



BTP SaaS Multitenancy exercise using SAP Cloud Application Programming Model

This document demonstrates the steps to develop, build and deploy a SaaS Multitenant app in BTP Cloud Foundry runtime using SAP Cloud Application Programming (CAP) Model.

www.sap.com/contactsap

© 2018 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies. See www.sap.com/copyright for additional trademark information and notices.

Table of Contents

DISCLAIMER	4
OBJECTIVE	4
SCENARIO	4
SUGGESTED PRE-READS	4
PREREQUISITES	4
SETUP THE PROVIDER ENVIRONMENT.....	4
Create an account	4
Configure SAP HANA Cloud Entitlement on the Provider Subaccount.....	7
Create a HANA Cloud Instance on the Provider Subaccount.....	9
Setup SAP Business Application Studio	13
Install the generator	15
GENERATE THE APPLICATION	16
Generate the app with the generator.....	16
Review the generated application code	21
Build and deploy the app in the Provider Subaccount.	30
Troubleshooting the Deployment.....	33
PREPARING THE CONSUMER ENVIRONMENT	35
Create at least one Consumer Subaccount	35
Subscribe to the Provider from the Tenant / Consumer	36
Assign Roles to the user.....	37
Access the app	38
Troubleshooting the “Requested route ... does not exist” error.....	38
Checking each Consumer is connected to a different HANA HDI Container	40
Viewing the destinations in the Provider subaccount	41
Using a new destination in a Consumer subaccount	41
TEST THE APP.....	42
GENERAL TROUBLESHOOTING.....	46
Troubleshooting general application errors and crashes	46
What to do when I get a red “Subscription failed” message?.....	46
TO GO DEEPER	47

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 beginners.

The objective of this exercise is to demonstrate the steps to develop, build and deploy a SaaS Multitenant app in BTP Cloud Foundry runtime using SAP Cloud Application Programming (CAP) Model. This document also shows how to subscribe to a SaaS Multitenant application as a consumer

SCENARIO

This exercise demonstrates developing, building, and deploying a SaaS Multitenant app and provisioning consumer tenants in BTP Cloud Foundry runtime using SAP Cloud Application Programming (CAP) Model.

SUGGESTED PRE-READS

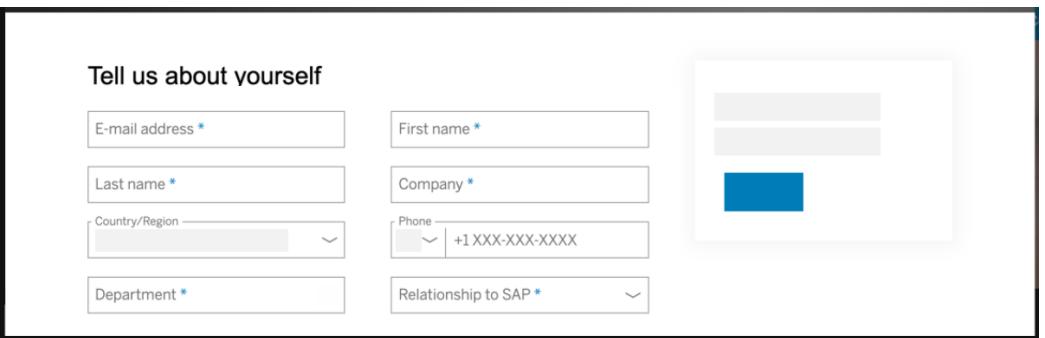
- [Get Ready to Develop on SAP BTP](#)
- [Getting Started in the Cloud Foundry Environment](#)
- [Development in the Cloud Foundry Environment](#)
- [Developing Node.js in the Cloud Foundry Environment](#)
- [Creating a Cloud Foundry Organization and Space](#)
- [Introduction to SAP Cloud Application Programming Model](#)
- [SAP Cloud Application Programming Model documentation](#)
- [Create an SAP Cloud Application Programming Model Project for SAP HANA Cloud](#)
- [Create an application using SAP HANA and the Cloud Application Programming Model](#)

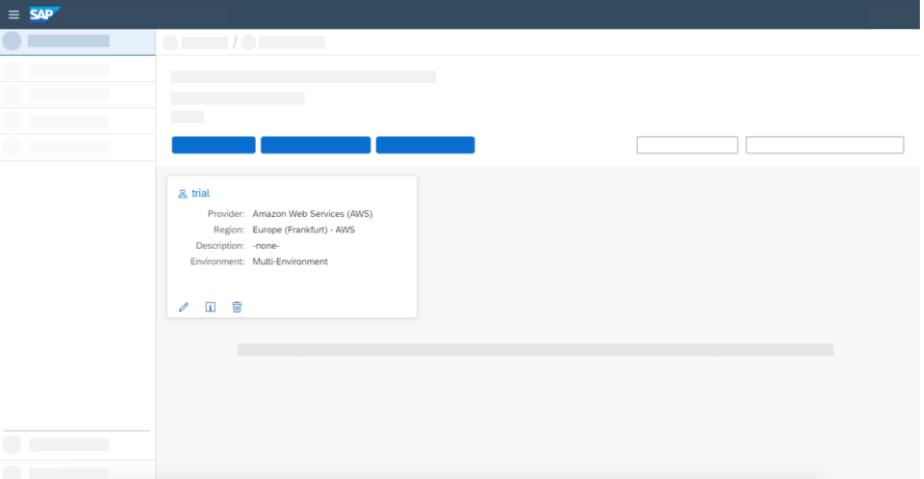
PREREQUISITES

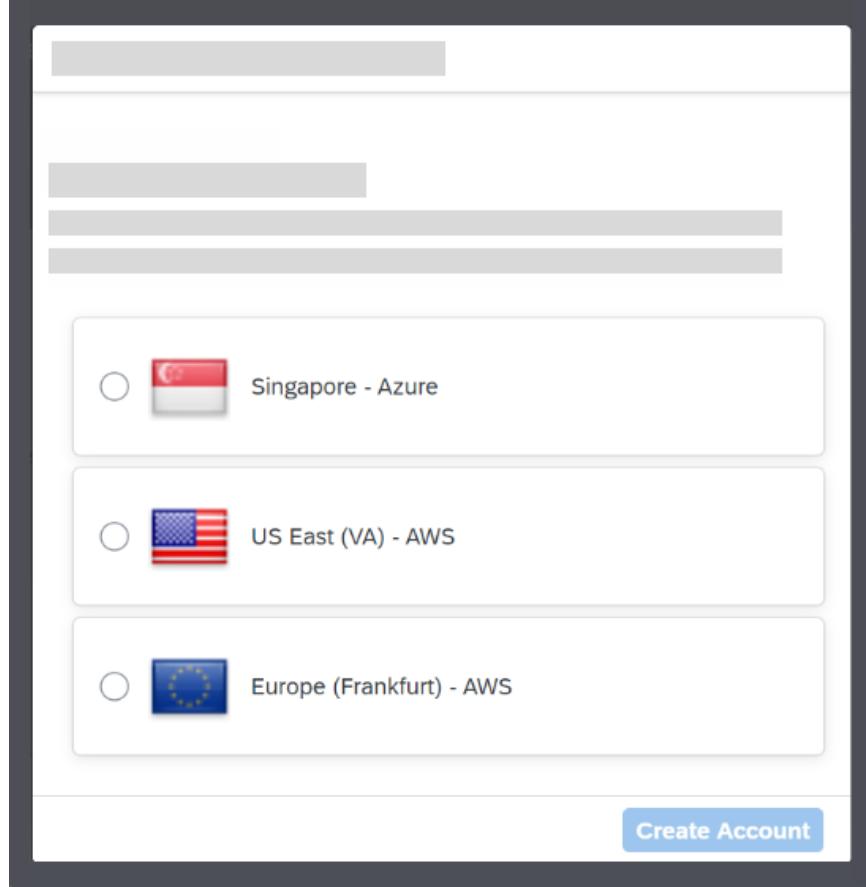
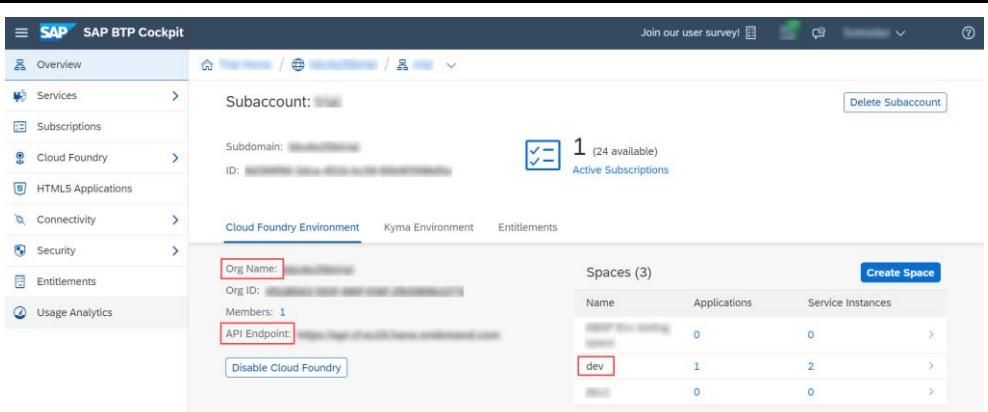
- [Install the Cloud Foundry Command Line Interface \(CLI\)](#)

SETUP THE PROVIDER ENVIRONMENT

Create an account

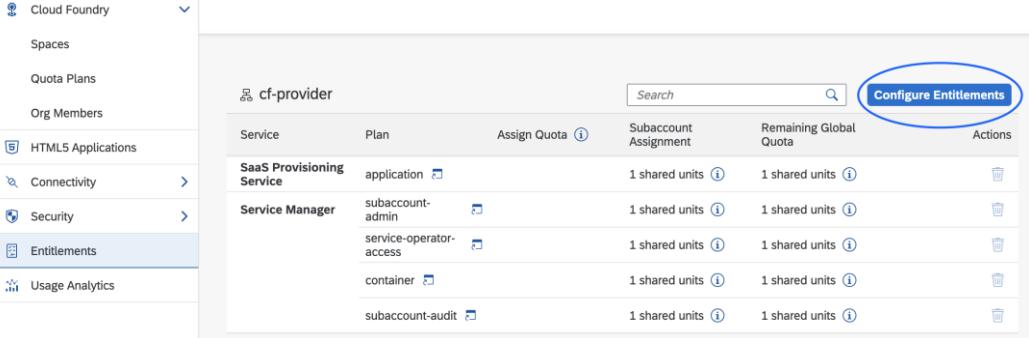
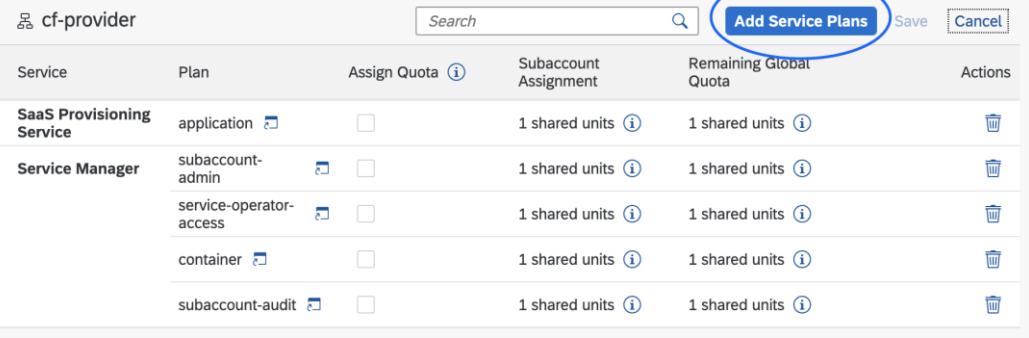
Explanation	Screenshot
Get a Free Account on SAP BTP Trial	 A screenshot of a web form titled "Tell us about yourself". It contains several input fields: "E-mail address *", "First name *", "Last name *", "Company *", "Country/Region" (dropdown), "Phone" (dropdown with value "+1 XXX-XXX-XXXX"), "Department *", and "Relationship to SAP *". There is also a large empty text area and a blue button on the right side of the form.

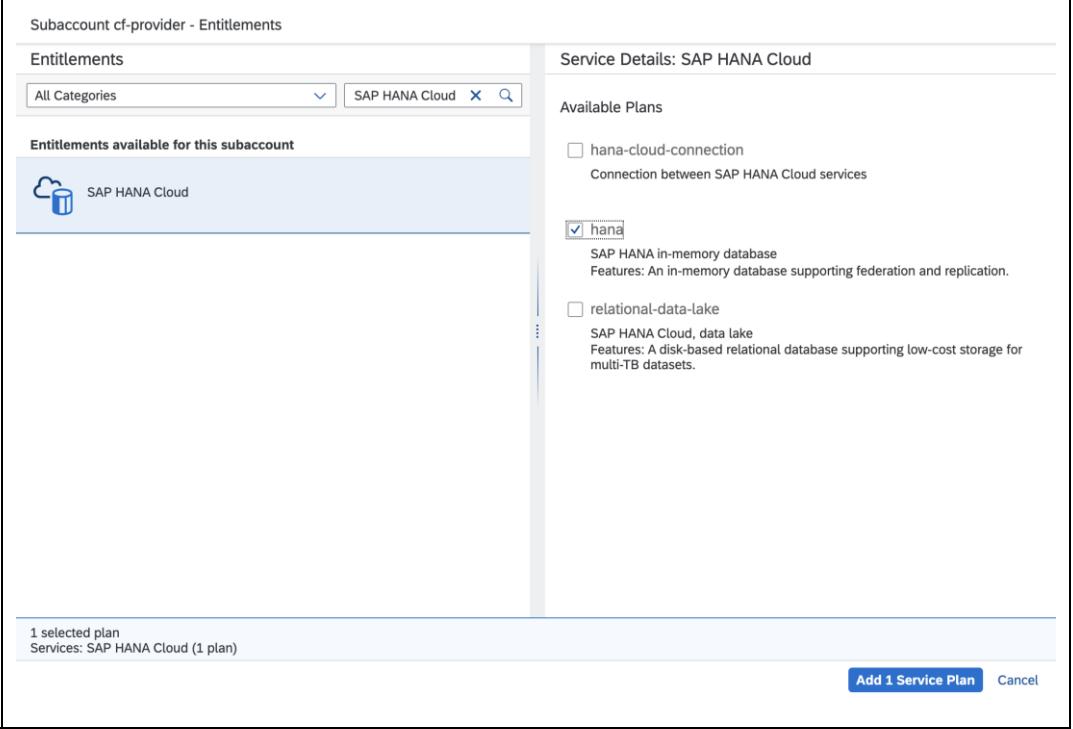
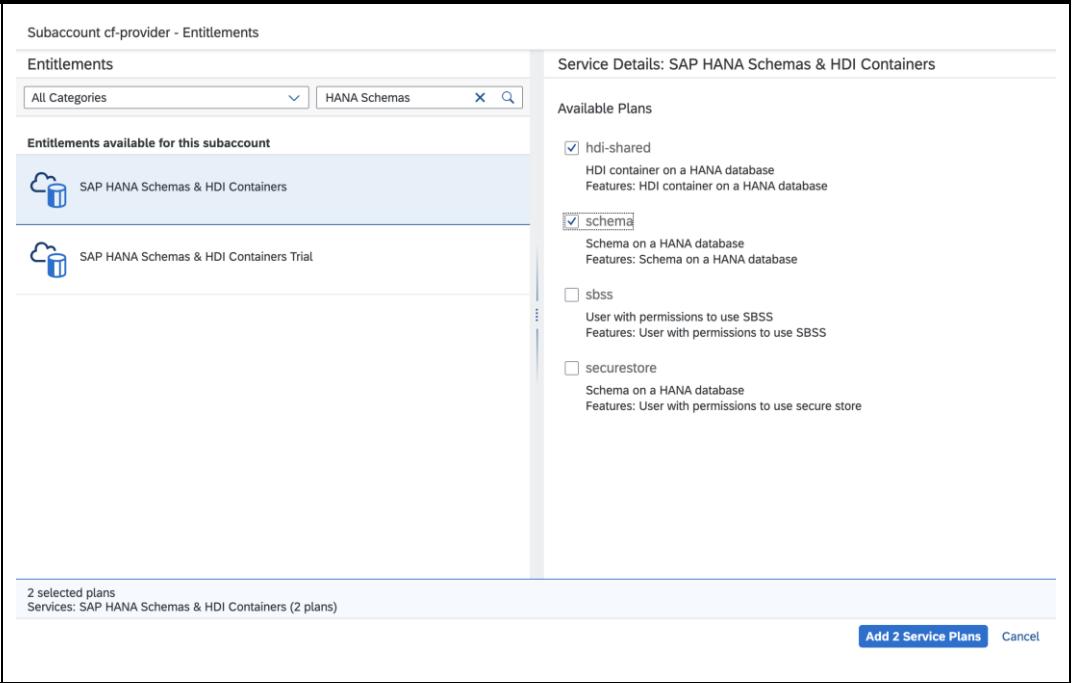
Explanation	Screenshot
<p>The global trial account contains one subaccount.</p> <p>You can navigate to the subaccount by clicking on the tile named trial (this name may vary if you created the subaccount manually).</p> <p>You can use this trial account or create a new subaccount as proposed in the next step.</p> <p>Note: If you're using the subaccount that was generated when you created your trial account, then skip the next step and go to the following step: Create a HANA Cloud Instance on the Provider Subaccount</p>	 <p>The screenshot shows the SAP Fiori Launchpad interface. A subaccount named "trial" is displayed in a card. The card contains the following information:</p> <ul style="list-style-type: none"> Provider: Amazon Web Services (AWS) Region: Europe (Frankfurt) - AWS Description: -none- Environment: Multi-Environment <p>Below the card are three blue navigation buttons. The background of the launchpad shows other subaccounts as small cards.</p>

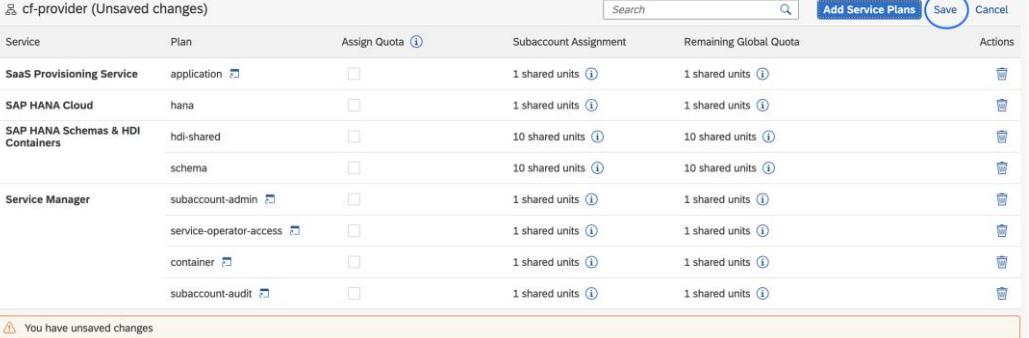
Explanation	Screenshot
<p>Create a Provider Subaccount</p> <p>You can create a subaccount where your application as provider will be deployed.</p> <p>Each subaccount lives in a geographic region. Choose any region from this list and click Create Account to trigger the provisioning process of a new subaccount.</p>	
<p>Enable Cloud Foundry Environment in the Provider Account and create an Organization and a Space – This is done automatically for you when you get a Free Account on SAP BTP trial</p> <p>- Creating a Cloud Foundry Organization and Space / Create a Cloud Foundry Space</p> <p>Important Note: Please make sure the Org Name only contain small letters and numbers. Please avoid underscores or other special characters as problems may raise</p>	

Explanation	Screenshot
afterwards when working with the provisioning service as the Org Name will be used to identify the subaccount.	

