

SAP HANA Cloud

Collaborative Database Development in SAP HANA Cloud, SAP HANA Database

**Session 3: Creating Project Templates
for Team Members in SAP Business Application Studio**

**SAP Data and Analytics
Virtual Forum:
Accelerating Outcomes**

GET STARTED

WELCOME	4
ABOUT THIS SESSION	4
PREREQUISITES	5
INTRODUCTION	6
1. CREATE AN EMPTY GIT PROJECT	7
2. CREATE AN EMPTY SAP BUSINESS APPLICATION STUDIO PROJECT	8
SYNCHRONIZE THE PROJECT WITH THE GITHUB REPOSITORY	11
3. GENERATE USERS AND ROLES FOR ASSIGNING PRIVILEGES	14
4. ENHANCE THE PROJECT WITH SERVICES, SYNONYMS AND GRANTS	17
CREATE A USER-PROVIDED SERVICE IN SAP BAS	17
GRANT PRIVILEGES USING USER-PROVIDED SERVICE	19
CREATE A SYNONYM	20
HOW TO COLLABORATE ON THE TEMPLATE PROJECT	23
STEPS FOR PROJECT OWNER:	23
SYNCHRONIZE THE PROJECT WITH GIT	23
INVITE A COLLABORATOR TO YOUR GIT REPOSITORY	23
STEPS FOR TEMPLATE USER:	26
CLONE THE TEMPLATE PROJECT FROM GIT	26
DEPLOY THE TEMPLATE PROJECT TO AN HDI CONTAINER	28
CREATE A SIMPLE CALCULATION VIEW	30

ADD A DATA SOURCE TO THE CALCULATION VIEW	31
ADD A CALCULATED COLUMN TO THE PROJECTION NODE	34
DEPLOY THE VIEW AND PREVIEW THE DATA	37
SUMMARY	40
ADDITIONAL RESOURCES	41

<https://community.sap.com/topics/hana-cloud>

© 2021 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. Service names mentioned are the trademarks of their respective companies. See www.sap.com/copyright for additional trademark information and notices.

WELCOME

Join the SAP HANA Cloud Workshop series to learn from experts and follow along live using your own SAP HANA Cloud (trial) account.

Learn in this workshop how database development in SAP HANA Cloud and SAP Business Application Studio can be done collaboratively. You will get to know basic HDI concepts, ways to develop applications and to automate their deployment as well as creating template projects that others in your organization can easily use and modify.

About this session

Are you wondering how to collaborate more efficiently when working in development projects? This session will give you the opportunity to create template projects that help you set up service connections, handle synonyms and prepare authorization management. You can create these templates once in the beginning and give your project team members a head-start. Using the template they can build on your work and can quickly start working productively. In the final session of this workshop, you will learn how to:

- Create templates to reduce the effort for other team members
- Prepare templates to easily connect to data sources
- Make use of roles to prepare granting of required privileges
- Create a calculation view that others can use for reporting
- Prepare for reporting, like with SAP Analytics Cloud
- Share the templates with team members

Your presenters:



Volker Saggau

Product Manager, SAP HANA Cloud | SAP



Jan Zwickel

Product Manager, SAP HANA Cloud | SAP

Prerequisites

Here is what you need to prepare before the workshop starts to be able to follow along:

1. [Sign up](#) for the free SAP HANA Cloud trial.
2. Provision an instance of SAP HANA Cloud, SAP HANA database and make sure it is running.
3. Subscribe to the free SAP Business Application Studio trial.
 - a. A running **SAP HANA Native Application Dev Space** (you can use the one created in session 1)
4. Subscribe to the CI/CD service in BTP cockpit.
5. Create a free account on www.github.com to use the required sample data.
 - a. Have your GitHub token ready (you can use the one created in session 1).

