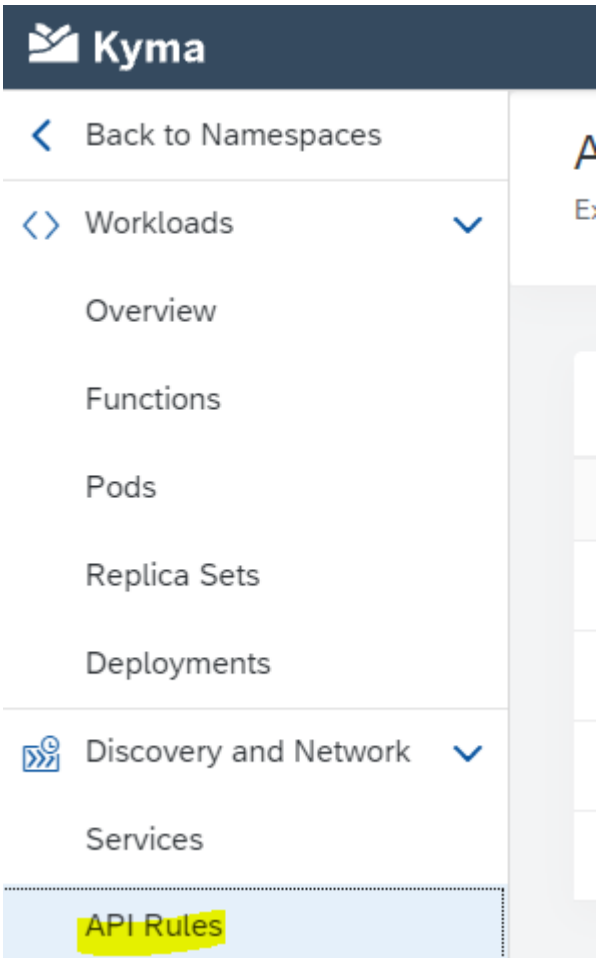
Explanation	Screenshot
Go back to the Kyma Console UI	
Look at the URL in browser and copy the cluster URL (text right after “console.” stopping right before the slash to the clipboard, like demonstrated in the screenshot.	
In the **File Explorer** on the left-hand pane, expand the **helmcharts** folder and open the file **values.yaml**. Replace the place holders accordingly: Over **<your docker account>** write your account name Over **<your cluster url>** paste the value copied in the previous step Over **<your destination name>** write the name of	

Explanation	Screenshot
the destination created at section III (i.e. S4HC). **Save the file**.	

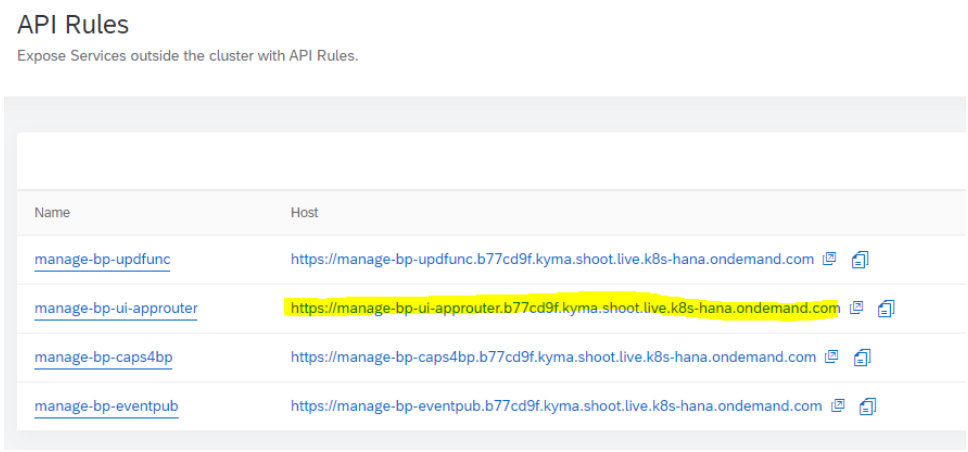
VI. Install the package

Explanation	Screenshot
In the terminal, CD to the **helmcharts** folder.	
Install the package with the command: **helm install manage-bp ./ -n business-partners**	
After some seconds you should see the contents displayed in the screenshot in the logs of your console.	