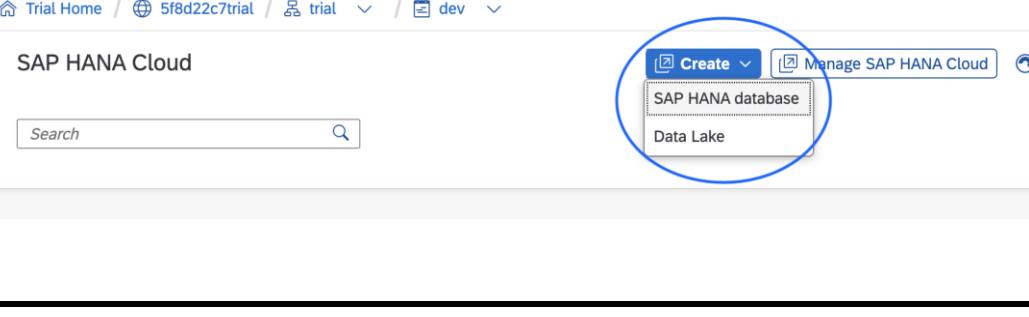
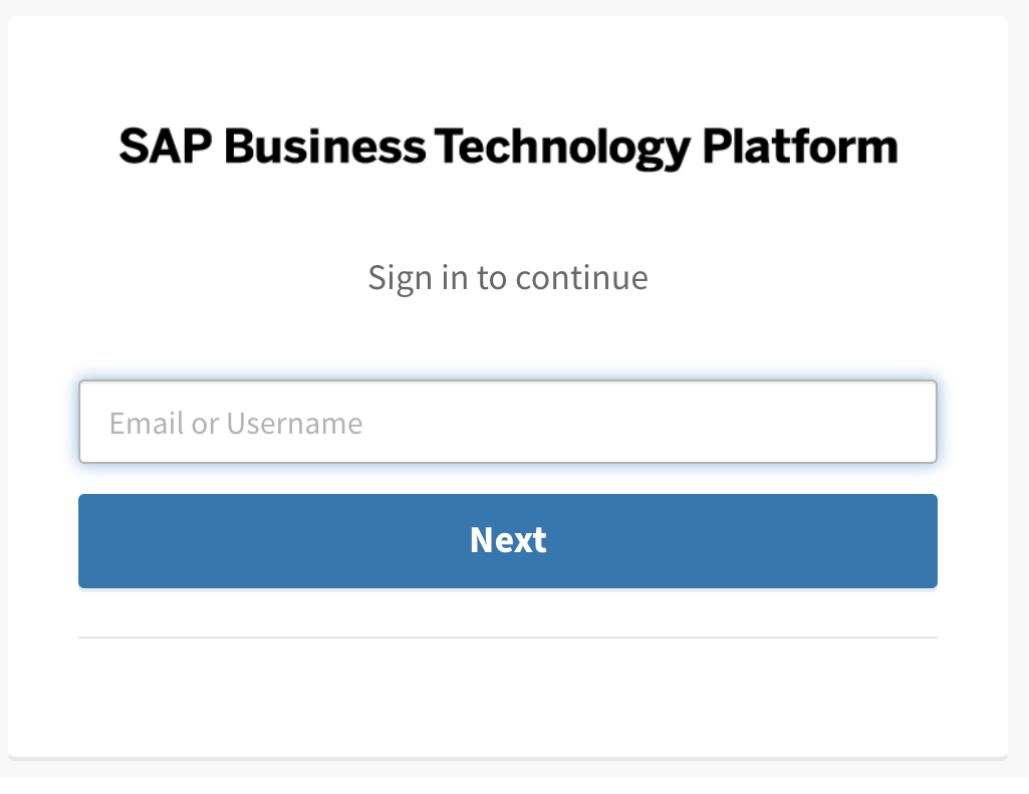
Configure SAP HANA Cloud Entitlement on the Provider Subaccount

Explanation	Screenshot																																	
<p>Go to your Provider Subaccount.</p> <p>Select Entitlements >> Configure Entitlements</p>	 <table border="1"> <thead> <tr> <th>Service</th> <th>Plan</th> <th>Assign Quota</th> <th>Subaccount Assignment</th> <th>Remaining Global Quota</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td>SaaS Provisioning Service</td> <td>application</td> <td><input type="checkbox"/></td> <td>1 shared units</td> <td>1 shared units</td> <td>trash</td> </tr> <tr> <td rowspan="3">Service Manager</td> <td>subaccount-admin</td> <td><input type="checkbox"/></td> <td>1 shared units</td> <td>1 shared units</td> <td>trash</td> </tr> <tr> <td>service-operator-access</td> <td><input type="checkbox"/></td> <td>1 shared units</td> <td>1 shared units</td> <td>trash</td> </tr> <tr> <td>container</td> <td><input type="checkbox"/></td> <td>1 shared units</td> <td>1 shared units</td> <td>trash</td> </tr> <tr> <td>subaccount-audit</td> <td><input type="checkbox"/></td> <td>1 shared units</td> <td>1 shared units</td> <td>trash</td> </tr> </tbody> </table>	Service	Plan	Assign Quota	Subaccount Assignment	Remaining Global Quota	Actions	SaaS Provisioning Service	application	<input type="checkbox"/>	1 shared units	1 shared units	trash	Service Manager	subaccount-admin	<input type="checkbox"/>	1 shared units	1 shared units	trash	service-operator-access	<input type="checkbox"/>	1 shared units	1 shared units	trash	container	<input type="checkbox"/>	1 shared units	1 shared units	trash	subaccount-audit	<input type="checkbox"/>	1 shared units	1 shared units	trash
Service	Plan	Assign Quota	Subaccount Assignment	Remaining Global Quota	Actions																													
SaaS Provisioning Service	application	<input type="checkbox"/>	1 shared units	1 shared units	trash																													
Service Manager	subaccount-admin	<input type="checkbox"/>	1 shared units	1 shared units	trash																													
	service-operator-access	<input type="checkbox"/>	1 shared units	1 shared units	trash																													
	container	<input type="checkbox"/>	1 shared units	1 shared units	trash																													
subaccount-audit	<input type="checkbox"/>	1 shared units	1 shared units	trash																														
Click on “Add Service Plans”	 <table border="1"> <thead> <tr> <th>Service</th> <th>Plan</th> <th>Assign Quota</th> <th>Subaccount Assignment</th> <th>Remaining Global Quota</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td>SaaS Provisioning Service</td> <td>application</td> <td><input type="checkbox"/></td> <td>1 shared units</td> <td>1 shared units</td> <td>trash</td> </tr> <tr> <td rowspan="3">Service Manager</td> <td>subaccount-admin</td> <td><input type="checkbox"/></td> <td>1 shared units</td> <td>1 shared units</td> <td>trash</td> </tr> <tr> <td>service-operator-access</td> <td><input type="checkbox"/></td> <td>1 shared units</td> <td>1 shared units</td> <td>trash</td> </tr> <tr> <td>container</td> <td><input type="checkbox"/></td> <td>1 shared units</td> <td>1 shared units</td> <td>trash</td> </tr> <tr> <td>subaccount-audit</td> <td><input type="checkbox"/></td> <td>1 shared units</td> <td>1 shared units</td> <td>trash</td> </tr> </tbody> </table>	Service	Plan	Assign Quota	Subaccount Assignment	Remaining Global Quota	Actions	SaaS Provisioning Service	application	<input type="checkbox"/>	1 shared units	1 shared units	trash	Service Manager	subaccount-admin	<input type="checkbox"/>	1 shared units	1 shared units	trash	service-operator-access	<input type="checkbox"/>	1 shared units	1 shared units	trash	container	<input type="checkbox"/>	1 shared units	1 shared units	trash	subaccount-audit	<input type="checkbox"/>	1 shared units	1 shared units	trash
Service	Plan	Assign Quota	Subaccount Assignment	Remaining Global Quota	Actions																													
SaaS Provisioning Service	application	<input type="checkbox"/>	1 shared units	1 shared units	trash																													
Service Manager	subaccount-admin	<input type="checkbox"/>	1 shared units	1 shared units	trash																													
	service-operator-access	<input type="checkbox"/>	1 shared units	1 shared units	trash																													
	container	<input type="checkbox"/>	1 shared units	1 shared units	trash																													
subaccount-audit	<input type="checkbox"/>	1 shared units	1 shared units	trash																														

Explanation	Screenshot
<p>Search for “SAP HANA Cloud”, select “hana” and click on “Add 1 Service Plan”.</p>	 <p>The screenshot shows the SAP HANA Cloud service provider interface. In the search bar, "SAP HANA Cloud" is entered. Below the search bar, a section titled "Entitlements available for this subaccount" lists "SAP HANA Cloud". On the right, under "Available Plans", the "hana" plan is selected (indicated by a checked checkbox). Other plans like "relational-data-lake" are also listed but not selected. At the bottom, it shows "1 selected plan" and "Services: SAP HANA Cloud (1 plan)".</p>
<p>Search for “HANA Schemas”. Select “hdi-shared” and “schema” plans. Click on “Add 2 Service Plans”.</p>	 <p>The screenshot shows the SAP HANA Schemas & HDI Containers service provider interface. In the search bar, "HANA Schemas" is entered. Below the search bar, a section titled "Entitlements available for this subaccount" lists "SAP HANA Schemas & HDI Containers Trial". On the right, under "Available Plans", the "hdi-shared" and "schema" plans are selected (indicated by checked checkboxes). Other plans like "sbss" and "securestore" are listed but not selected. At the bottom, it shows "2 selected plans" and "Services: SAP HANA Schemas & HDI Containers (2 plans)".</p>

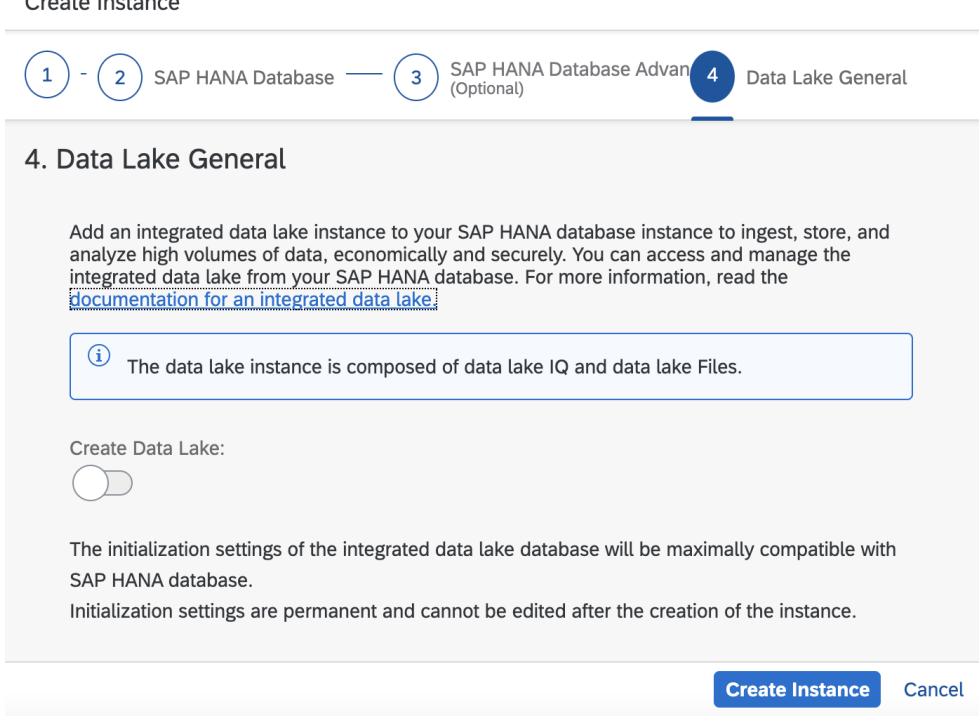
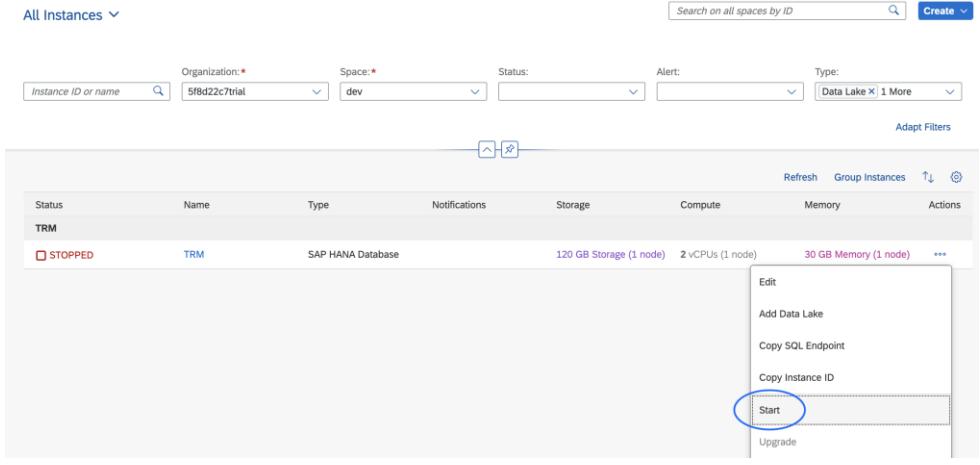
Explanation	Screenshot
Click on “Save”.	 <p>The screenshot shows a table of service quotas. The columns are: Service, Plan, Assign Quota, Subaccount Assignment, Remaining Global Quota, and Actions. Services listed include SaaS Provisioning Service, SAP HANA Cloud, SAP HANA Schemas & HDI Containers, Service Manager, and several subaccounts. Most rows have an 'Assign Quota' checkbox. The 'Remaining Global Quota' column shows values like 1 shared unit, 10 shared units, etc. The 'Actions' column has icons for edit and delete. A status bar at the bottom says 'You have unsaved changes'.</p>

Create a HANA Cloud Instance on the Provider Subaccount

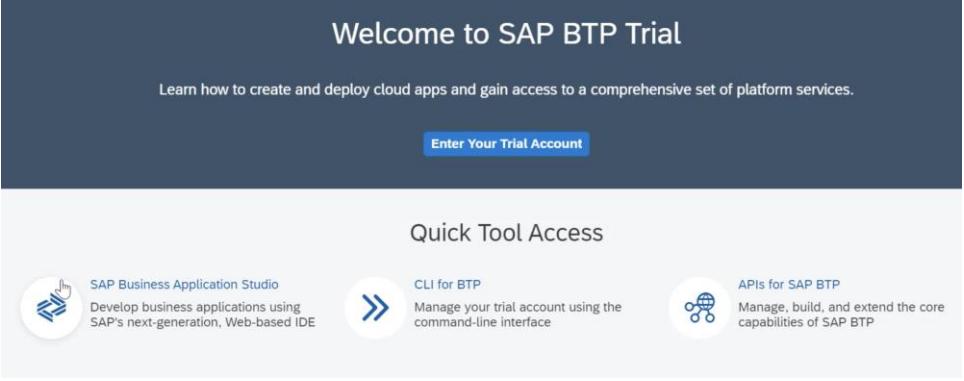
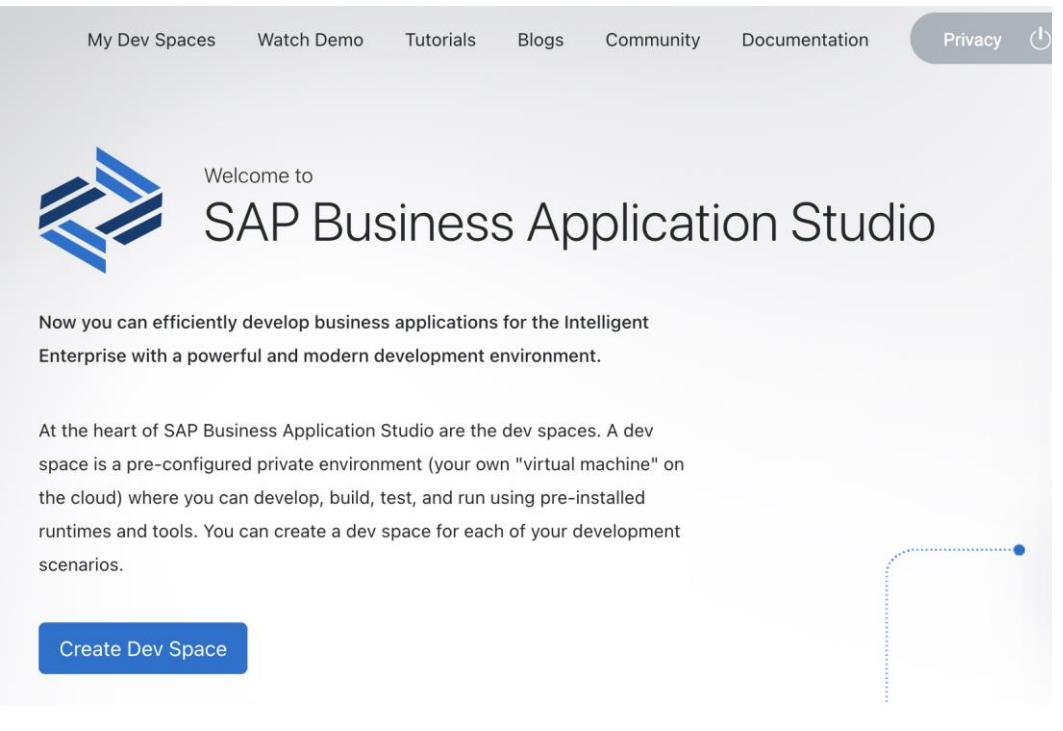
Explanation	Screenshot
<p>Go to your Cloud Foundry space and click on “SAP HANA Cloud” left hand side menu.</p> <p>Click Create >> “SAP HANA database”</p>	 <p>The screenshot shows a navigation bar with Trial Home, 5f8d22c7trial, trial, dev. Below it is a search bar and a 'Create' button. A dropdown menu is open under 'Create' with two options: 'SAP HANA database' and 'Data Lake'. The 'SAP HANA database' option is highlighted with a blue box and a circle.</p>
Sign in to continue.	 <p>The screenshot shows a sign-in form for SAP Business Technology Platform. It includes a header with the SAP logo, a 'Sign in to continue' button, an 'Email or Username' input field, and a large blue 'Next' button.</p>