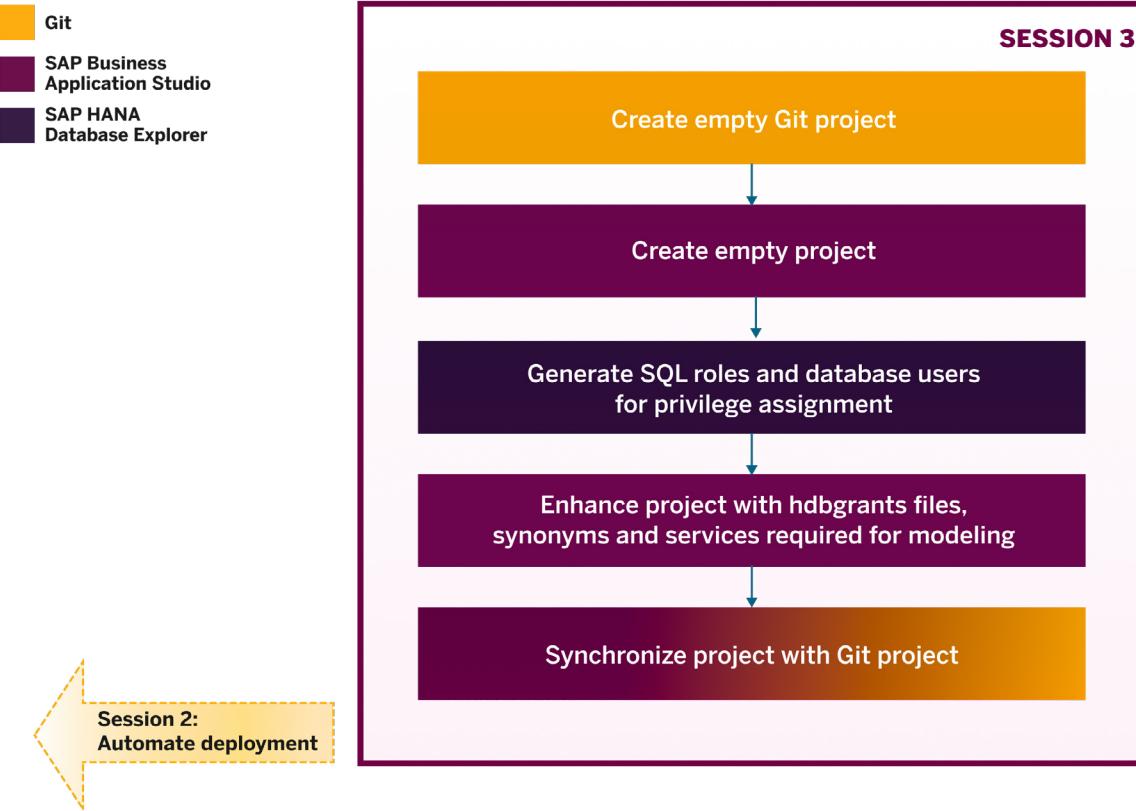
The steps needed to prepare are all covered in our [preparation reader](#).

If you do, however, have any issues, please ask a question in the [SAP Community](#) using the tag SAP HANA Cloud or refer to the [SAP HANA Cloud Technical Documentation](#).

The steps documented in this workbook require completion of the previous sessions of this workshop series, especially the creation of the INVENTORY table, the data import and the creation of two roles in session 1.

Introduction

In this session, you will focus on how to collaborate on a project within a team. The main goal is to show you how to create a template project in SAP Business Application Studio, that your colleagues can re-use and modify without having to prepare services and synonyms each time they wish to collaborate on the project. This session will also illustrate to you the advantage of collaborating through a template project using a sample scenario.