VII. Test the application

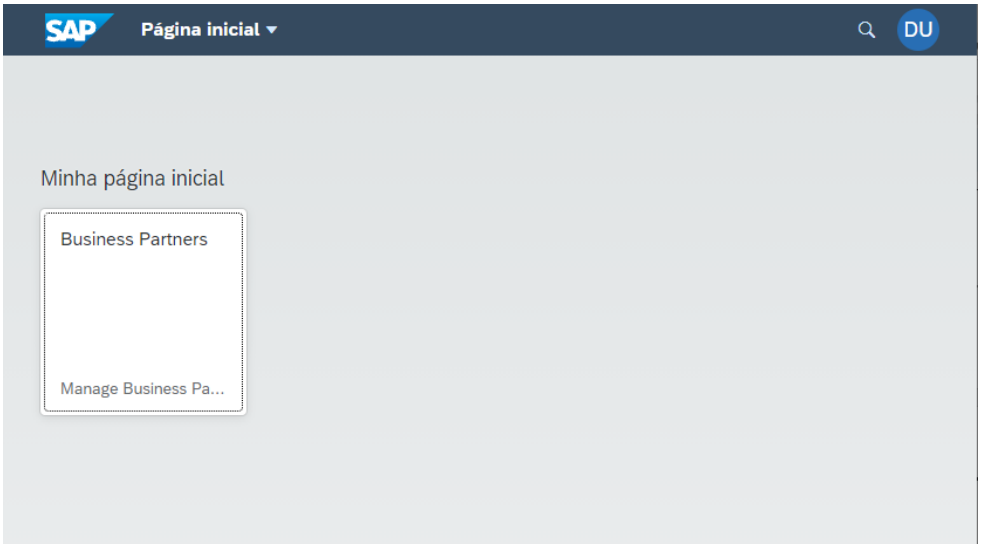
Explanation	Screenshot
It's going to take a couple of minutes until all deployments and pods are running healthy in your Kyma namespace. So, go back to your Kyma Console UI and wait until it shows a status exactly like in the screenshot.	
On the left-hand pane expand **Discovery and Network** and click on **API Rules**	

Click on the **URL** next to the **manage-bp-ui-app-router** API Rule, like demonstrated in the screenshot to open the application.

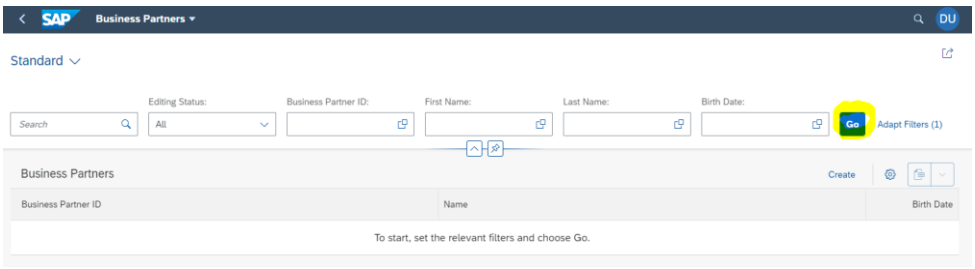


A new tab should be opened in your browser displaying a tile like in the screenshot.

Click on the **Business Partners** tile.