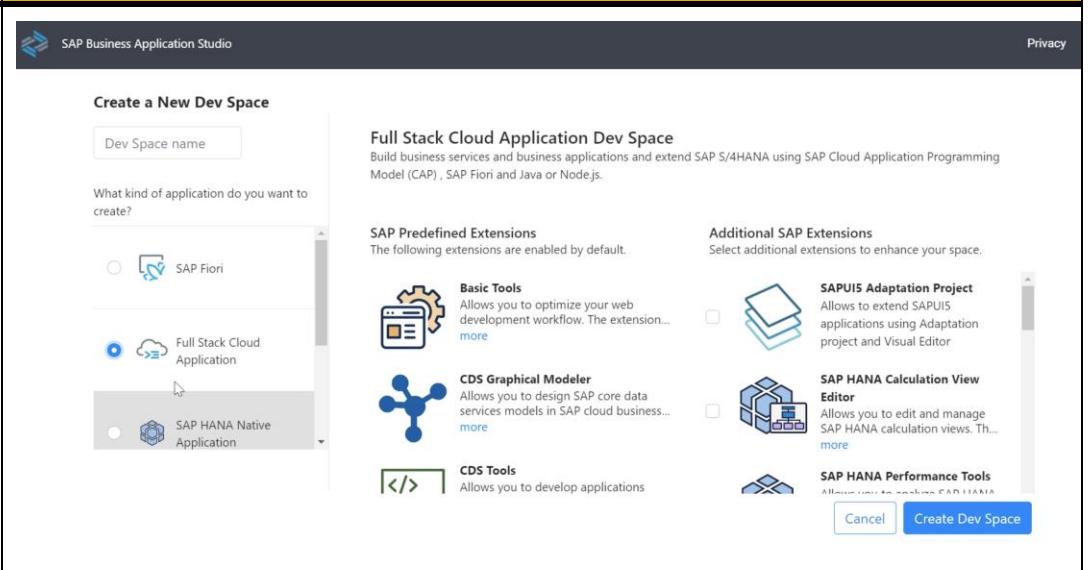
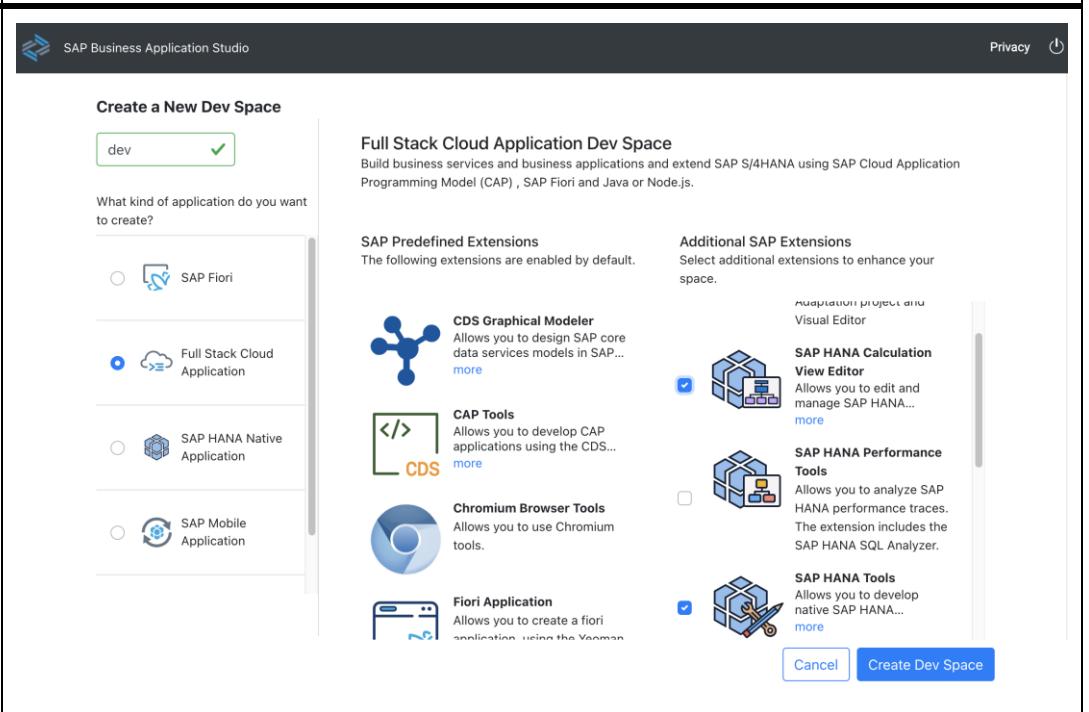
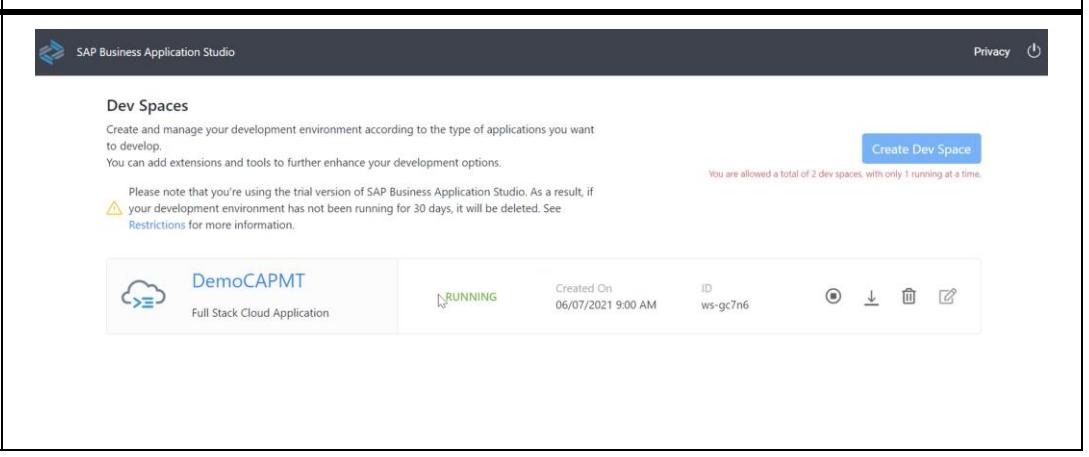
Explanation	Screenshot
<p>Enter an Instance Name, for example, “hana” and an Administrator Password.</p> <p>Important Note: For the Instance Name Please use ONLY small letters without any special characters to avoid further issues on the auto-generated application services.</p> <p>Click “Step 2”.</p>	<p>Create Instance</p> <p>1 Location and Basics 2 SAP HANA Database 3 SAP HANA Database Adv... (Optional) 4 Data Lake General</p> <p>Basics</p> <p>Provide a name and description for your instance.</p> <p>Instance Name: * <input type="text" value="hana"/> ⓘ</p> <p>Description: <input type="text"/> 40 characters remaining</p> <p>The administration user DBADMIN is automatically created with the instance. Enter a password for this account. You use this user to log on to the SAP HANA cockpit and perform all initial user administration tasks.</p> <p>User: DBADMIN</p> <p>Administrator Password: * <input type="password"/> ⓘ</p> <p>Confirm Administrator Password: * <input type="password"/> ⓘ</p> <p>If you choose to add an integrated data lake instance, both DBADMIN and HDLADMIN are automatically created. HDLADMIN is initially assigned the same password as DBADMIN but going forward, both are treated as separate users.</p> <p>Step 2</p> <p>Cancel</p>

Explanation	Screenshot
<p>Select “Allow all IP addresses”.</p>	<p>Create Instance</p>  <p>3. SAP HANA Database Advanced Settings</p> <p>Refine and customize your instance setup.</p> <p>Connections</p> <p>Select who can have access to your SAP HANA database instance.</p> <p>You can choose to only allow access to the instance from Business Technology Platform (default), allow all access, or only from trusted IP addresses by specifying IP address filter ranges.</p> <div style="border: 1px solid orange; padding: 10px;"> <p>⚠ Changing the allowed IP addresses in SAP HANA database does not automatically sync the new settings to data lake. You have to edit data lake settings individually for them to match.</p> </div> <p>Allowed connections:</p> <ul style="list-style-type: none"> <input type="radio"/> Deny all IP addresses (except Business Technology Platform) <input checked="" type="radio"/> Allow all IP addresses <input type="radio"/> Allow only specific IP addresses and IP ranges (in addition to Business Technology Platform) (i) <p>Select whether you want your SAP HANA database to connect to your on-premise remote sources through the cloud connector. For details, see the SAP HANA Database Connectivity Documentation.</p> <p>Enable: <input type="checkbox"/></p>

Explanation	Screenshot
<p>Leave “Create Data Lake” disabled and click on “Create Instance”.</p>	 <p>The screenshot shows the “Create Instance” wizard. Step 1 is “SAP HANA Database”, Step 2 is “SAP HANA Database Advanced”, Step 3 is “(Optional)”, and Step 4 is “Data Lake General”. The “Data Lake General” tab is highlighted. Below it, the “4. Data Lake General” section is visible, containing text about adding an integrated data lake instance and a note that the data lake instance is composed of data lake IQ and data lake Files.</p>
<p>After creating the instance, click on “Manage SAP HANA Cloud”, and get it started.</p>	 <p>The screenshot shows the SAP HANA Cloud management interface. The URL is Trial Home / 059820e6trial / Provider / dev. The “SAP HANA Cloud” section has a “Create” button and a “Manage SAP HANA Cloud” button, with the latter being highlighted.</p>
<p>Ensure that the instance is running and start it from the Actions dropdown menu if it is stopped.</p> <p>Note: In a trial account the HANA instance will be stopped automatically after some hours.</p>	 <p>The screenshot shows the “All Instances” list. A search bar and filter options are at the top. The instance “TRM” is listed, showing it is “STOPPED”. An “Actions” dropdown menu is open next to the instance, with the “Start” option highlighted with a blue circle.</p>

Setup SAP Business Application Studio

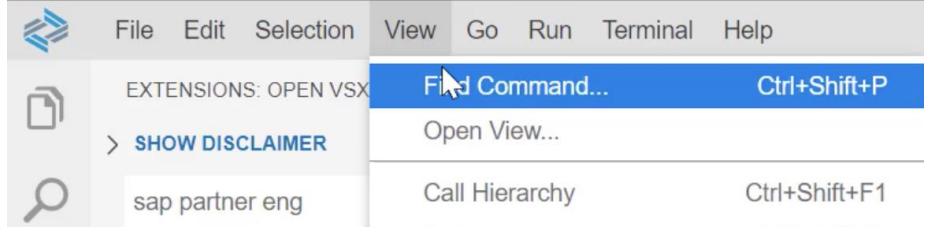
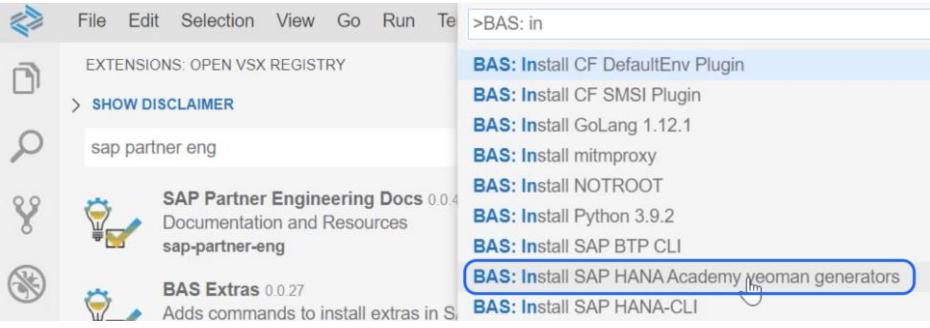
Explanation	Screenshot
Ensure that you have SAP Business Application Studio enabled, which is the IDE that will be used to develop the application.	 The screenshot shows the "Welcome to SAP BTP Trial" page. It features a dark header with the title and a subtext: "Learn how to create and deploy cloud apps and gain access to a comprehensive set of platform services." A blue button labeled "Enter Your Trial Account" is centered. Below the header is a section titled "Quick Tool Access" containing three items: "SAP Business Application Studio" (with a brief description and icon), "CLI for BTP" (with a brief description and icon), and "APIs for SAP BTP" (with a brief description and icon).
Create a new development space by clicking on " Create Dev Space ". You can have several workspaces in SAP Business Application Studio.	 The screenshot shows the SAP Business Application Studio home page. At the top, there's a navigation bar with links: "My Dev Spaces", "Watch Demo", "Tutorials", "Blogs", "Community", "Documentation", "Privacy", and a power icon. The main content area features the SAP logo and the text "Welcome to SAP Business Application Studio". Below this, a subtext reads: "Now you can efficiently develop business applications for the Intelligent Enterprise with a powerful and modern development environment." A larger text block explains the purpose of dev spaces. At the bottom left is a blue button labeled "Create Dev Space". A dotted line with an arrow points from the "Create Dev Space" button towards the "Create Dev Space" link in the explanation text.

Explanation	Screenshot
<p>Choose “Full Stack Cloud Application” as application kind.</p>	
<p>Select the SAP HANA Calculation View Editor and SAP HANA Tools as additional extensions.</p>	
<p>Give the Dev Space a name and click on “Create Dev Space”.</p> <p>Wait for the Dev Space to be provisioned, it will be marked RUNNING when ready.</p> <p>You can open your Dev Space with a click on</p>	

Explanation	Screenshot
your Dev Space name link.	

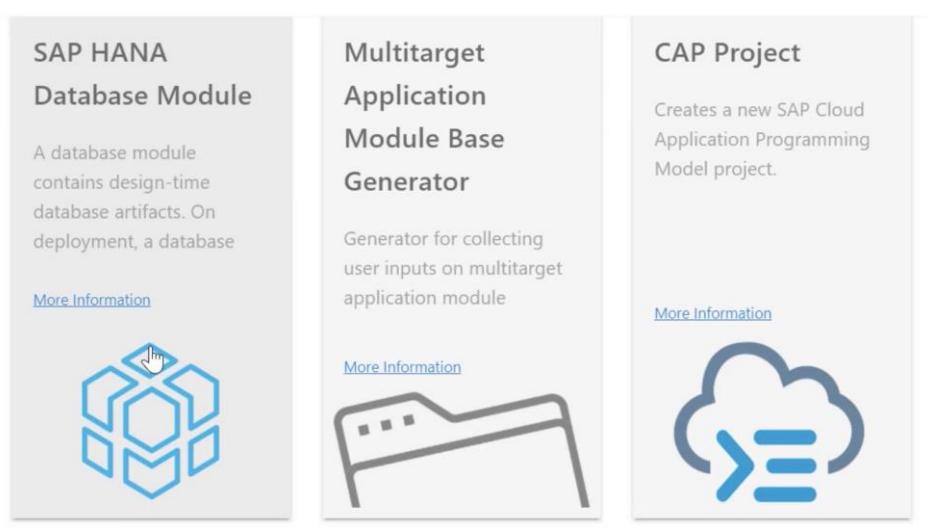
Install the generator

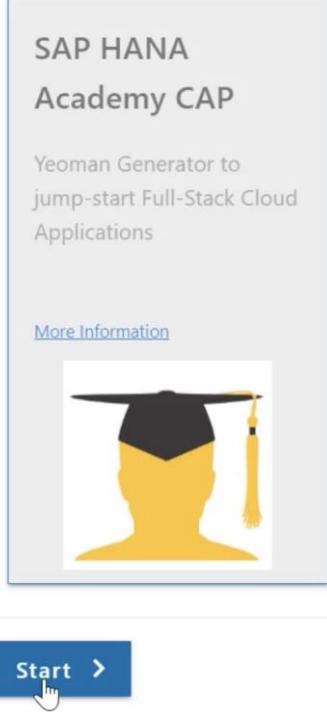
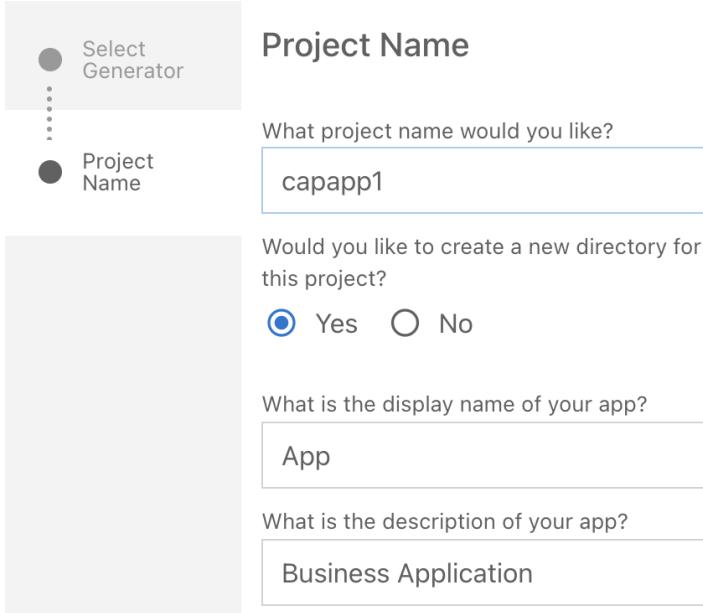
Explanation	Screenshot
Click on the Extensions icon, and then click on the OK button to accept the conditions.	
Then, search for an extension called " SAP Partner Engineering BAS Installer ". This generator will help in generating a CAP application.	

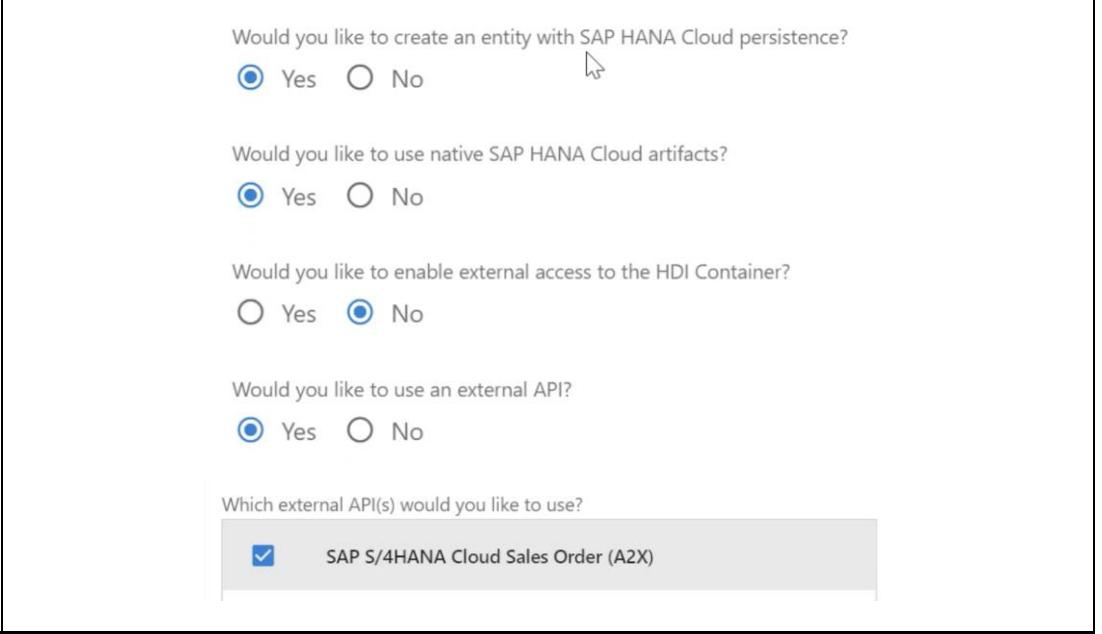
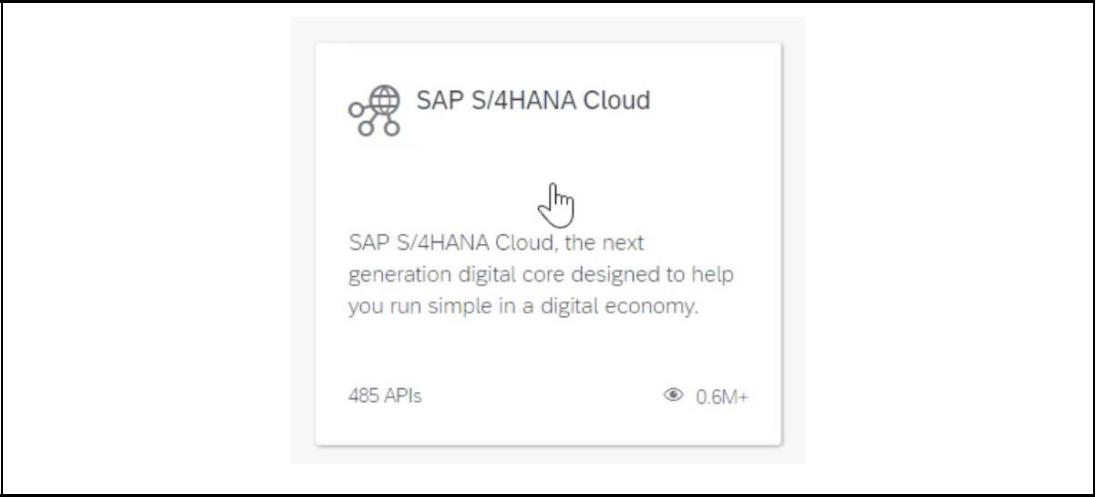
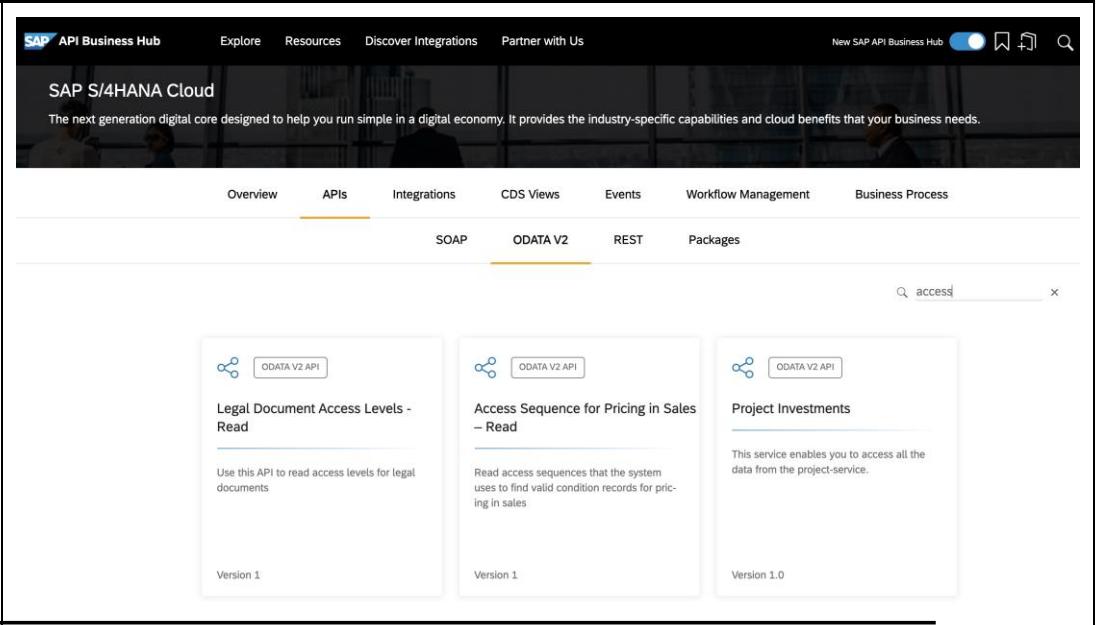
Explanation	Screenshot
<p>After it is installed, click on the menu View >> Find Command.</p>	
<p>Then search for “BAS: Install SAP HANA Academy yeoman generators” and click on it.</p>	