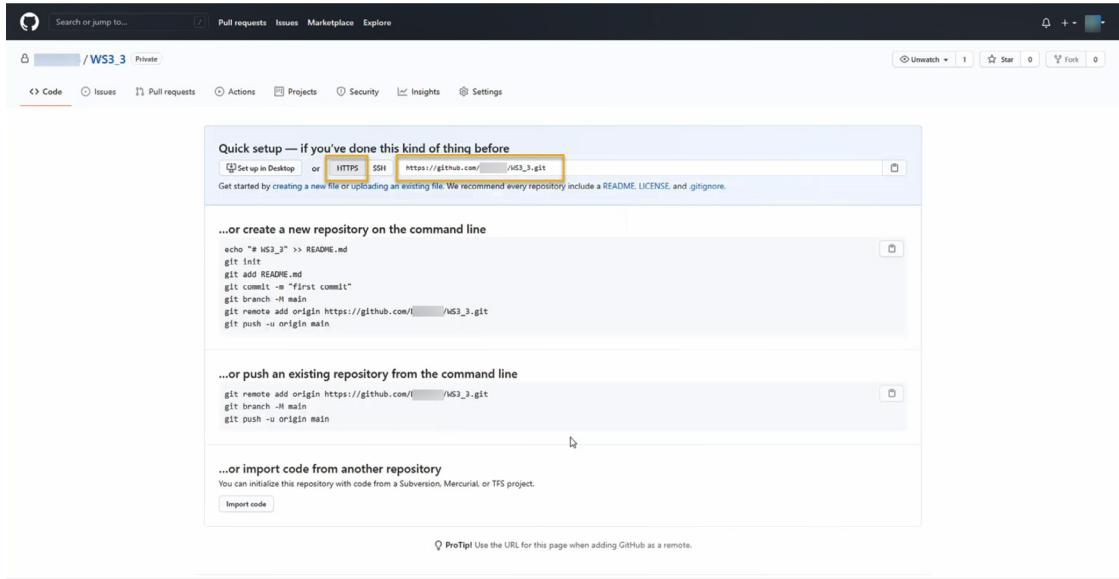
You will begin this session by creating an empty Git project that you can share with your colleagues later. Next, you will create an empty project in SAP Business Application Studio. Subsequently, you will generate roles and users in SAP HANA Database Explore, to assign the necessary privileges for collaborating on the project. Afterwards, you will generate hdbgrants files, synonyms, and services within the project. Finally, you will synchronize the project with your GitHub repository and share the Git project with your colleague.

In the last part of this session, we will illustrate how a collaborator can work with the template project you have created. You can simulate these steps in your own environment by creating another project in your dev space. The instructions that are meant for collaborators to use the template are marked in this workbook accordingly.

1. CREATE AN EMPTY GIT PROJECT

First, we will create an empty git project that will be used for the template project created in SAP Business Application Studio.

1. Open github.com. Sign in to your GitHub account using your credentials.
2. Click on the profile icon in the top right corner and select **Your repositories**.
3. In the top right corner, click on **New** to create a new repository.
4. Under **Create a new repository**, enter **WS3_3** as the Repository name.
5. Optionally, you can add a short description.
6. You can choose to make your repository visible publicly or privately. We suggest you select the option **Private**.
7. Click on **Create repository** to finish.
8. Now, you will be taken to a Quick setup page for your new repository. Copy the HTTPS connection link for the repository to a safe location. This is required to set up the connection with your SAP Business Application Studio project.



2. CREATE AN EMPTY SAP BUSINESS APPLICATION STUDIO PROJECT

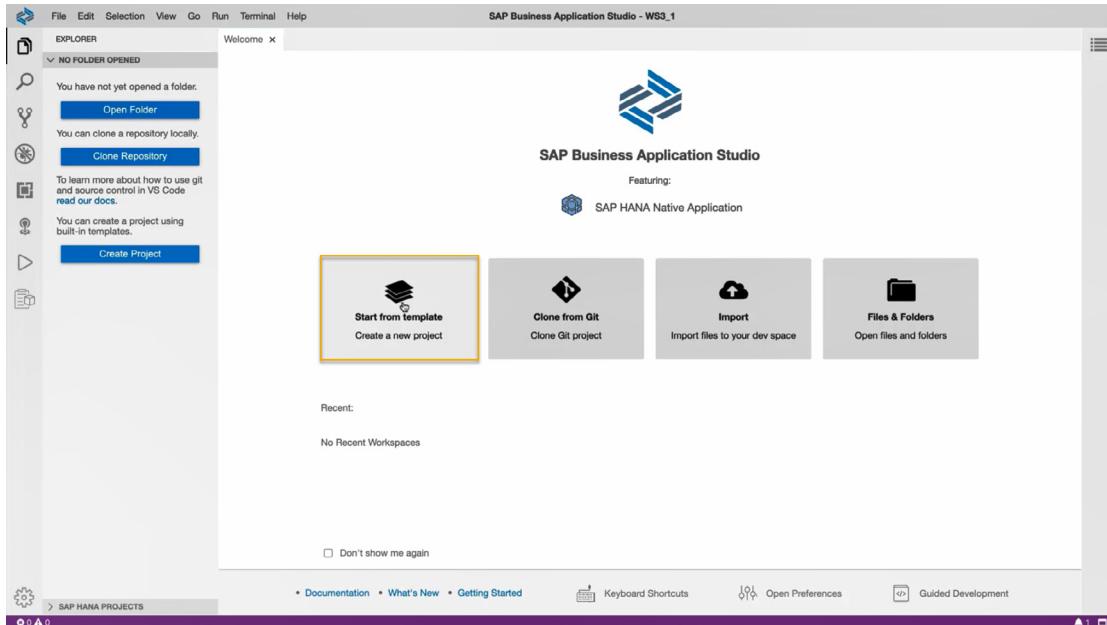
Next, we will create an SAP Business Application Studio project. For that, you need a running dev space with the SAP HANA Native Application configuration.