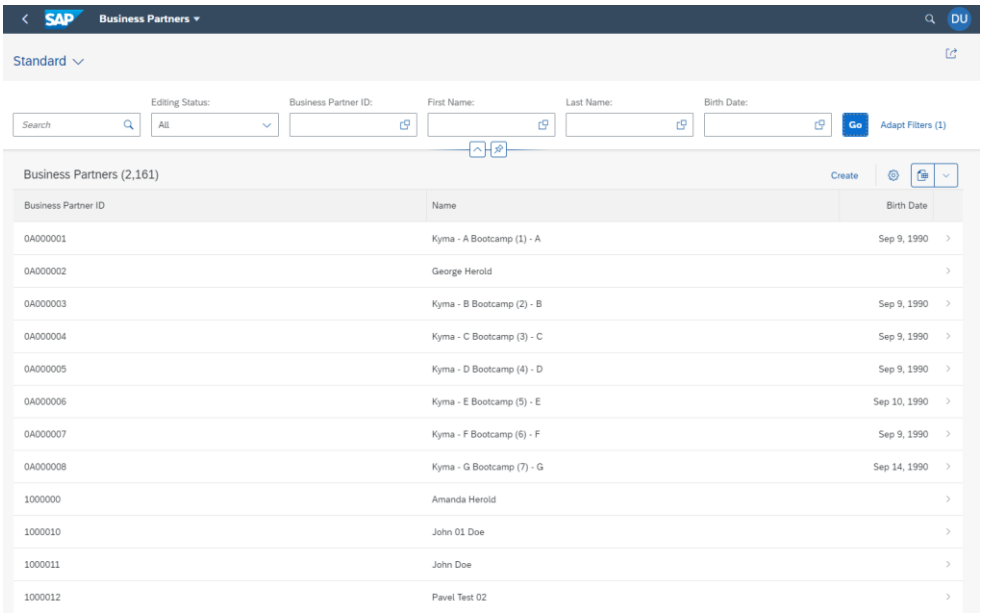


In the next page click on the **Go** button.

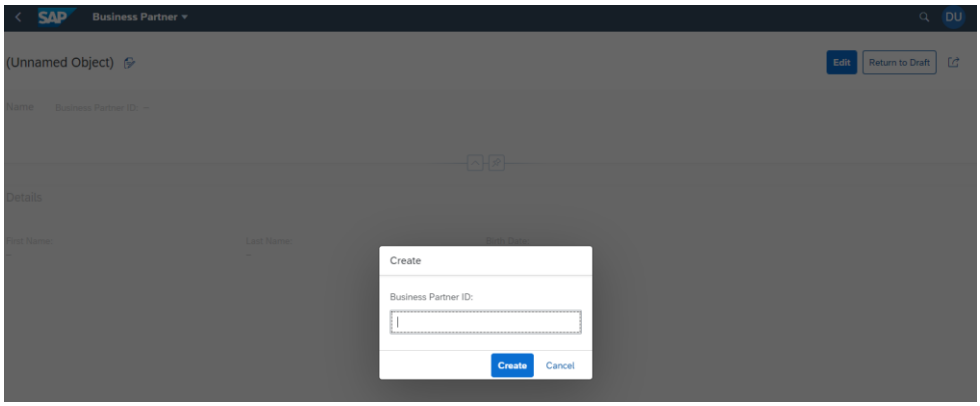


In the first click it's going to take a couple of seconds until the Business Partners list is displayed like in the screenshot. This is due to the in-memory table creation and fill-up at first access.

Click on the **Create** button to create a new Business Partner.