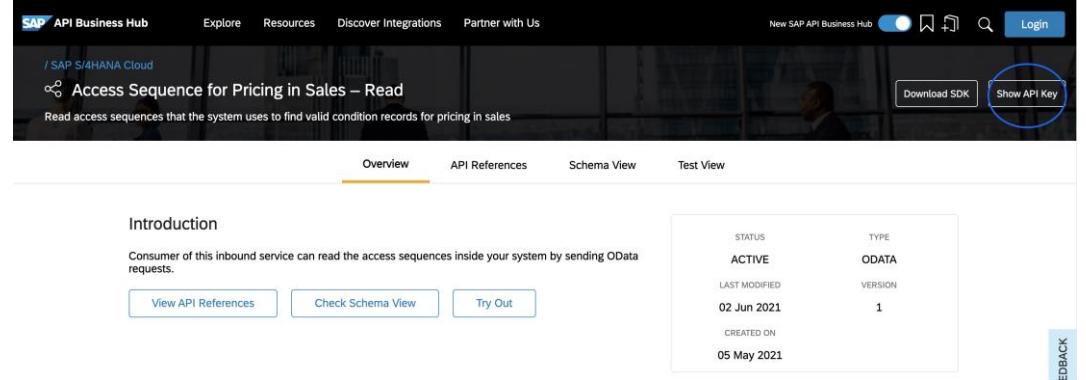
GENERATE THE APPLICATION

Generate the app with the generator

Explanation	Screenshot
<p>Click on the menu View >> Find Command.</p> <p>Search for “Open Template Wizard”.</p>	
<p>Different generators are displayed.</p>	

Explanation	Screenshot
<p>Select the “SAP HANA Academy CAP” wizard and click on “Start”.</p>	
<p>Then enter the following values for the fields that are displayed next.</p> <p>Important Note: To avoid further problems as BTP services are auto generated out of your project name please enter a <u>project name</u>:</p> <ul style="list-style-type: none"> - containing ONLY small letters and - globally unique. <p>Select “Yes” to create a new directory.</p> <p>Enter a display name.</p> <p>Enter the description of your app</p>	

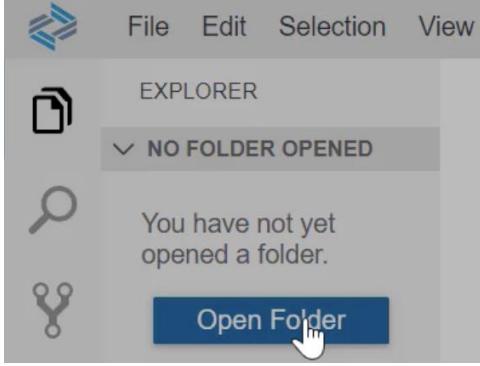
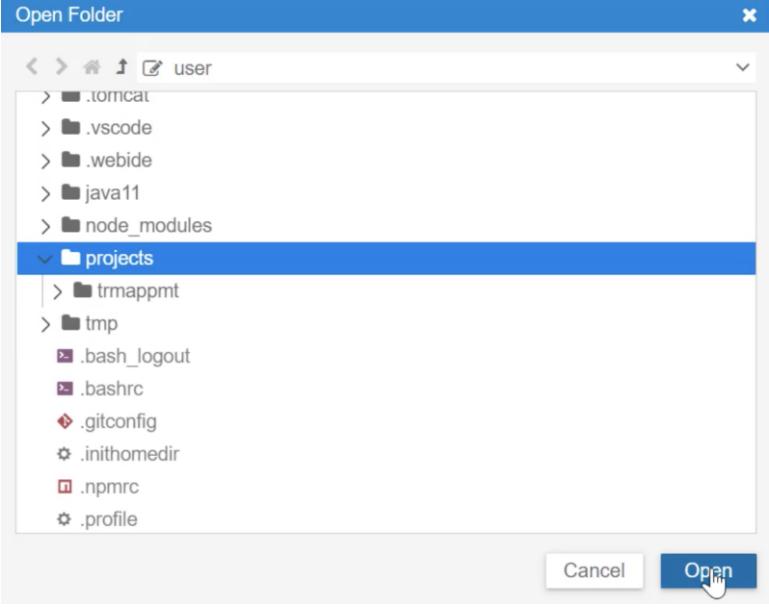
Explanation	Screenshot
<p>Leave the other selections with the default values.</p> <p>Select Yes to use an external API, then “SAP S/4 HANA Cloud Sales Order (A2X)” as we will retrieve Sales Order details from S/4 HANA Cloud.</p>	
<p>We will use the SAP API Business Hub sandbox S/4HANA Cloud APIs for our testing. Therefore, we require the APIKey.</p> <p>In order to get the APIHUB API Key, go to https://api.sap.com and click on “SAP S/4HANA Cloud”.</p>	
<p>Then, click on “Access Sequence for Pricing in Sales – Read” or any other ODATA API.</p>	

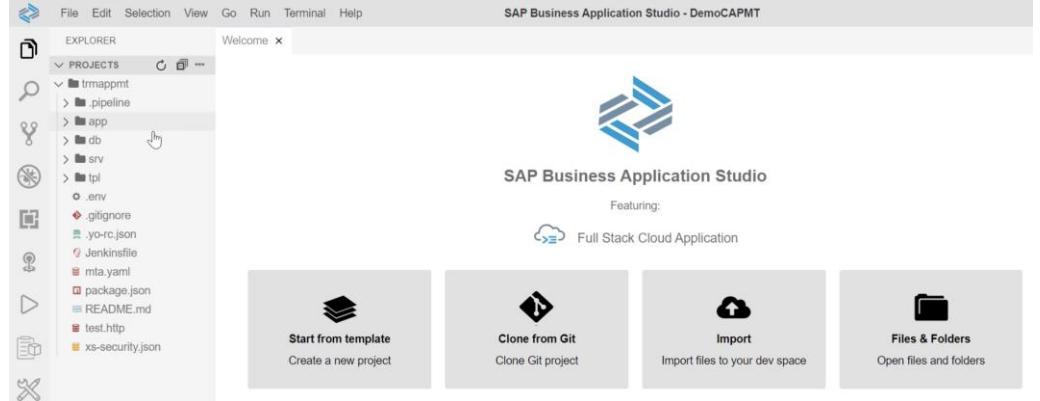
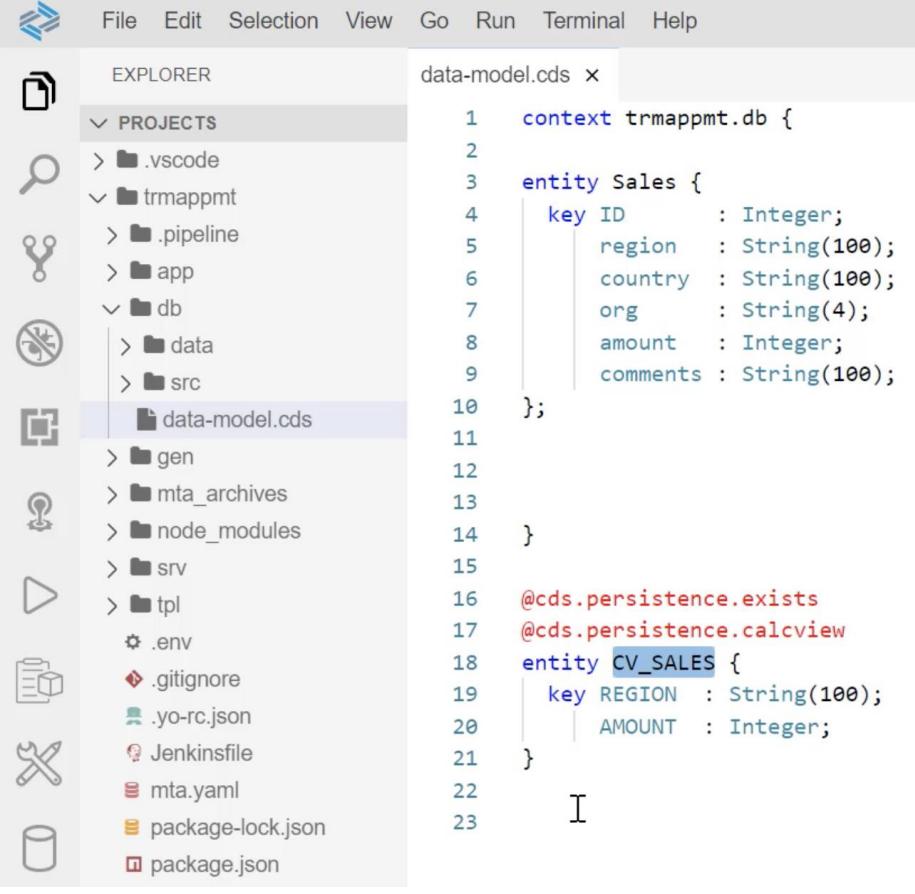
Explanation	Screenshot
<p>Then, click on “Show API Key”.</p>	
<p>You may need to login.</p>	<p>Login Required</p> <p>Log on to access the API key</p> <p>Login On Cancel</p>
<p>Next, the API Key will be displayed. Copy it by clicking on “Copy Key and Close”.</p>	<p>API Key X</p> <p>API Key: <input type="text" value="6R"/></p> <p>Copy Key and Close</p>
<p>Then, paste the API Key in the SAP HANA Academy wizard generating the application.</p>	<p>What is your API Key for the SAP API Business Hub sandbox?</p> <p><input type="text" value="....."/> <input style="width: 100px; margin-left: 10px;" type="text"/></p>

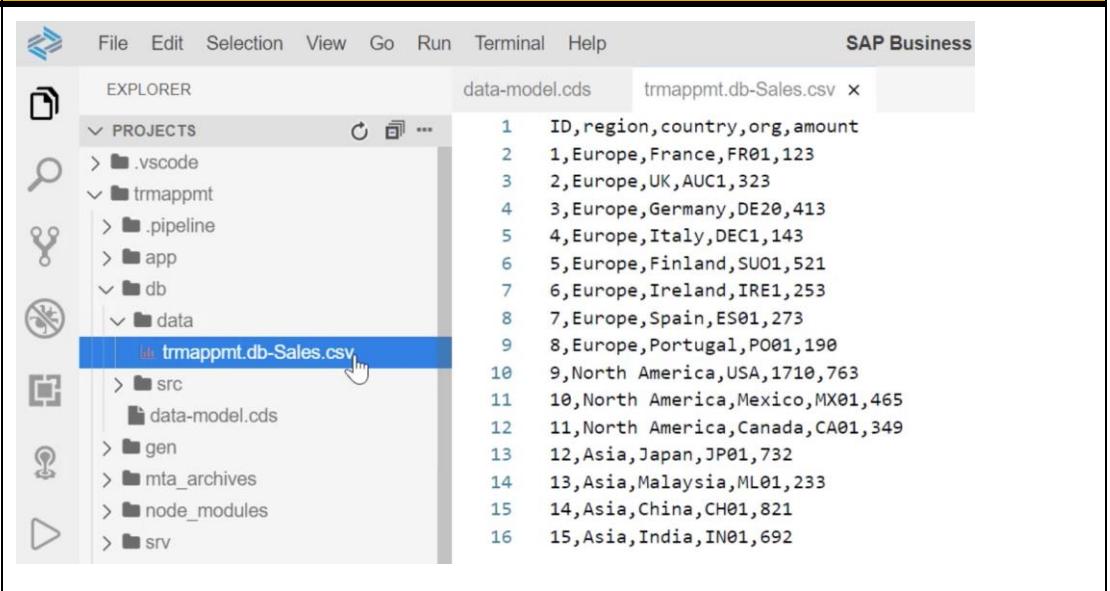
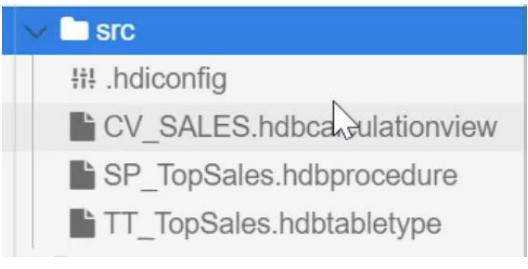
Explanation	Screenshot
<p>Leave the other selections with the default values.</p>	<p>Would you like authentication? <input checked="" type="radio"/> Yes <input type="radio"/> No</p> <p>Would you like authorization? <input checked="" type="radio"/> Yes <input type="radio"/> No</p> <p>Would you like to use role attributes? <input type="radio"/> Yes <input checked="" type="radio"/> No</p> <p>Would you like to enable OData v2 support? <input type="radio"/> Yes <input checked="" type="radio"/> No</p> <p>Would you like a UI? <input checked="" type="radio"/> Yes <input type="radio"/> No</p>
	<p>Would you like to use the HTML5 Application Repository? <input type="radio"/> Yes <input checked="" type="radio"/> No</p> <p>Will you be using a wildcard custom domain (eg: apps.mydomain.com)? If so please enter the custom domain name here. Leave blank to use the platform default. <input type="text"/></p>
<p>Select “Yes” to create a SaaS multitenant app.</p> <p>Select “Yes” to include creation/deletion of tenant routes.</p> <p>Select “Yes” to enable CI/CD and Application Logging.</p> <p>Leave the other selections with the default values.</p>	<p>Would you like to create a SaaS multitenant app? <input checked="" type="radio"/> Yes <input type="radio"/> No</p> <p>What is the category of your app? <input type="text" value="SaaS Multitenant Apps"/></p> <p>Would you like to include creation/deletion of tenant routes on subscribe/unsubscribe (via the CF API)? <input checked="" type="radio"/> Yes <input type="radio"/> No</p> <p>Would you like to enable messaging with SAP Event Mesh? <input type="radio"/> Yes <input checked="" type="radio"/> No</p> <p>Would you like to enable Continuous Integration and Delivery (CI/CD)? <input checked="" type="radio"/> Yes <input type="radio"/> No</p> <p>Would you like to enable Application Logging? <input checked="" type="radio"/> Yes <input type="radio"/> No</p>

Explanation	Screenshot
<p>Select “No” to build and deploy the project immediately.</p> <p>Click on the “Next” button to generate the app.</p>	<p>Would you like to build and deploy the project immediately?</p> <p><input type="radio"/> Yes <input checked="" type="radio"/> No</p> <p style="text-align: center;">Start Over Next ></p>

Review the generated application code

Explanation	Screenshot
<p>Next, click on “Open Folder” after the files are generated.</p>	
<p>You'll notice that the generated app has been added under the projects folder.</p> <p>Select the projects folder and click “Open”.</p>	

Explanation	Screenshot
<p>Clicking on “Open” will open your development environment with your workspace being the projects folder.</p> <p>You’ll notice that the application has been generated and it includes the following main modules.</p> <p>app – the main module</p> <p>db – that manages the connection to the HANA database</p> <p>srv – server-side module</p>	 <p>The screenshot shows the SAP Business Application Studio interface. The left sidebar displays the project structure under 'PROJECTS' for 'trmappmt'. The 'db' folder contains several files: .pipeline, app, db, srv, tpl, .env, .gitignore, .yo-rc.json, Jenkinsfile, mta.yaml, package.json, README.md, test.http, and xs-security.json. On the right, there's a logo for SAP Business Application Studio and a banner for 'Full Stack Cloud Application'. Below the banner are four buttons: 'Start from template' (Create a new project), 'Clone from Git' (Clone Git project), 'Import' (Import files to your dev space), and 'Files & Folders' (Open files and folders).</p>
<p>The db module has a data-model.cds file with two entities with the names Sales and CV_SALES.</p>	 <p>The screenshot shows the SAP Business Application Studio interface with the 'data-model.cds' file open in the editor. The code defines two entities: 'Sales' and 'CV_SALES'. The 'Sales' entity has attributes: ID (Integer), region (String(100)), country (String(100)), org (String(4)), amount (Integer), and comments (String(100)). The 'CV_SALES' entity has attributes: REGION (String(100)) and AMOUNT (Integer). The code is as follows:</p> <pre> 1 context trmappmt.db { 2 3 entity Sales { 4 key ID : Integer; 5 region : String(100); 6 country : String(100); 7 org : String(4); 8 amount : Integer; 9 comments : String(100); 10 }; 11 12 13 14 } 15 16 @cds.persistence.exists 17 @cds.persistence.calcview 18 entity CV_SALES { 19 key REGION : String(100); 20 AMOUNT : Integer; 21 } 22 23 </pre>

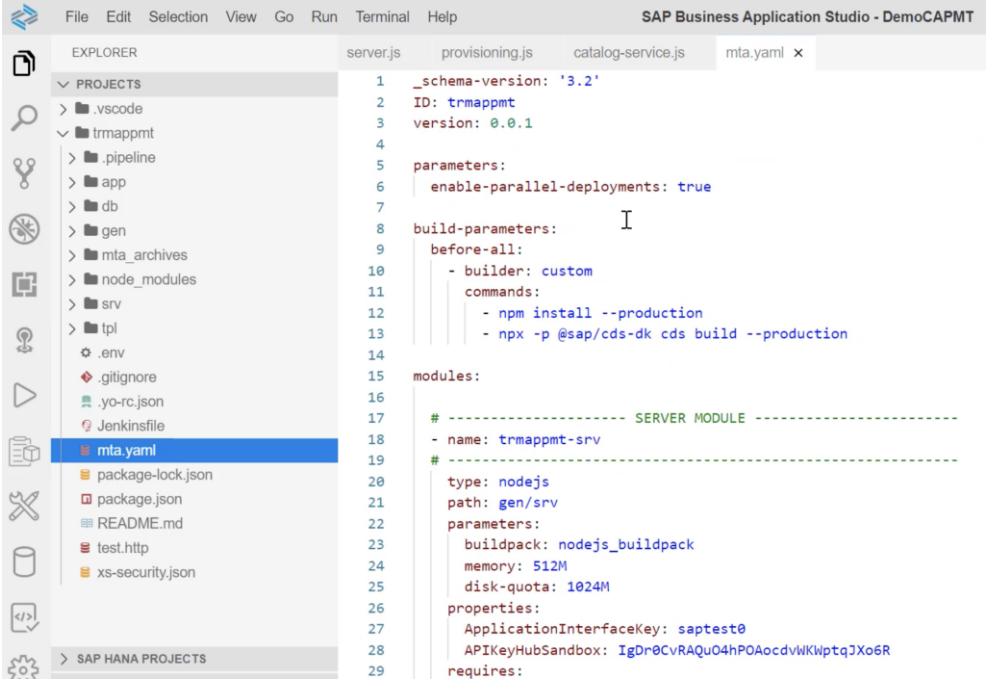
Explanation	Screenshot
Some default data is included in a csv file.	 <p>The screenshot shows the SAP Business Studio interface with the Explorer view open. The 'PROJECTS' section lists '.vscode', 'trmappmt' (which contains '.pipeline', 'app', 'db' (with 'data' as a child), 'src' (containing 'data-model.cds'), 'gen', 'mta_archives', 'node_modules', and 'srv'). The 'data' folder under 'db' is expanded, and the 'trmappmt.db-Sales.csv' file is selected, highlighted with a blue background. The right-hand panel displays the contents of the CSV file:</p> <pre> 1 ID,region,country,org,amount 2 1,Europe,France,FR01,123 3 2,Europe,UK,AUC1,323 4 3,Europe,Germany,DE20,413 5 4,Europe,Italy,DEC1,143 6 5,Europe,Finland,SU01,521 7 6,Europe,Ireland,IRE1,253 8 7,Europe,Spain,ES01,273 9 8,Europe,Portugal,PO01,190 10 9,North America,USA,1710,763 11 10,North America,Mexico,MX01,465 12 11,North America,Canada,CA01,349 13 12,Asia,Japan,JP01,732 14 13,Asia,Malaysia,ML01,233 15 14,Asia,China,CH01,821 16 15,Asia,India,IN01,692 </pre>
The db/src folder has a calculation view, a procedure and a table.	 <p>The screenshot shows the SAP Business Studio interface with the Explorer view open. The 'src' folder is expanded, showing files: '.hdiconfig', 'CV_SALES.hdbc', 'SP_TopSales.hdbprocedure', and 'TT_TopSales.hdbtabletype'.</p>
The srv module has the following files.	 <p>The screenshot shows the SAP Business Studio interface with the Explorer view open. The 'srv' folder is expanded, showing files: 'external', 'i18n', 'catalog-service.cds', 'catalog-service.js', 'provisioning.js', and 'server.js'.</p>