Important: In a trial landscape, you can only have two dev spaces at a time with only one of them running. If you have followed the previous sessions of this workshop, you will already have two dev spaces

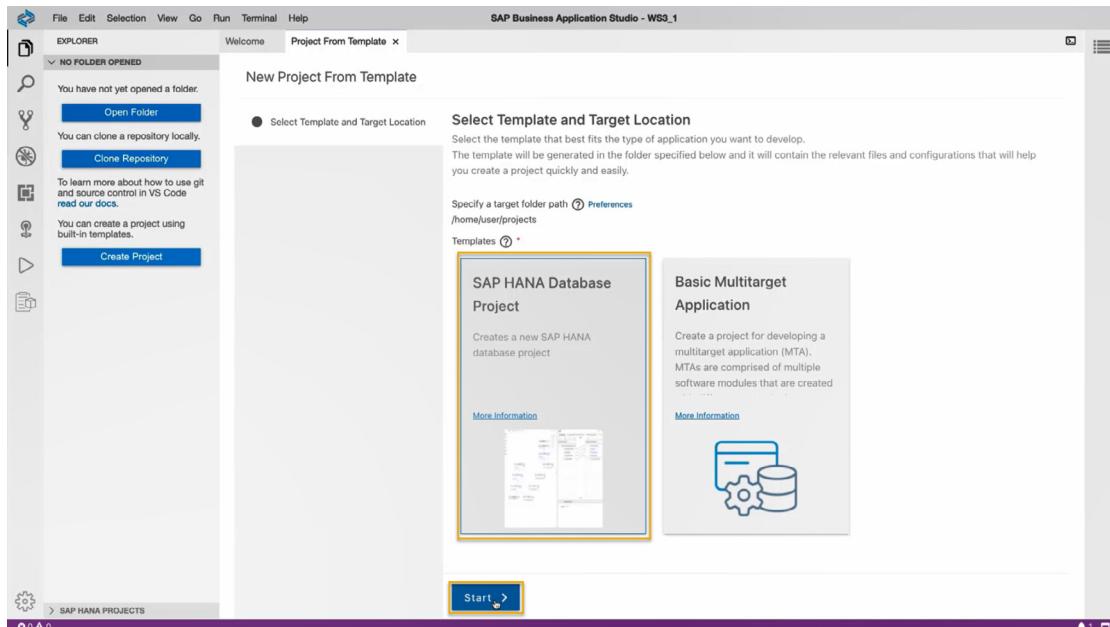
- One SAP HANA Native Application dev spaces created in session 1
- One Full Stack Cloud Application dev space created in session 2

If you have followed along the two sessions, you can simply restart the SAP HANA Native Application dev space created in session 1 and follow the instructions below.