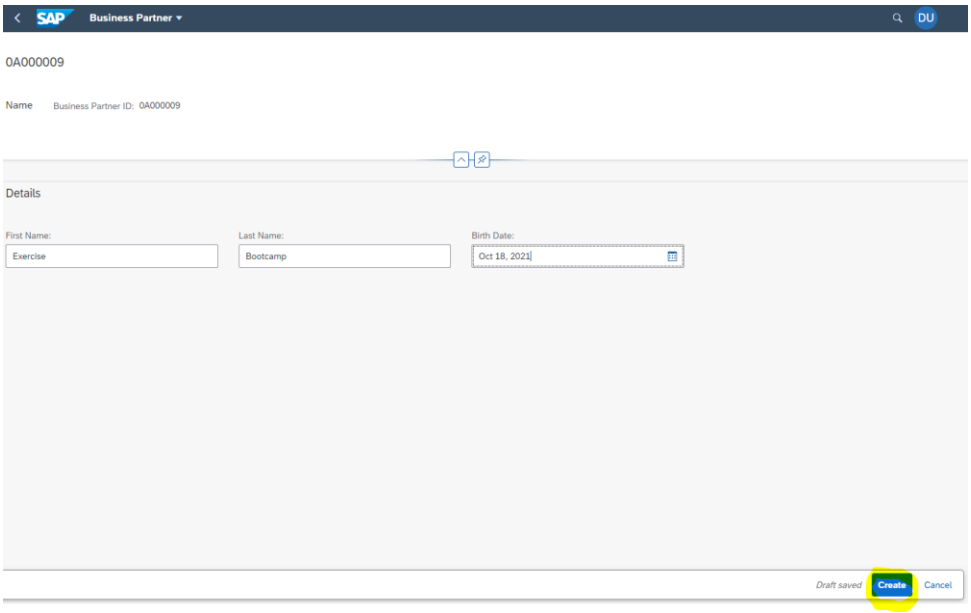


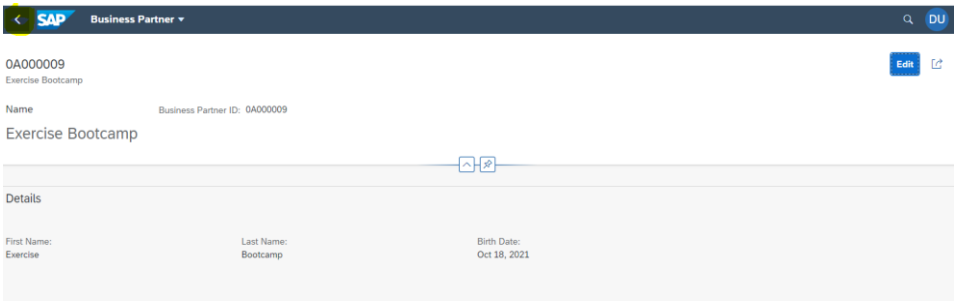
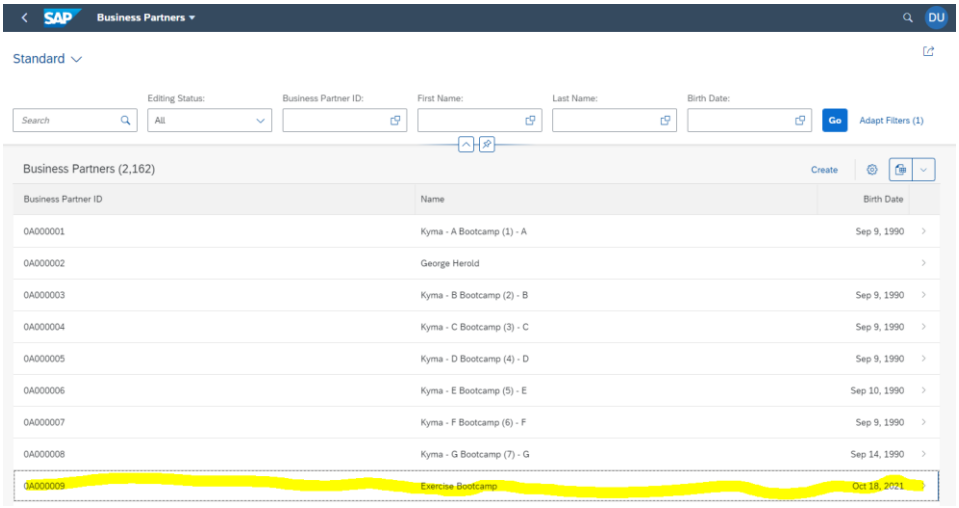
Provide a Business Partner ID that does not exist yet in the S/4 system and click on **Create**.