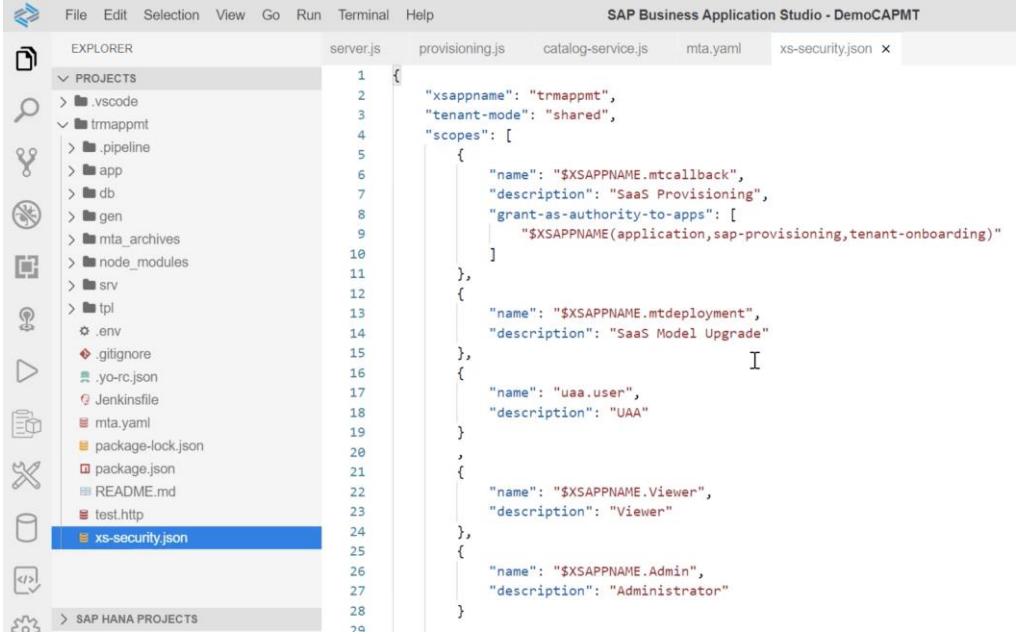
Explanation	Screenshot
<p>The server.js file is the main entry point of the app.</p> <p>It can be observed that the main module initializes the mtx module. During initialization, a connection is made to the provisioning service, that helps to expose the application as a service in BTP.</p>	<pre>server.js x 1 const cds = ...require('@sap/cds'); 2 3 4 cds.on('bootstrap', app => { 5 6 7 cds.mtx.in(app).then(async () => { 8 const provisioning = await cds.connect.to('ProvisioningService'); 9 provisioning.impl(require('./provisioning')); 10 }); 11 12 }); 13 14 module.exports = cds.server; 15</pre>
<p>The implementation of the provisioning service is within the provisioning.js file, that handles the requirements for provisioning of the application. It can be observed that it is using the SaaS registry and the destination service to connect to S/4HANA Cloud.</p>	<pre>server.js provisioning.js x 1 const debug = require('debug')('srv:provisioning'); 2 const cfenv = require('cfenv'); 3 const appEnv = cfenv.getAppEnv(); 4 const xsenv = require('@sap/xsenv'); 5 xsenv.loadEnv(); 6 const services = xsenv.getServices({ 7 registry: { tag: 'SaaS' } 8 , 9 dest: { tag: 'destination' } 10 }); 11 12 const axios = require('axios'); 13 const qs = require('qs'); 14 15 async function getCFInfo(appname) { 16 try { 17 // get authentication url 18 let options = { 19 method: 'GET', 20 url: appEnv.app.cf_api + '/info' 21 }; 22 let res = await axios(options); 23 try { 24 // get access token 25 let options1 = { 26 method: 'POST', 27 url: res.data.authorization_endpoint + '/oauth/token?grant_type=password',</pre>

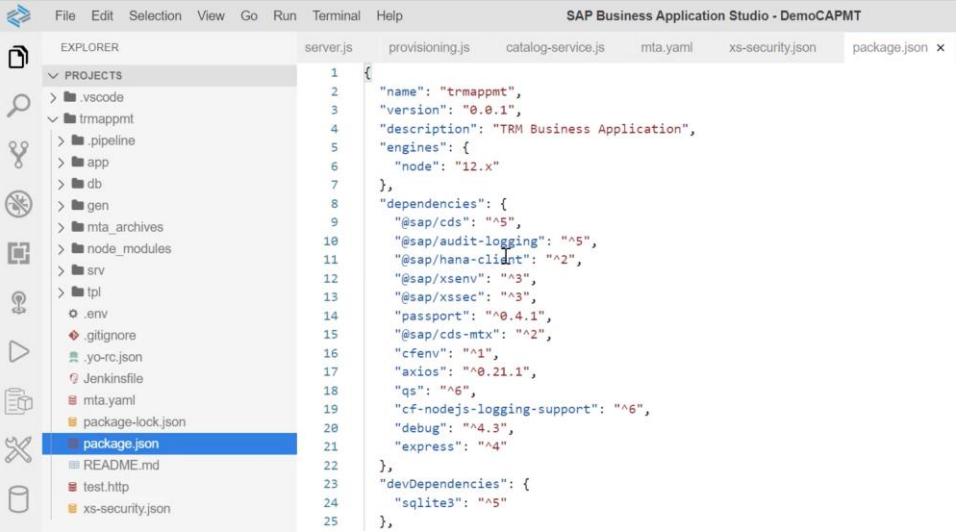
Explanation	Screenshot
<p>It can be observed within the same file that the app has the following three entry points related to the SaaS Provisioning Service:</p> <p>UPDATE – Invoked during the SaaS subscription and returns the specific tenant URL corresponding to the subscription, which is created based on the subaccount. A route is created based on the subaccount and the subscriber's tenant ID.</p>	<pre data-bbox="568 202 1548 1242"> module.exports = (service) => { service.on('UPDATE', 'tenant', async (req, next) => { let tenantHost = req.data.subscribedSubdomain + '-' + appEnv.app.space_name.toLowerCase().replace(/\s/g, ''); let tenantURL = 'https:///' + tenantHost + '/(.*)/gm.exec(appEnv.app.application_uris[0])'; console.log('Subscribe: ', req.data.subscribedSubdomain, req.data.subscribedTenantId, tenantHost); await next(); createRoute(tenantHost, services.registry.appName).then(function (res2) { console.log('Subscribe: - Create Route: ', req.data.subscribedTenantId, tenantHost, tenantURL); return tenantURL; }, function (err) { debug(err.stack); return ''; }); return tenantURL; I }); service.on('DELETE', 'tenant', async (req, next) => { let tenantHost = req.data.subscribedSubdomain + '-' + appEnv.app.space_name.toLowerCase().replace(/\s/g, ''); console.log('Unsubscribe: ', req.data.subscribedSubdomain, req.data.subscribedTenantId, tenantHost); await next(); deleteRoute(tenantHost, services.registry.appName).then(async function (res2) { console.log('Unsubscribe: - Delete Route: ', req.data.subscribedTenantId); return req.data.subscribedTenantId; }, function (err) { debug(err.stack); return ''; }); return req.data.subscribedTenantId; }); service.on('dependencies', async (req) => { let dependencies = [{ I 'xsappname': services.dest.xsappname }]; console.log('Dependencies: ', dependencies); return dependencies; }); } </pre>
<p>DELETE – Invoked during subscriptions and deletes the route that was initially created during subscription.</p> <p>Dependencies – Used to return the dependencies such as S/4HANA</p>	
<p>The implementation of the above methods is also included with the same file.</p>	<pre data-bbox="568 1284 1548 1643"> 14 15 > async function getCFInfo(appname) { ... 80 }; I 81 82 > async function createRoute(tenantHost, appname) { ... 144 }; 145 146 > async function deleteRoute(tenantHost, appname) { ... 190 }; </pre>

Explanation	Screenshot
<p>Within the implementation of the createRoute method, an HTTP POST request is made to create the route given the information of the tenant.</p>	<pre data-bbox="600 207 1455 1010"> async function createRoute(tenantHost, appname) { getCFInfo(appname).then(async function (CFInfo) { try { // create route let options = { method: 'POST', url: appEnv.app.cf_api + '/v3/routes', data: { 'host': tenantHost, 'relationships': { 'space': { 'data': { 'guid': appEnv.app.space_id } }, 'domain': { 'data': { 'guid': CFInfo.domain_id } } }, headers: { 'Authorization': 'Bearer ' + CFInfo.access_token, 'Content-Type': 'application/json' } } }; } }); } </pre>
<p>It can be observed that the route is mapped to the application.</p>	<pre data-bbox="556 1052 1548 1738"> let res = await axios(options); try { // map route to app let options2 = { method: 'POST', url: appEnv.app.cf_api + '/v3/routes/' + res.data.guid + '/destinations', data: { 'destinations': [{ 'app': { 'guid': CFInfo[app_id] } }] }, headers: { 'Authorization': 'Bearer ' + CFInfo.access_token, 'Content-Type': 'application/json' } }; let res2 = await axios(options2); console.log('Route created for ' + tenantHost); return res2.data; } catch (err) { debug(err.stack); } </pre>

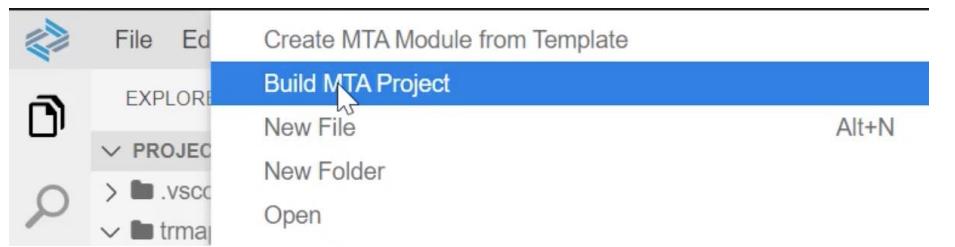
Explanation	Screenshot
<p>The getCFInfo method gets the Cloud Foundry information.</p>	<pre data-bbox="556 202 1536 910"> async function getCFInfo(appname) { try { // get authentication url let options = { method: 'GET', url: appEnv.app.cf_api + '/info' }; let res = await axios(options); try { // get access token let options1 = { method: 'POST', url: res.data.authorization_endpoint + '/oauth/token?grant_type=password', data: qs.stringify({ username: process.env.CFAPIUser, password: process.env.CFAPIPassword }), headers: { 'Authorization': 'Basic ' + Buffer.from('cf:').toString('base64'), 'Content-Type': 'application/x-www-form-urlencoded' } }; let res1 = await axios(options1); try { // get app guid let options2 = { method: 'GET', </pre>
<p>Within the catalog-service.js file, the cds.tx call is used to get specific information about a tenant of the application.</p>	<pre data-bbox="548 963 1536 1519"> this.on('boost', async req => { try { const ID = req.params[0]; const tx = cds.tx(req); await tx.update(Sales) .with({ amount: { '+=': 250 }, comments: 'Boosted!' }) .where({ ID: { '=': ID } }) ; debug('Boosted ID:', ID); return {}; } catch (err) { console.error(err); return {}; } }); </pre>

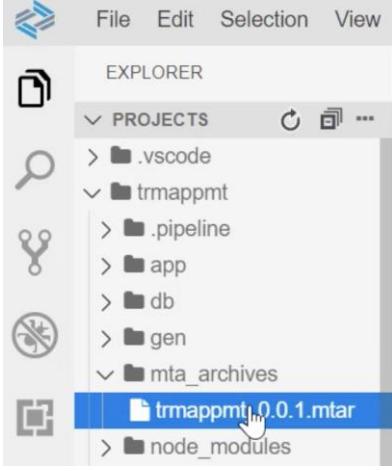
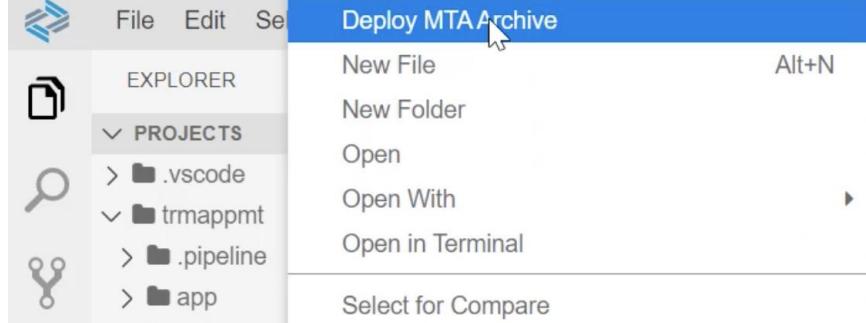
Explanation	Screenshot
<p>The mta.yaml file has the definitions of the various modules such as the Server module, the app router module, the UAA service, the Destination service (for accessing S/4HANA), the Registry service, the Service manager service (used to connect the HANA HDI container depending on the specific tenant) and the Application logging service.</p>	 <pre> 1 _schema-version: '3.2' 2 ID: trmappmt 3 version: 0.0.1 4 5 parameters: 6 enable-parallel-deployments: true 7 8 build-parameters: 9 before-all: 10 - builder: custom 11 commands: 12 - npm install --production 13 - npx -p @sap/cds-dk cds build --production 14 15 modules: 16 17 # ----- SERVER MODULE ----- 18 - name: trmappmt-srv 19 # 20 type: nodejs 21 path: gen/srv 22 parameters: 23 buildpack: nodejs_buildpack 24 memory: 512M 25 disk-quota: 1024M 26 properties: 27 ApplicationInterfaceKey: saptest0 28 APIKeyHubSandbox: IgDr0CvRAQu04hPOAocdvWKWptqJXo6R 29 requires: </pre>
<p>It can be observed under the definition of the destination service that we are using the sandbox API, for which the API Key was configured when generating the app.</p>	<pre> # ----- DESTINATION SERVICE ----- - name: trmappmt-dest # type: org.cloudfoundry.managed-service parameters: service: destination service-plan: lite config: init_data: subaccount: existing_destinations_policy: update destinations: - Name: trmappmt-API_SALES_ORDER_SRV Description: SAP S/4HANA Cloud Sales Order (A2X) URL: https://sandbox.api.sap.com Type: HTTP ProxyType: Internet Authentication: NoAuthentication HTML5.DynamicDestination: true </pre>

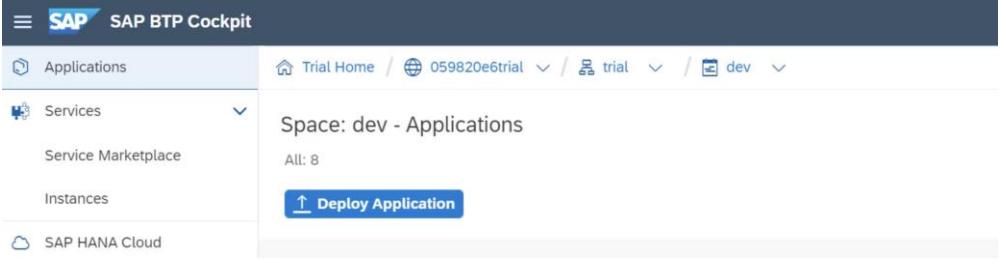
Explanation	Screenshot
<p>The xs-security.json file contains all the scopes of the application.</p>	 <p>The screenshot shows the SAP Business Application Studio interface. In the top right, the title bar reads "SAP Business Application Studio - DemoCAPMT". Below it, several tabs are visible: server.js, provisioning.js, catalog-service.js, mta.yaml, and xs-security.json (which is currently active). On the left, the Explorer view displays a project structure under "PROJECTS" for "trmappmt". The "xs-security.json" file is highlighted with a blue selection bar at the bottom of the list. The main workspace shows the contents of the xs-security.json file:</p> <pre> 1 { 2 "xsappname": "trmappmt", 3 "tenant-mode": "shared", 4 "scopes": [5 { 6 "name": "\$XSAPPNAME.mtcallback", 7 "description": "SaaS Provisioning", 8 "grant-as-authority-to-apps": [9 "\$XSAPPNAME(application,sap-provisioning,tenant-onboarding)" 10] 11 }, 12 { 13 "name": "\$XSAPPNAME.mtdeployment", 14 "description": "SaaS Model Upgrade" 15 }, 16 { 17 "name": "uaa.user", 18 "description": "UAA" 19 }, 20 { 21 "name": "\$XSAPPNAME.Viewer", 22 "description": "Viewer" 23 }, 24 { 25 "name": "\$XSAPPNAME.Admin", 26 "description": "Administrator" 27 } 28 } 29 </pre>
<p>The multitenancy scopes are mtcallback and mtdeployment.</p> <p>mtcallback authorizes the provisioning and onboarding services.</p>	 <p>The screenshot shows the xs-security.json file open in the code editor. The "mtcallback" scope is highlighted with a blue selection bar. The code snippet below illustrates its definition:</p> <pre> { "xsappname": "trmappmt", "tenant-mode": "shared", "scopes": [{ "name": "\$XSAPPNAME.mtcallback", "description": "SaaS Provisioning", "grant-as-authority-to-apps": ["\$XSAPPNAME(application,sap-provisioning,tenant-onboarding)"] }, { "name": "\$XSAPPNAME.mtdeployment", "description": "SaaS Model Upgrade" }] } </pre>

Explanation	Screenshot
<p>The package.json file contains all the dependencies of the app. The cds-mtx dependency is required for multitenancy.</p>	 <pre> 1 "name": "trmappmt", 2 "version": "0.0.1", 3 "description": "TRM Business Application", 4 "engines": { 5 "node": "12.x" 6 }, 7 "dependencies": { 8 "@sap/cds": "^5", 9 "@sap/audit-logging": "^5", 10 "@sap/hana-client": "2", 11 "@sap/xsenv": "3", 12 "@sap/xssec": "3", 13 "passport": "0.4.1", 14 "@sap/cds-mtx": "2", 15 "cfenv": "1", 16 "axios": "0.21.1", 17 "qs": "6", 18 "cf-nodejs-logging-support": "6", 19 "debug": "4.3", 20 "express": "4" 21 }, 22 "devDependencies": { 23 "sqlite3": "5" 24 } 25 </pre>
<p>In this file, it can also be observed that multitenancy has been set as true and the destination points to the sandbox API Hub. This destination can be overridden in a consumer subaccount.</p>	 <pre> "multitenancy": true, "API_SALES_ORDER_SRV": { "kind": "odata", "model": "srv/external/API_SALES_ORDER_SRV", "credentials": { "[production)": { "destination": "trmappmt-API_SALES_ORDER_SRV", "path": "s4hanacloud/sap/opu/odata/sap/API_SALES_ORDER_SRV" }, "[development)": { "url": "https://sandbox.api.sap.com/s4hanacloud/sap/opu/odata/sap/API_SALES_ORDER_SRV" } } } </pre>

Build and deploy the app in the Provider Subaccount.