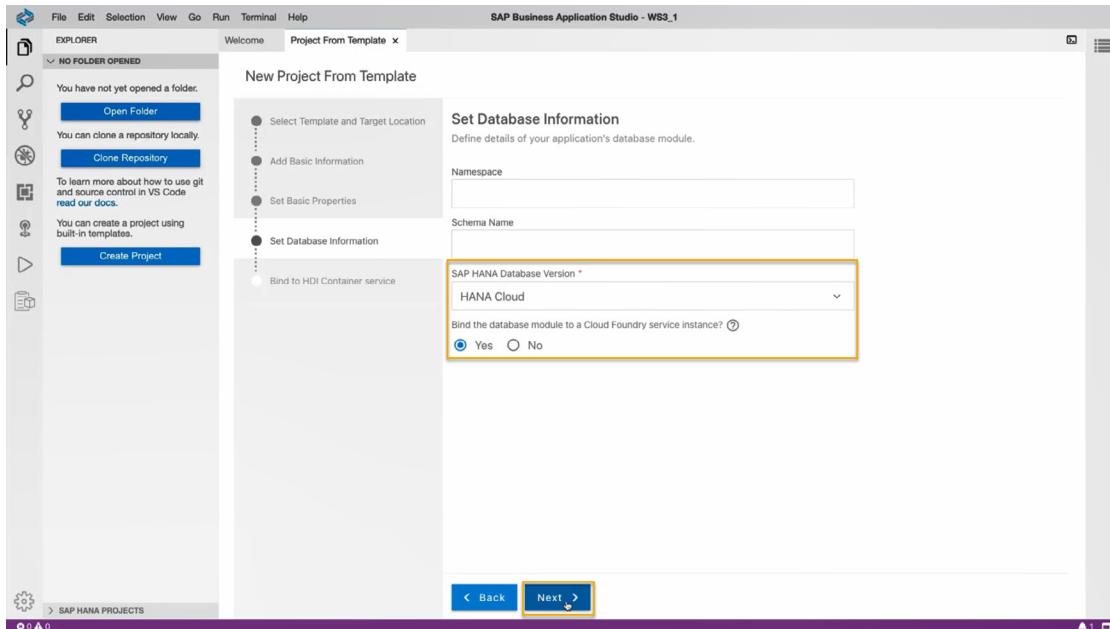
1. Open **SAP Business Application Studio**.
2. On the page that shows your Dev Spaces in SAP Business Application Studio, restart the Dev Space **WS3_1** that you created for your first workshop session.
3. Once the status has changed to RUNNING, click on the Dev Space **WS3_1** to open it.
4. After you have opened your Dev Space in SAP Business Application Studio, select **Start from template** from the **Welcome** screen to create a new project.



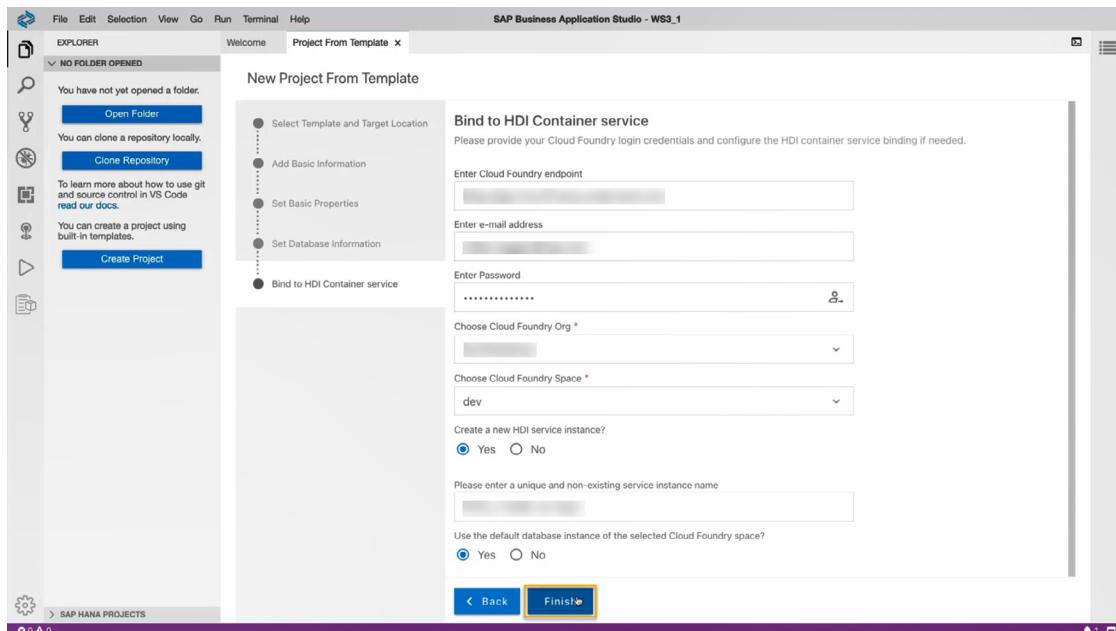
5. Choose SAP HANA Database Project template and click on Start.