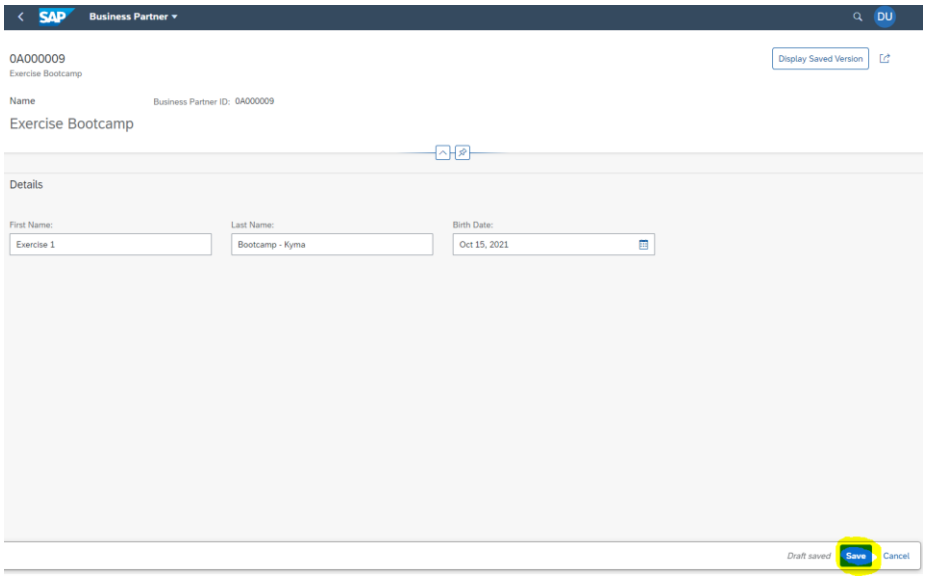
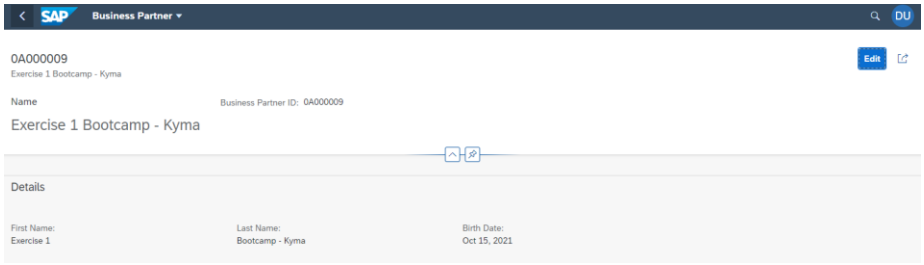
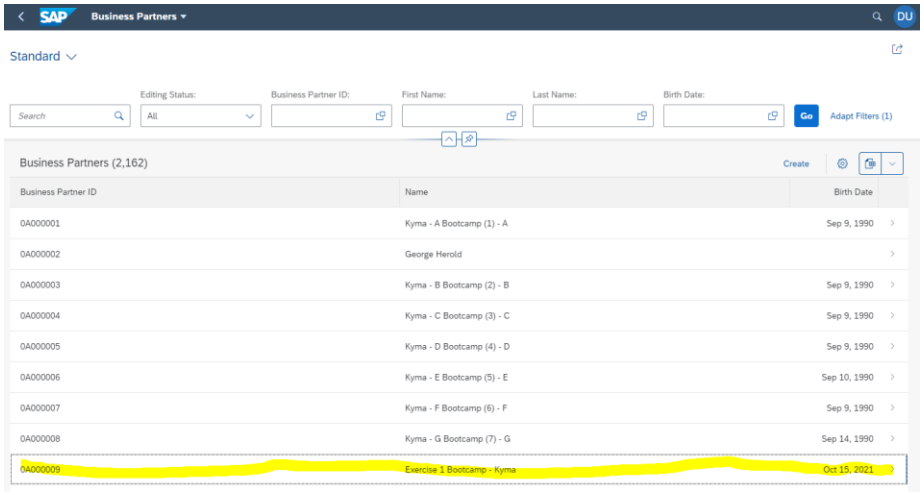


Provide First Name, Last Name and Birth Date.

Click on **Create**.



Check that the Business Partner has been successfully created and click on the arrow at the top left corner to go back to the Business Partners list.	
In the list, locate the newly created Business Partner and click on it.	
Click on the **Edit** button on the top right corner.	

Change the information on the fields at your will and click on **Save**.	
Check that the data has been successfully updated and click on the arrow at the top left corner to go back to the Business Partners list.	
In the list, locate the recently updated record.	

Congratulations! You have successfully completed the exercise! To claim your expert badge, just **submit the exact same screenshots** of this topic taken from your own deployment to the corresponding **MS Teams Channel**.	

APPENDIX

Stretch and challenge yourself!

Now that you have successfully completed the exercise you can go further and try the application in different modes using the **Kyma eventing feature**.

In the **values.yaml** file of the helm charts, just change the **updatemode** variable to **in-cluster** (to publish the even from inside the cluster) or **out-cluster (**to publish the event from outside the cluster).

Then, to update your deployment, CD to the helmcharts folder and run:

helm upgrade manage-bp ./ -n business-partners

After the command successfully completes, repeat the steps in the last block of the exercise to test the application.

Please note that this is **totally optional** thus **will not be required** for the expert badge request.