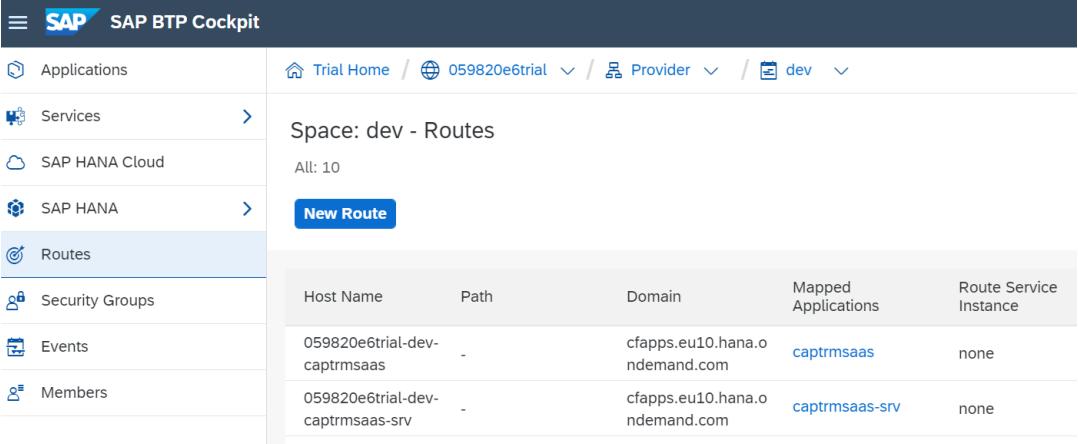
Explanation	Screenshot
<p>Right click on the mta.yaml file, and click on “Build MTA Project”.</p>	

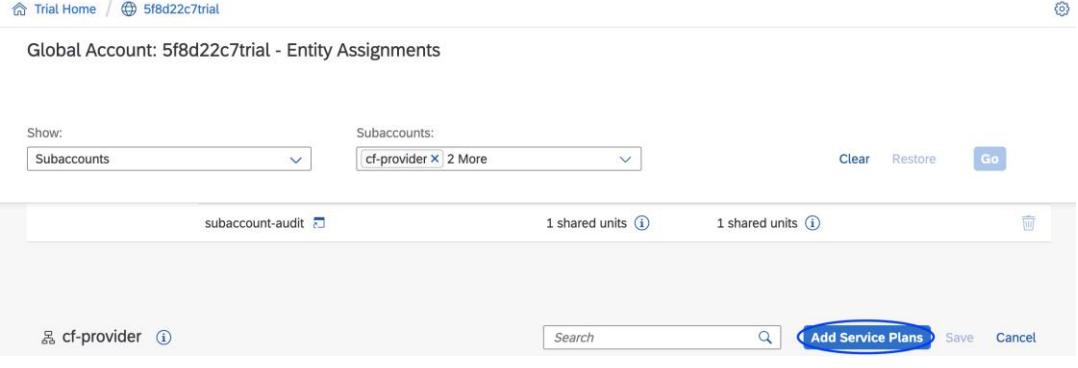
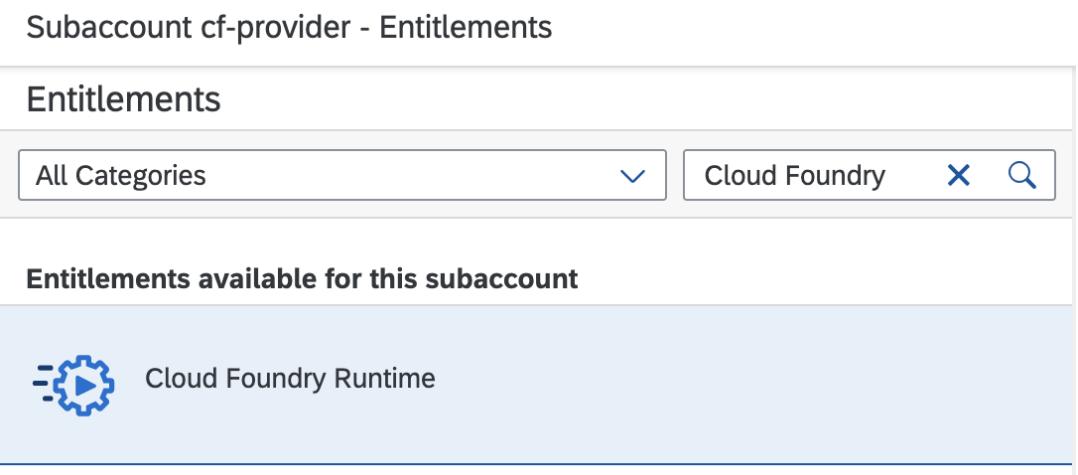
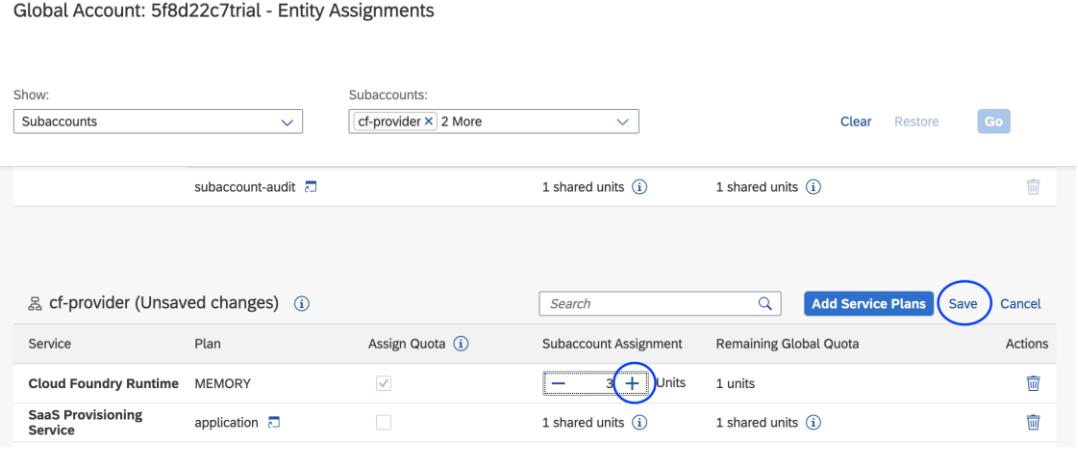
Explanation	Screenshot
<p>This will cause the app to be built.</p>	<pre>added 263 packages from 163 contributors in 6.758s 3 packages are looking for funding run 'npm fund' for details [2021-06-07 07:18:46] INFO the build results of the "trmappmt" module will be packaged and saved in the "/home/user/projects/trmappmt/.trmappmt_mta_build_tmp/trmappmt" folder [2021-06-07 07:18:49] INFO finished building the "trmappmt" module [2021-06-07 07:18:49] INFO running the "after-all" build... [2021-06-07 07:18:49] INFO generating the metadata... [2021-06-07 07:18:49] INFO generating the "/home/user/projects/trmappmt/.trmappmt_mta_build_tmp/META-INF/mtad.yaml" file ... [2021-06-07 07:18:49] INFO generating the MTA archive... [2021-06-07 07:18:53] INFO the MTA archive generated at: /home/user/projects/trmappmt/mta_archives/trmappmt_0.0.1.mtar [2021-06-07 07:18:53] INFO cleaning temporary files... Terminal will be reused by tasks.</pre>
<p>A file with mtar extension is generated and added to the mta_archives folder.</p>	
<p>Right click on the file and click on “Deploy MTA Archive”.</p>	
<p>To deploy the app, we need to select the Cloud Foundry environment that we would like to connect, and we're prompted to enter an email and password, after which we need to select the organization/account and space, where the app is to be deployed.</p>	

Explanation	Screenshot
The app is deployed next.	<pre>Uploading 1 files... [/home/user/projects/trmappmt/mta_archives/trmappmt_0.0.1.mtar Binding service instance "trmappmt-uaa" to application "trmappmt"... Uploading application "trmappmt"... Started async upload of application "trmappmt" Scaling application "trmappmt-srv" to "1" instances... Staging application "trmappmt-srv"... Scaling application "trmappmt" to "1" instances... Staging application "trmappmt"... Application "trmappmt" staged Starting application "trmappmt"... Application "trmappmt" started and available at "059820e6trial-dev-trmappmt.cfapps.eu10.hana.ondemand.com" Application "trmappmt-srv" staged Starting application "trmappmt-srv"... Application "trmappmt-srv" started and available at "059820e6trial-dev-trmappmt-srv.cfapps.eu10.hana.ondemand.com" Skipping deletion of services, because the command line option "--delete-services" is not specified. Process finished. Use "cf dmol -i da6741d8-c760-11eb-8e0a-e0a90f50c" to download the logs of the process.</pre>
To access the app, go to the Cloud Foundry space of the Provider Subaccount.	 <p>The SAP BTP Cockpit interface shows the Applications section. It lists two applications: "trmappmt" and "trmappmt-srv". Both applications are marked as "Started". The "trmappmt" application has 1/1 instances, 1024 MB memory, and 256 MB swap. The "trmappmt-srv" application also has 1/1 instances, 1024 MB memory, and 512 MB swap. There are standard navigation buttons for "Deploy Application" and "Delete" (trash can).</p>
It can be observed that the main app and the server-side app are running there.	 <p>The Cloud Foundry dashboard displays the "trmappmt" and "trmappmt-srv" applications. Both are listed as "Started". The "trmappmt" application has 1/1 instances, 1024 MB memory, and 256 MB swap. The "trmappmt-srv" application has 1/1 instances, 1024 MB memory, and 512 MB swap. Each application has a "Details" button, a "Logs" button, and a "Delete" button.</p>
<p>To be able to subscribe, the Cloud Foundry (CF) credentials need to be setup as environment variables.</p> <p>Run the login command first.</p> <p>(More info here)</p>	<p>cf login -a <API-URL> -u <USERNAME> -p <PASSWORD> -o <ORG> -s <SPACE></p> <p>Where:</p> <ul style="list-style-type: none"> <API-URL> is your API endpoint. <USERNAME> is your username. <PASSWORD> is your password. <ORG> is the org where you want to deploy your apps. <SPACE> is the space in the org where you want to deploy your apps. <pre>cf set-env <app-srv> CFAPIUser '<email>' cf set-env <app-srv> CFAPIPassword '<password>' cf restage <app-srv></pre>

Explanation	Screenshot
Setup your Cloud Foundry (CF) credentials as environment variables for your application by running the CF CLI commands.	

Troubleshooting the Deployment

Explanation	Screenshot															
If you get a “Process failed” error with the message “You have exceeded the total routes of your organization” while deploying your application, please check the following.	<pre>Proceeding with automatic retry... (1 of 3 attempts left) Updating application "trmappmt"... Updating application "trmappmt-srv"... Error creating or updating application "trmappmt-srv": Controller operation failed: 400 Bad Request: CF-OrgQuotaTotalRoutesExceeded(310006): You have exceeded the total routes for your organization's quota. Error creating or updating application "trmappmt": Controller operation failed: 400 Bad Request: CF-OrgQuotaTotalRoutesExceeded(310006): You have exceeded the total routes for your organization's quota. Process failed. Use "cf deploy -i 9711ab99-df74-11eb-9e1c-eeee0a959482 -a abort" to abort the process. Use "cf deploy -i 9711ab99-df74-11eb-9e1c-eeee0a959482 -a retry" to retry the process. Use "cf dmol -i 9711ab99-df74-11eb-9e1c-eeee0a959482" to download the logs of the process.</pre>															
You don't have any Route assignment left. A trial account allows a maximum of 10 Routes, if you have already 10 Routes assigned remove some of them. Go to your Provider subaccount >> Routes and check the number of Routes used doesn't exceed the number of 10 if using a trial account.	 <table border="1"> <thead> <tr> <th>Host Name</th> <th>Path</th> <th>Domain</th> <th>Mapped Applications</th> <th>Route Service Instance</th> </tr> </thead> <tbody> <tr> <td>059820e6trial-dev-captrmsaas</td> <td>-</td> <td>cfapps.eu10.hana.ondemand.com</td> <td>captrmsaas</td> <td>none</td> </tr> <tr> <td>059820e6trial-dev-captrmsaas-srv</td> <td>-</td> <td>cfapps.eu10.hana.ondemand.com</td> <td>captrmsaas-srv</td> <td>none</td> </tr> </tbody> </table>	Host Name	Path	Domain	Mapped Applications	Route Service Instance	059820e6trial-dev-captrmsaas	-	cfapps.eu10.hana.ondemand.com	captrmsaas	none	059820e6trial-dev-captrmsaas-srv	-	cfapps.eu10.hana.ondemand.com	captrmsaas-srv	none
Host Name	Path	Domain	Mapped Applications	Route Service Instance												
059820e6trial-dev-captrmsaas	-	cfapps.eu10.hana.ondemand.com	captrmsaas	none												
059820e6trial-dev-captrmsaas-srv	-	cfapps.eu10.hana.ondemand.com	captrmsaas-srv	none												

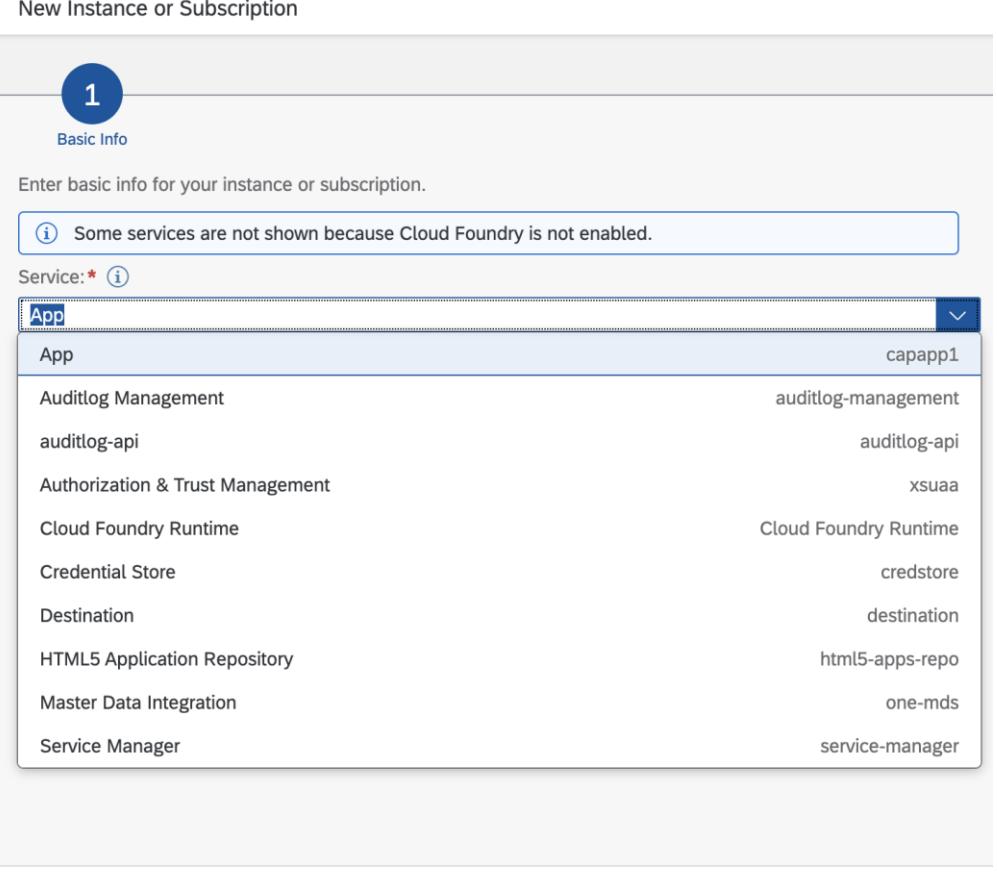
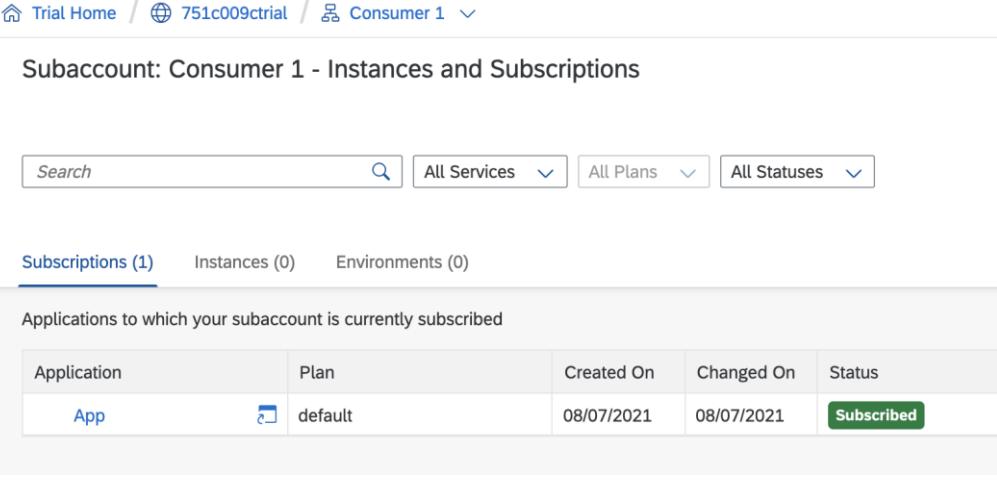
Explanation	Screenshot
<p>You don't have enough Cloud Foundry Runtime entitlement.</p> <p>Go to your Global account >> Entity Assignments >> Add Service Plans</p>	 <p>The screenshot shows the 'Entity Assignments' page for a global account. It displays two subaccounts: 'subaccount-audit' and 'cf-provider'. Both have 1 shared unit assigned. A blue oval highlights the 'Add Service Plans' button at the bottom right of the interface.</p>
<p>Search for “Cloud Foundry” and select the “Cloud Foundry Runtime” entitlement.</p>	 <p>The screenshot shows the 'Entitlements' page for the 'cf-provider' subaccount. It lists the 'Cloud Foundry Runtime' entitlement under 'Entitlements available for this subaccount'. A blue oval highlights the 'Cloud Foundry Runtime' row.</p>
<p>Click on the “+” sign on to “Cloud Foundry Runtime” row and increase the number of units to 3 or 4 and click on “Save”.</p>	 <p>The screenshot shows the 'Entity Assignments' page again. The 'Cloud Foundry Runtime' entitlement is assigned to 'subaccount-audit' with 3 units. A blue oval highlights the '+' sign next to the unit count '3' in the 'Assign Quota' column.</p>

PREPARING THE CONSUMER ENVIRONMENT

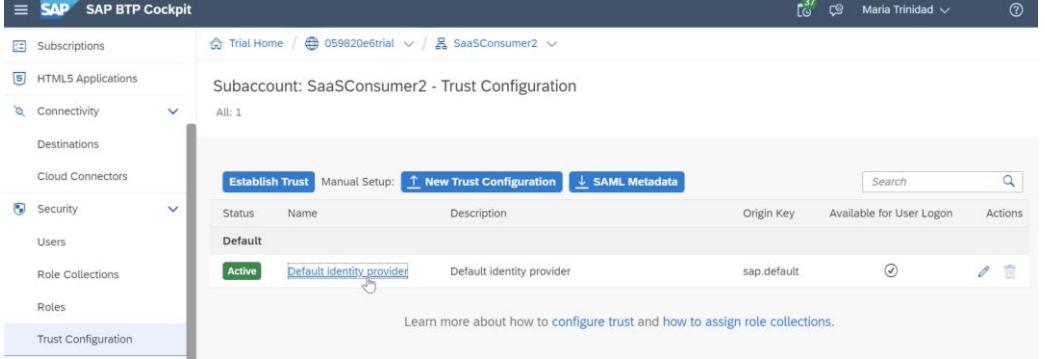
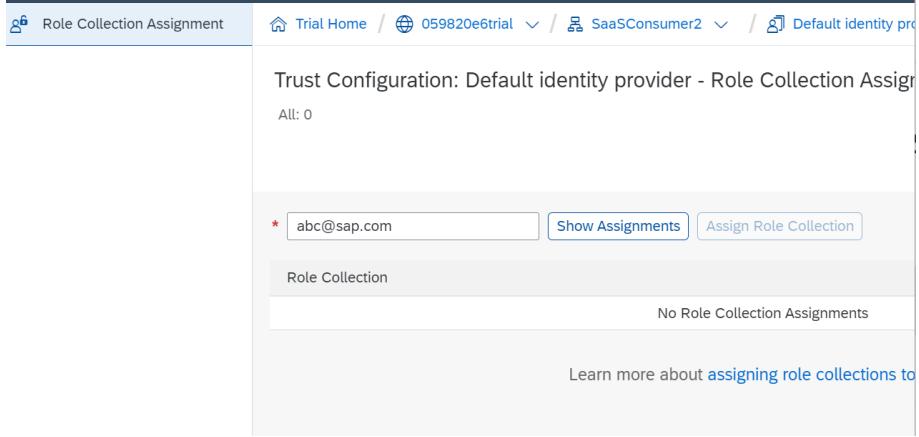
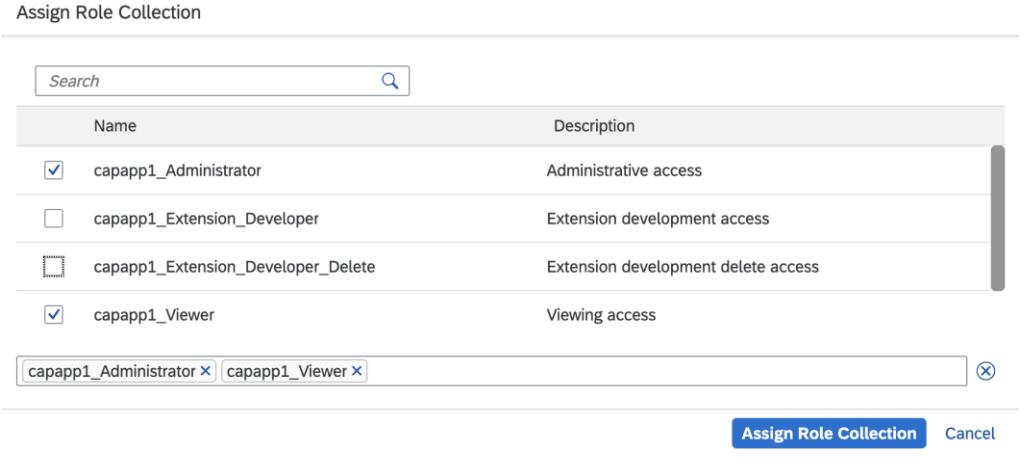
Create at least one Consumer Subaccount