6. In the step **Add Basic Information**, name your new Project as **WS3_3**. Click on **Next**.
7. In the step **Set Basic Properties**, we suggest you keep the database module name as **db**. Click on **Next**.
8. In the step **Set Database Information**, we recommend not to fill in the fields **Namespace** and **Schema Name** for the purpose of this workshop.
9. Set **SAP HANA Database Version** as **HANA Cloud**.
10. Select **Yes** for the option '**Bind the database module to a Cloud Foundry service instance?**'. Click on **Next**.



11. In the step **Bind to HDI Container service**, sign in with your trial account credentials and click on the  login icon to make the project target your trial account. This will automatically select your Cloud Foundry Organization and Cloud Foundry Space.
12. Select **Yes** for the fields **Create a new HDI service instance?** and **Use the default database instance of the selected Cloud Foundry space?**. A unique and non-existing service instance name will be generated automatically for you. This generates a HDI container in your SAP HANA Cloud, SAP HANA database.



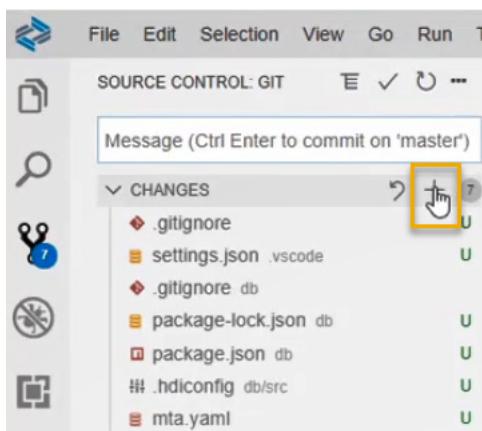
13. Click on **Finish**. It might take a few minutes to complete the process.

Synchronize the project with the GitHub repository

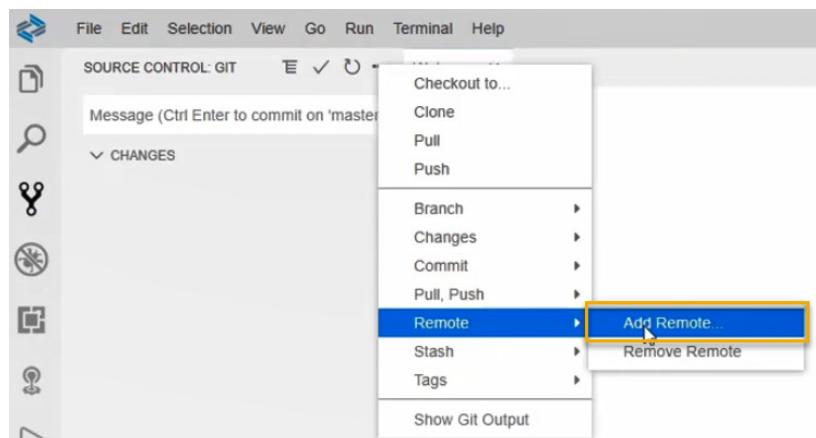
1. Your project should automatically open in a new workspace. If it does not, open the project in a new workspace or click on the project name WS3_3 on the welcome screen.
2. On the left side menu of your SAP BAS environment, click on the  icon to go to the **Source Control** panel. Here, you will see all the recent updates to your project that have not been pushed to Git yet.
3. Next, click on the **plus icon** to add a local git repository.



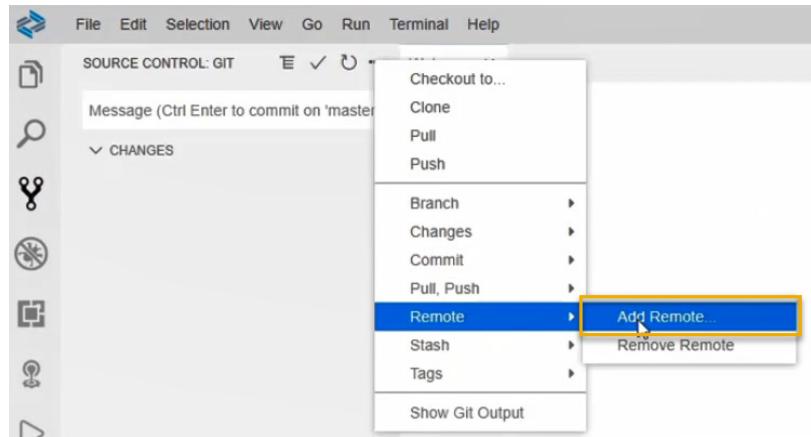
4. In the Source Control panel, you can see a list of changes made when the empty project WS3_3 was created. Click on the plus icon to stage these changes.