Explanation	Screenshot
<p>Create a new Subaccount for each one of your consumers.</p> <p>Select the same Provider and Region as for your Provider subaccount.</p> <p>Set the subdomain of each consumer subaccount by appending the consumer number to the name of the global account to have meaningful app urls. For example, if the name of the global account is "012trial", then "c1", "c2", etc. are appended for each consumer subaccount.</p> <p>Example:</p> <ul style="list-style-type: none">• 012trialc1• 012trialc2	<p>The screenshot shows the AWS CloudFormation console with four subaccounts listed:</p> <ul style="list-style-type: none">Graph: Provider: Amazon Web Services (AWS), Region: Europe (Frankfurt), Description: -none-, Environment: Multi-EnvironmentProvider: Provider: Amazon Web Services (AWS), Region: Europe (Frankfurt), Description: -none-, Environment: Multi-EnvironmentSaaSConsumer1: Provider: Amazon Web Services (AWS), Region: Europe (Frankfurt), Description: SaaS Consumer 1, Environment: Multi-EnvironmentSaaSConsumer2: Provider: Amazon Web Services (AWS), Region: Europe (Frankfurt), Description: SaaS Consumer 2, Environment: Multi-Environment <p>The "Provider" and "SaaSConsumer1" subaccounts are highlighted with blue circles.</p>

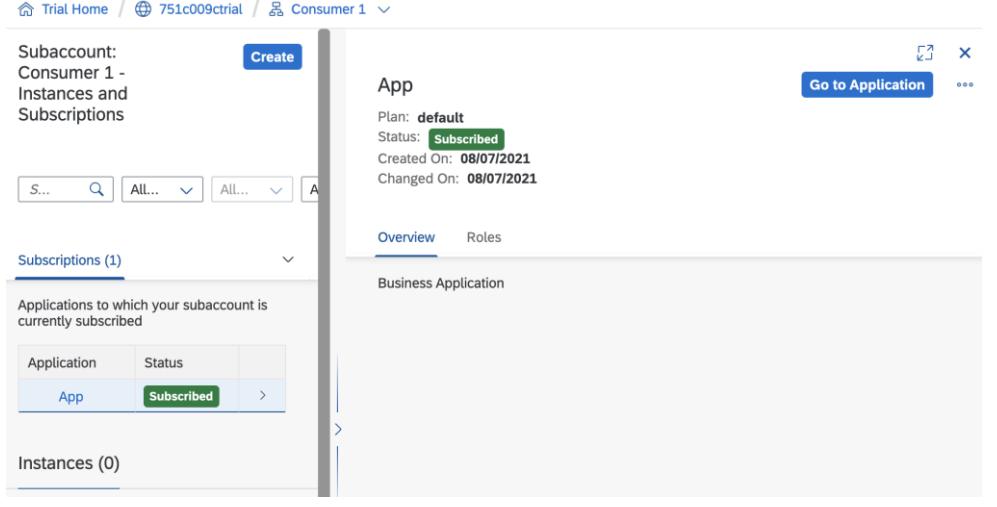
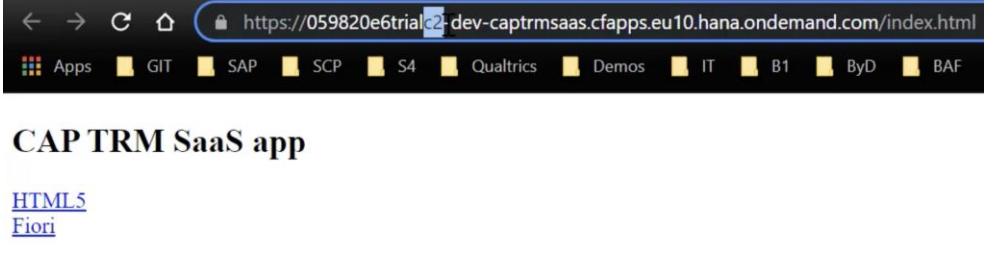
Subscribe to the Provider from the Tenant / Consumer

Explanation	Screenshot
<p>Within the consumer subaccount, go to Services >> Instances and Subscriptions and click on the “Create” button.</p> <p>Select the “<App>” Service (corresponding to your app name) and “default” plan and click on the “Create” button.</p>	
<p>A “Processing” status will be displayed while the subscription is in process.</p> <p>At the end you should get a “Subscribed” green status.</p> <p>Note: If you get a “Subscription Failed” status please check the General Troubleshooting section at the end of this document.</p>	

Assign Roles to the user

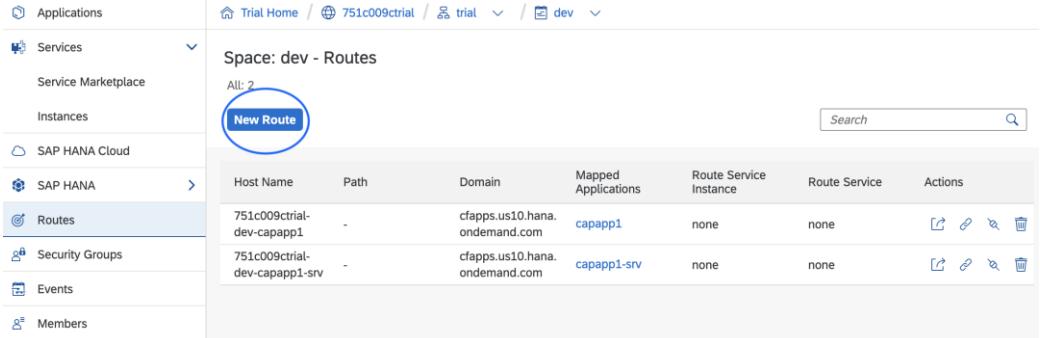
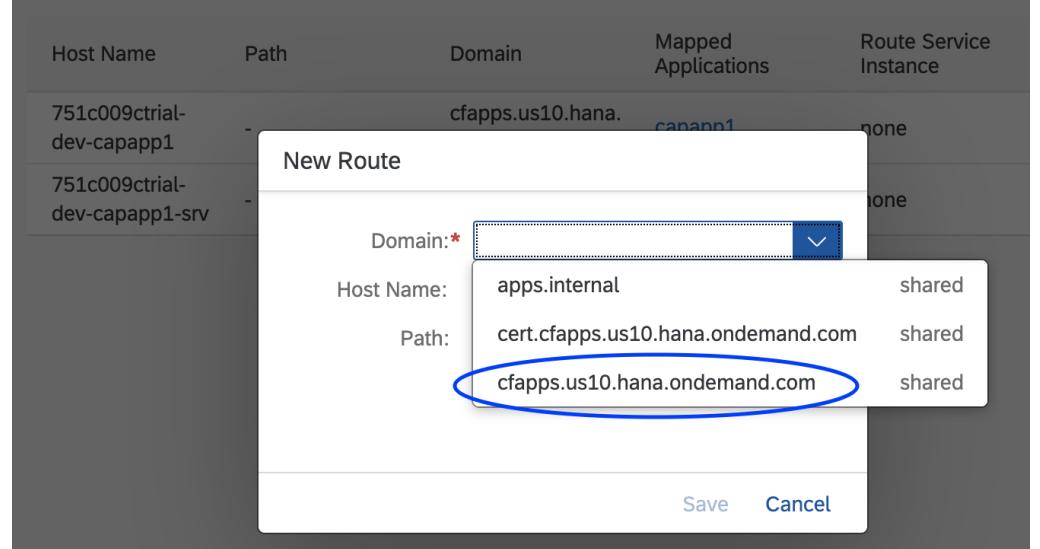
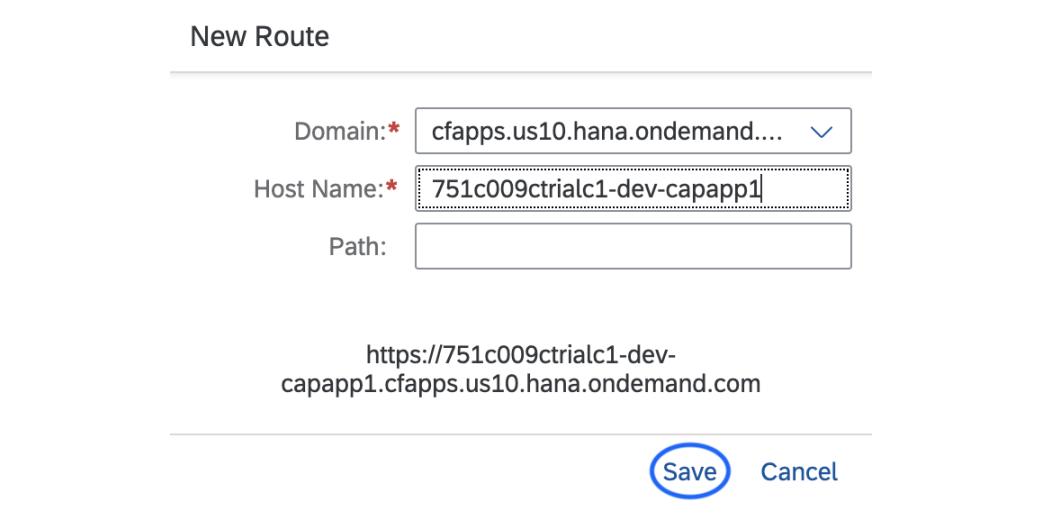
Explanation	Screenshot										
<p>Before launching the application, we will need to assign some roles to the user.</p> <p>Go to Security >> Trust Configuration >> Default Identity Provider.</p>	 <p>The screenshot shows the SAP BTP Cockpit interface. The left sidebar has sections like Subscriptions, HTML5 Applications, Connectivity, Destinations, Cloud Connectors, Security, Users, Role Collections, and Roles. Under Security, 'Trust Configuration' is selected. The main area shows 'Subaccount: SaaSConsumer2 - Trust Configuration'. It lists one entry: 'Default' (Status: Active, Name: Default identity provider, Description: Default identity provider, Origin Key: sap.default). There are buttons for 'Establish Trust', 'New Trust Configuration', and 'SAML Metadata'. A note at the bottom says 'Learn more about how to configure trust and how to assign role collections.'</p>										
<p>Enter your email and click “Show Assignments”.</p>	 <p>The screenshot shows the 'Role Collection Assignment' screen. The URL is Trial Home / 059820e6trial / SaaSConsumer2 / Default identity provider. It displays 'Trust Configuration: Default identity provider - Role Collection Assignment'. A search bar shows 'abc@sap.com'. Below it is a 'Role Collection' section with a note 'No Role Collection Assignments'. A link 'Learn more about assigning role collections to users' is present.</p>										
<p>Click “Assign Role Collection” and assign your application role collections to the user.</p> <ul style="list-style-type: none"> • <appname_Viewer> • <appname_Administrator> 	 <p>The screenshot shows the 'Assign Role Collection' dialog. It has a search bar and a table with columns 'Name' and 'Description'. Four items are listed: <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>capapp1_Administrator</td> <td>Administrative access</td> </tr> <tr> <td>capapp1_Extension_Developer</td> <td>Extension development access</td> </tr> <tr> <td>capapp1_Extension_Developer_Delete</td> <td>Extension development delete access</td> </tr> <tr> <td>capapp1_Visitor</td> <td>Viewing access</td> </tr> </tbody> </table> At the bottom, two checkboxes are selected: 'capapp1_Administrator' and 'capapp1_Visitor'. There are 'Assign Role Collection' and 'Cancel' buttons. </p>	Name	Description	capapp1_Administrator	Administrative access	capapp1_Extension_Developer	Extension development access	capapp1_Extension_Developer_Delete	Extension development delete access	capapp1_Visitor	Viewing access
Name	Description										
capapp1_Administrator	Administrative access										
capapp1_Extension_Developer	Extension development access										
capapp1_Extension_Developer_Delete	Extension development delete access										
capapp1_Visitor	Viewing access										

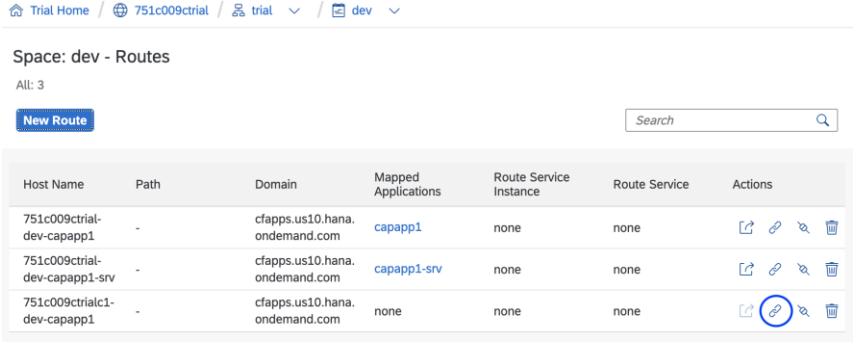
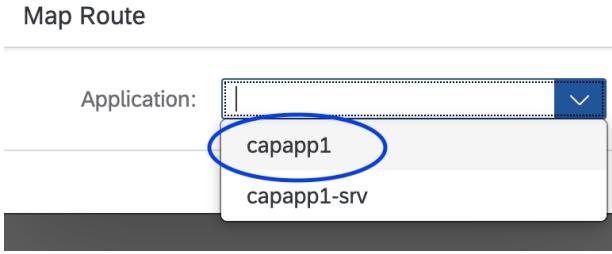
Access the app

Explanation	Screenshot
<p>After the subscription is successfully processed, and the roles are assigned to the user, clicking on the “Go to Application” button will launch the application.</p>	
<p>Please note that the URL of the consumer app is different from the other consumer app.</p>	

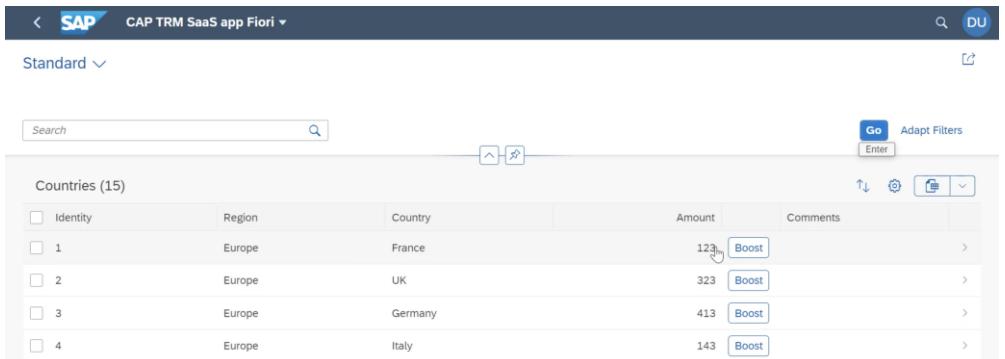
Troubleshooting the “Requested route ... does not exist” error

Explanation	Screenshot
<p>If you see the “Requested route ... does not exist” error, it means the specific route for your subaccount subscription wasn't correctly created. For creating the routes the CF environment variables need to be correctly set as indicated in the section Build and deploy the app in the Provider Subaccount.</p>	

Explanation	Screenshot
You can also manually create the route for your subaccount.	
<p>Go to the Provider subaccount >> Cloud Foundry >> Spaces >> “Space Name” >> Routes and press New Route</p>	 <p>The screenshot shows the SAP HANA Cloud provider subaccount interface. On the left, there's a sidebar with 'Applications', 'Services' (selected), 'Service Marketplace', 'Instances', 'SAP HANA Cloud', 'SAP HANA' (selected), 'Routes' (highlighted with a blue circle), 'Security Groups', 'Events', and 'Members'. The main area is titled 'Space: dev - Routes' with 'All: 2'. It lists two routes: '751c009ctrial-dev-capapp1' and '751c009ctrial-dev-capapp1-srv'. Each route has columns for Host Name, Path, Domain, Mapped Applications, Route Service Instance, Route Service, and Actions. A search bar is at the top right.</p>
Select the domain starting with “cfapps”.	 <p>The screenshot shows the 'New Route' dialog box. It has fields for 'Domain:' (set to 'cfapps.us10.hana.ondemand.com'), 'Host Name:' (set to '751c009ctrial-dev-capapp1'), and 'Path:' (with options 'cert.cfapps.us10.hana.ondemand.com' and 'cfapps.us10.hana.ondemand.com'). The 'Path:' field is highlighted with a blue oval.</p>
<p>Copy and paste the prefix of your consumer subaccount domain from the URL of the app and click “Save”.</p>	 <p>The screenshot shows the 'New Route' dialog box again. The 'Domain:' field is set to 'cfapps.us10.hana.ondemand....'. The 'Host Name:' field contains '751c009ctrialc1-dev-capapp1'. Below the dialog, the URL 'https://751c009ctrialc1-dev-capapp1.cfapps.us10.hana.ondemand.com' is displayed. At the bottom right of the dialog, the 'Save' button is circled in blue.</p>

Explanation	Screenshot
Click on the map route icon .	
Select the app name that doesn't have -srv and click "Save".	

Checking each Consumer is connected to a different HANA HDI Container

Explanation	Screenshot
<p>Open the application.</p> <p>Click the Fiori link to see the generated Fiori screens.</p> <p>If you create two different consumers, you will see that the data of this each tenant is different.</p> <p>It can also be observed that boosting the data of one tenant does not cause the data of the other tenant to change.</p> <p>Each tenant is connected to a different SAP HANA Cloud HDI-container.</p>	

Viewing the destinations in the Provider subaccount

Explanation	Screenshot
Multitenancy works for the HANA database as well as the destinations or any other services. A destination is automatically generated by the generator for our application within the provider subaccount under Connectivity >> Destinations .	
Clicking on the destination will display the details of the destination.	