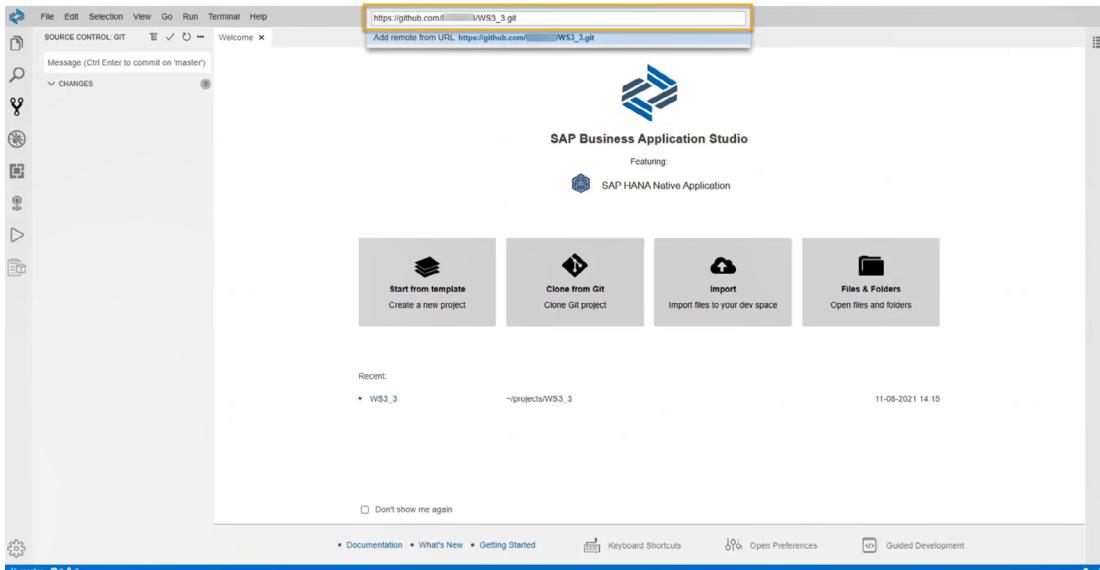
5. Enter a commit message in the Message box (for example: Initial commit).
6. Click on **tick mark icon** above to commit the changes.



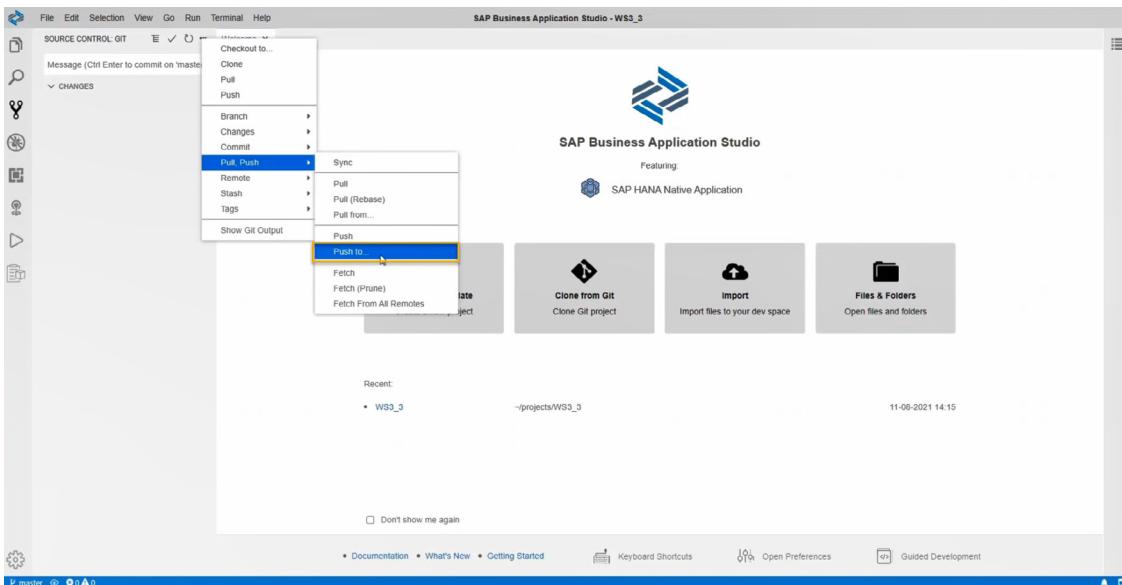
7. Next, click on the **three dots** icon in the same navigation area.
8. Select **Remote** and then **Add Remote**. This will add the remote connection to your GitHub repository