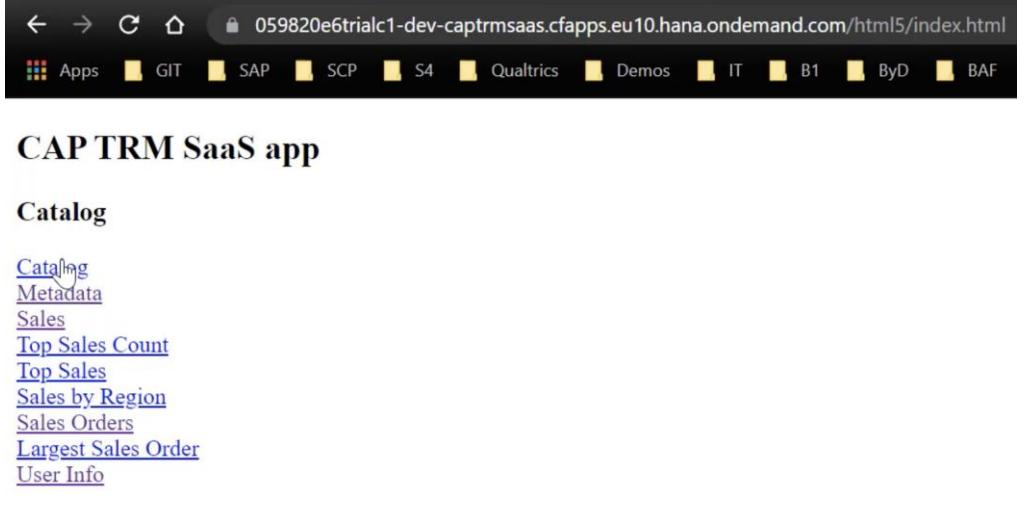
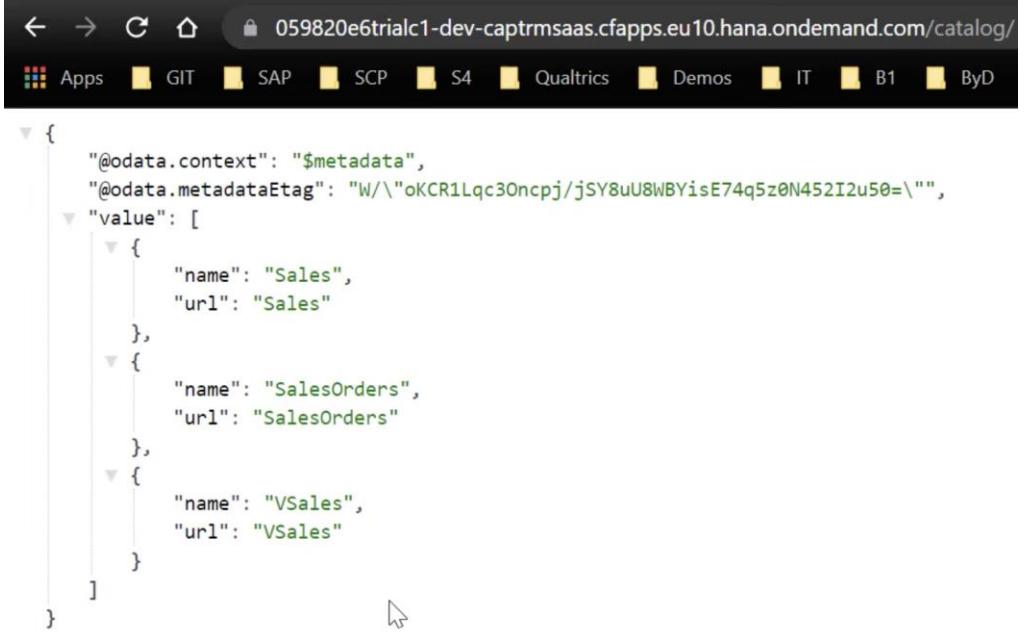
Using a new destination in a Consumer subaccount

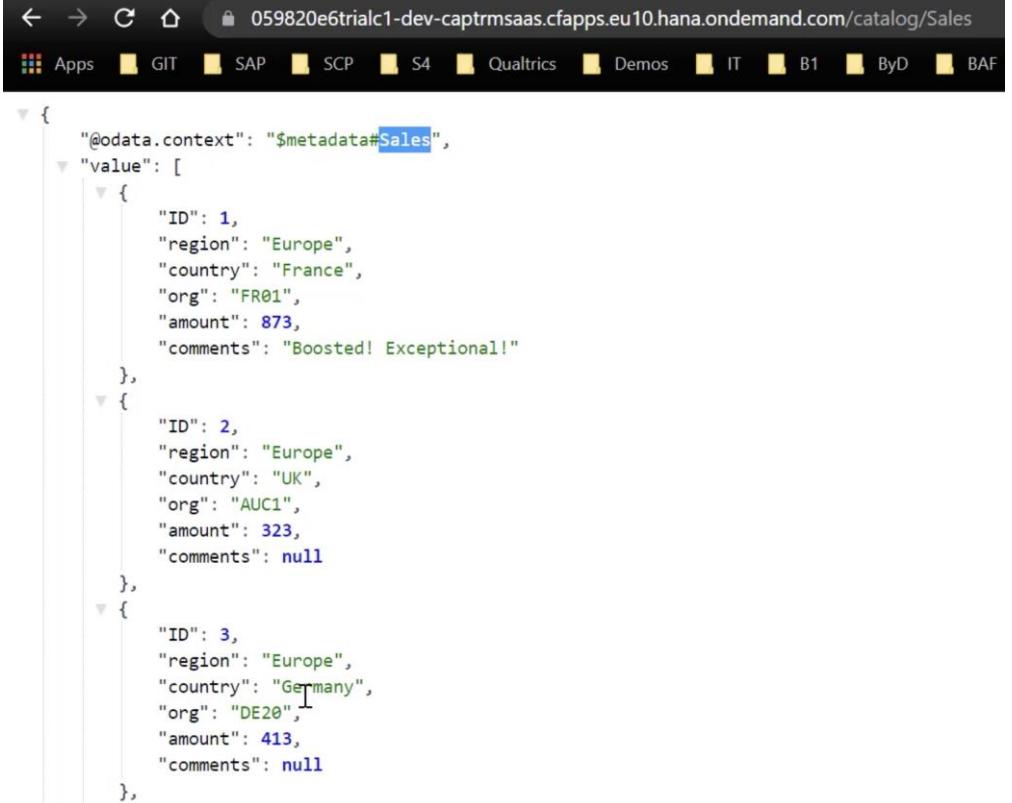
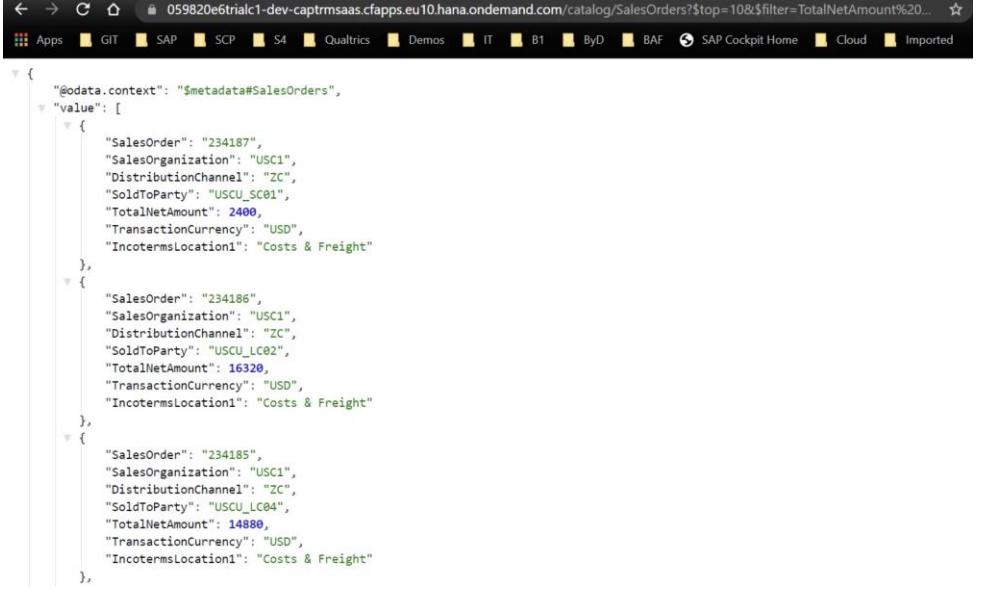
Explanation	Screenshot
For a specific tenant to use a different destination, the destination needs to be created within that tenant to override the default destination. Go to the specific Consumer subaccount and click on Connectivity >> Destinations >> New Destination .	

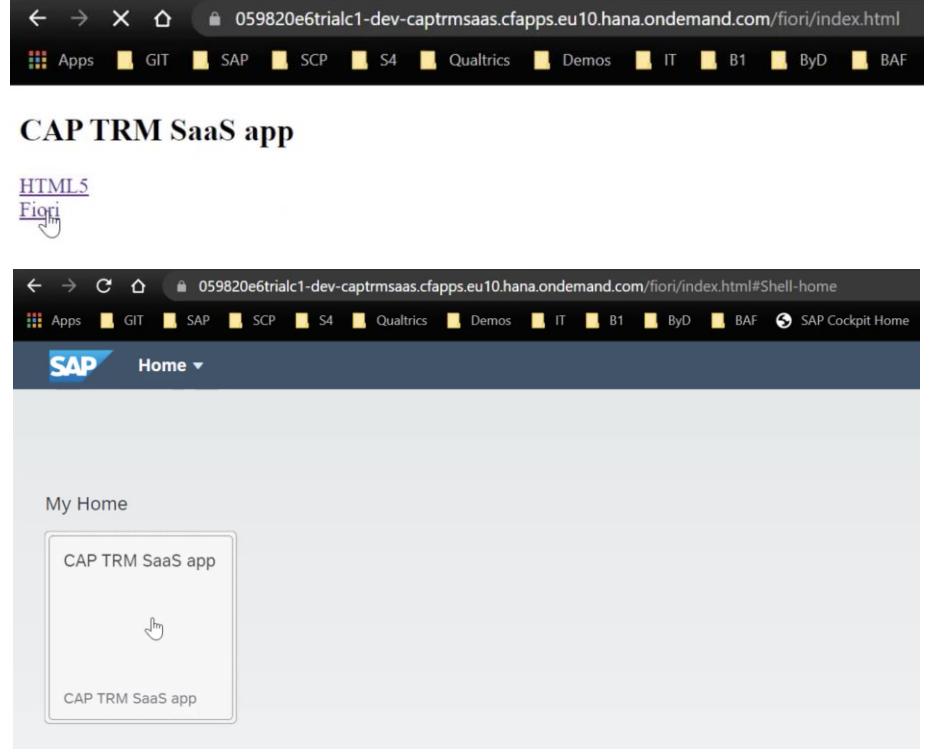
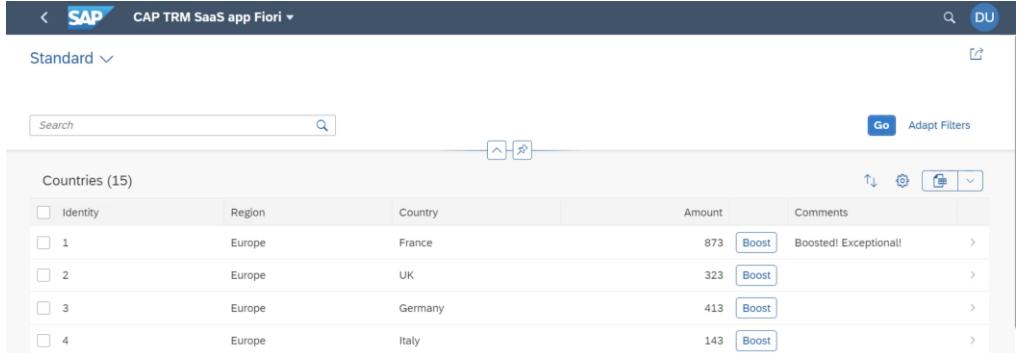
Explanation	Screenshot
Then, enter the details of the new destination while using the same name as the default destination in the Provider subaccount and click on “Save”.	<p>Destination Configuration</p> <p>Name: * captrmsaas-API_SALES_ORDER_SRV</p> <p>Type: HTTP</p> <p>Description: Will fail</p> <p>URL: * https://help.sap.com</p> <p>Proxy Type: Internet</p> <p>Authentication: NoAuthentication</p> <p>New Property</p>

TEST THE APP

Explanation	Screenshot												
In the consumer account, there is no need to provision Cloud Foundry. Only a subscription to the provider needs to be created in the consumer account.	<p>Trial Home / 059820e6trial / SaaSConsumer1</p> <p>Subaccount: SaaSConsumer1 - Instances and Subscriptions Create</p> <p>Search All Services All Plans All Statuses</p> <p>Subscriptions (1) Instances (0) Environments (0)</p> <p>Applications to which your subaccount is currently subscribed</p> <table border="1"> <thead> <tr> <th>Application</th> <th>Plan</th> <th>Created On</th> <th>Changed On</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>CAP TRM SaaS app</td> <td>default</td> <td>03/06/2021</td> <td>03/06/2021</td> <td>Subscribed</td> </tr> </tbody> </table>	Application	Plan	Created On	Changed On	Status	CAP TRM SaaS app	default	03/06/2021	03/06/2021	Subscribed		
Application	Plan	Created On	Changed On	Status									
CAP TRM SaaS app	default	03/06/2021	03/06/2021	Subscribed									
To go to the provisioned app, click on the “Go to Application” link.	<p>Trial Home / 059820e6trial / SaaSConsumer1</p> <p>Subaccount: SaaSConsumer1 - Instances and Subscriptions Create</p> <p>Subscriptions (1)</p> <p>CAP TRM SaaS app</p> <p>Plan: default Status: Subscribed Created On: 03/06/2021 Changed On: 03/06/2021</p> <p>Overview Roles</p> <p>Roles by Role Template</p> <table border="1"> <thead> <tr> <th>Role Template</th> <th>Role Name</th> <th>Description</th> <th>Attributes</th> <th>Role Collections</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td>Admin</td> <td>Admin</td> <td>Administrator</td> <td>0</td> <td>1</td> <td>+ / edit / delete</td> </tr> </tbody> </table> <p>Go to Application</p>	Role Template	Role Name	Description	Attributes	Role Collections	Actions	Admin	Admin	Administrator	0	1	+ / edit / delete
Role Template	Role Name	Description	Attributes	Role Collections	Actions								
Admin	Admin	Administrator	0	1	+ / edit / delete								
The provisioned app is launched via a URL of the specific consumer.	<p>https://059820e6trial1-dev-captrmsaas.cfapps.eu10.hana.ondemand.com/index.html</p> <p>Apps GIT SAP SCP S4 Qualtrics Demos IT B1 ByD BAF</p> <p>CAP TRM SaaS app</p> <p>HTML5 Fiori</p>												

Explanation	Screenshot
<p>Clicking on the HTML5 link loads a list of services.</p>	 <p>CAP TRM SaaS app</p> <p>Catalog</p> <ul style="list-style-type: none"> Catalog Metadata Sales Top Sales Count Top Sales Sales by Region Sales Orders Largest Sales Order User Info
<p>Example: The Catalog Service lists all the entities.</p>	 <pre> { "@odata.context": "\$metadata", "@odata.metadataEtag": "W/\\"oKCR1Lqc30ncpj/jSY8uU8WBYisE74q5z0N452I2u50="", "value": [{ "name": "Sales", "url": "Sales" }, { "name": "SalesOrders", "url": "SalesOrders" }, { "name": "VSales", "url": "VSales" }] } </pre>

Explanation	Screenshot
<p>Example: The Sales Service lists all the sales data stored in the corresponding HANA tenant's schema.</p>	 <pre> { "@odata.context": "\$metadata#Sales", "value": [{ "ID": 1, "region": "Europe", "country": "France", "org": "FR01", "amount": 873, "comments": "Boosted! Exceptional!" }, { "ID": 2, "region": "Europe", "country": "UK", "org": "AUC1", "amount": 323, "comments": null }, { "ID": 3, "region": "Europe", "country": "Germany", "org": "DE20", "amount": 413, "comments": null }] } </pre>
<p>Example: The SalesOrders Service lists all the sales orders from the S/4HANA Cloud backend.</p>	 <pre> { "@odata.context": "\$metadata#SalesOrders", "value": [{ "SalesOrder": "234187", "SalesOrganization": "USC1", "DistributionChannel": "ZC", "SoldToParty": "USCU_SC01", "TotalNetAmount": 2400, "TransactionCurrency": "USD", "IncotermsLocation": "Costs & Freight" }, { "SalesOrder": "234186", "SalesOrganization": "USC1", "DistributionChannel": "ZC", "SoldToParty": "USCU_LC02", "TotalNetAmount": 16320, "TransactionCurrency": "USD", "IncotermsLocation": "Costs & Freight" }, { "SalesOrder": "234185", "SalesOrganization": "USC1", "DistributionChannel": "ZC", "SoldToParty": "USCU_LC04", "TotalNetAmount": 14880, "TransactionCurrency": "USD", "IncotermsLocation": "Costs & Freight" }] } </pre>

Explanation	Screenshot																									
<p>Clicking on the Fiori link from the main application loads a Fiori application.</p>	 <p>The screenshot shows two consecutive screenshots of a web browser. The top screenshot displays the SAP Fiori launchpad with various application icons like Apps, GIT, SAP, SCP, S4, Qualtrics, Demos, IT, B1, ByD, and BAF. A mouse cursor hovers over the 'Fiori' link. The bottom screenshot shows the SAP Fiori launchpad again, but the 'CAP TRM SaaS app' icon is now highlighted, indicating it has been selected.</p>																									
<p>Clicking on the CAP TRM SaaS app, loads the app. Clicking on the “Go” button in the app loads the data from the database. And the “Boost” button can be clicked to boost the amount. The boosted amount can be seen by clicking on the “Go” button again to refresh the data of the app.</p> <p>Each tenant will have its own data. Therefore, boosting an amount in one tenant will only update the amount in that tenant.</p> <p>This can be verified by viewing the data of another consumer tenant.</p>	 <p>The screenshot shows the SAP Fiori launchpad with the 'CAP TRM SaaS app' selected. Below, a detailed view of the app is shown. The interface includes a search bar, a 'Standard' dropdown, and a table titled 'Countries (15)'. The table lists four entries:</p> <table border="1"> <thead> <tr> <th>Identity</th> <th>Region</th> <th>Country</th> <th>Amount</th> <th>Comments</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Europe</td> <td>France</td> <td>873</td> <td>Boost Boosted! Exceptional!</td> </tr> <tr> <td>2</td> <td>Europe</td> <td>UK</td> <td>323</td> <td>Boost</td> </tr> <tr> <td>3</td> <td>Europe</td> <td>Germany</td> <td>413</td> <td>Boost</td> </tr> <tr> <td>4</td> <td>Europe</td> <td>Italy</td> <td>143</td> <td>Boost</td> </tr> </tbody> </table>	Identity	Region	Country	Amount	Comments	1	Europe	France	873	Boost Boosted! Exceptional!	2	Europe	UK	323	Boost	3	Europe	Germany	413	Boost	4	Europe	Italy	143	Boost
Identity	Region	Country	Amount	Comments																						
1	Europe	France	873	Boost Boosted! Exceptional!																						
2	Europe	UK	323	Boost																						
3	Europe	Germany	413	Boost																						
4	Europe	Italy	143	Boost																						

GENERAL TROUBLESHOOTING

Troubleshooting general application errors and crashes

Check the logs of your router or srv app by running the following command from a Terminal on your Business Application Studio.

During execution:

cf logs replacewithyourappname

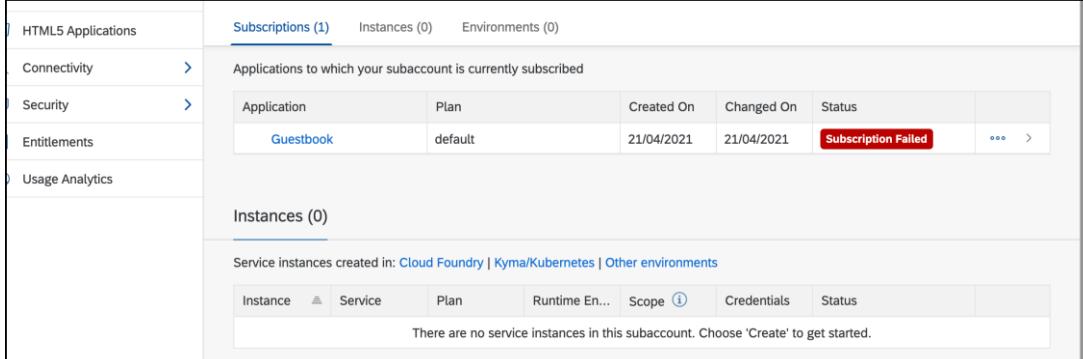
After execution:

cf logs replacewithyourappname --recent

Most of the issues occur on the srv app as this application is the one in charge of the subscription process as well as the access to the HANA services.

What to do when I get a red “Subscription failed” message?

1. Check the logs of the srv app running in your account from a Terminal as explained in the section before
2. Check the Network logs from your browser Dev Tools.

Explanation	Screenshot
There can be multiple reasons for a “Subscription failed” error message.	 <p>The screenshot shows the Cloud Foundry interface for managing subscriptions. On the left, there's a sidebar with options like HTML5 Applications, Connectivity, Security, Entitlements, and Usage Analytics. The main area is titled "Subscriptions (1)" and shows a table with one row. The table columns are Application, Plan, Created On, Changed On, and Status. The "Guestbook" application has a "default" plan, was created on 21/04/2021, and last changed on 21/04/2021. The "Status" column is red and displays the text "Subscription Failed". Below this table, there's a section for "Instances (0)" with a note: "Service instances created in: Cloud Foundry Kyma/Kubernetes Other environments". A table header for "Instances" is shown but remains empty.</p>

Explanation

You could inspect the payload of the request that returned the error message:

1. Open the network tab of the dev tools of your browser.
2. Filter all requests for getCFSaaSApplications.
3. Find the object within the array that represents your subscription.
4. Inspect the property stateDetails.message.

The cause is usually mentioned in the last sentence of this string (as shown in the screenshot). In this example, the error message indicates that the application is not running.

Screenshot

The screenshot shows the SAP BTP Cockpit interface. The left sidebar includes sections for Overview, Services (selected), Subscriptions, HTML5 Applications, Connectivity, Security, Entitlements, and Usage Analytics. The main area displays the 'Subaccount: Consumer' section. Under 'Subscriptions', there is one entry: 'Subscriptions (1)' for 'Guestbook'. Below it, 'Instances (0)' and 'Environments (0)' are listed. At the bottom of the sidebar are 'Useful Links' and 'Legal Information'. The right side of the screen is the 'Network' tab of the developer tools. It shows a single request labeled 'getCFSaaSApplications'. The 'Preview' tab is selected, displaying a JSON response. The response contains an array of objects, with the first one being the relevant subscription. The 'stateDetails.message' field of this object contains the error message: 'subscribe failed. Error: subscribe failed. Parameters: rootSubscription: RootSubscription {...}'. The 'Timing' tab shows a timeline with several green bars indicating request and response times.

TO GO DEEPER

This exercise steps have produced after following the YouTube playlist [SAP Business Technology Platform Multitenant Business Applications](#). Don't hesitate to review the great videos proposed by this playlist to go deeper into any of the SaaS Multitenancy topics covered by this exercise.