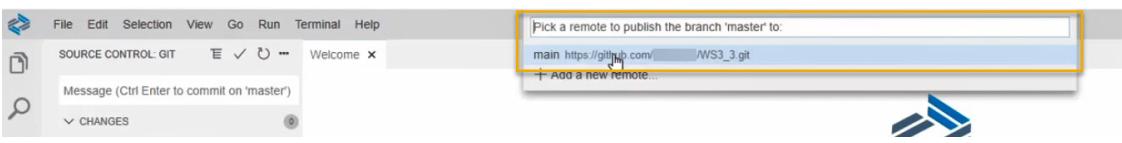
9. A prompt asking to provide repository URL appears on the top center of the screen. Paste the GitHub **HTTPS connection link** that you obtained from GitHub after creating your new repository WS3_3 into the prompt. Press **Enter**.



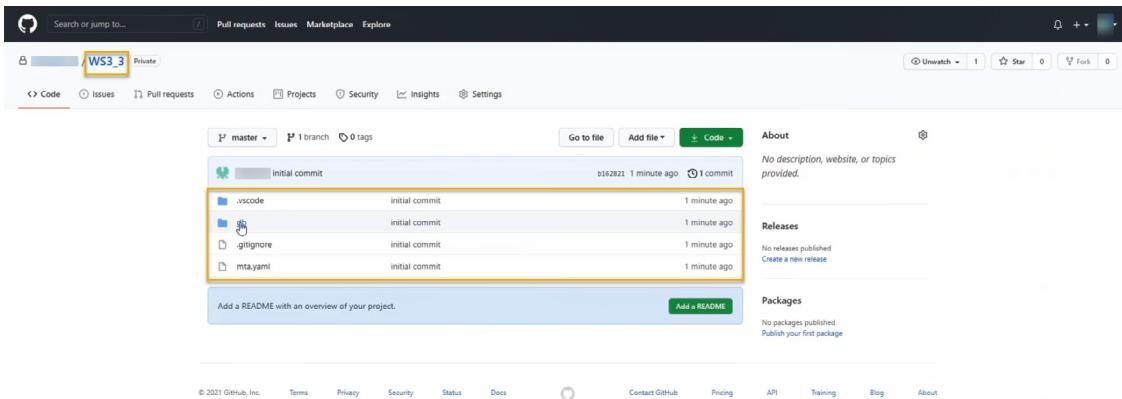
10. Now, enter the remote name "**main**". Press **Enter**.
11. Next, enter your **GitHub username** and press **Enter**.
12. Enter your **Git account token**. If you have already generated a token in the first session of this workshop series, use that token here. Press **Enter**.
13. Finally, we recommend that you select to save your credentials for this session only. You can also select **No** in the notification box asking for periodically running git fetch, as shown below.
14. Go to the Source Control panel and click on the three dots icon. Select **Pull, Push** and then the **Push to...** option.



15. A prompt asking for a remote to publish the master branch appears on the top center of the screen. Select the remote connection **main** that you have just created.



16. You can verify the synchronization by checking your GitHub repository **WS3_3**. You should now see the folders and objects from your SAP Business Application Studio project there, for example, the **db folder** and **mta.yaml file**.



3. GENERATE USERS AND ROLES FOR ASSIGNING PRIVILEGES

In this section, you will create two generic roles and database users. Then, you will assign these roles to a user that grants project so that you can add new privileges to these roles in the future.

1. Go to your SAP BAS environment, press **F1 key** or click on **View – Find Command** at the top of the screen to open a **search prompt**. Enter **Data** in the field and from the dropdown list of functionalities, choose **SAP HANA: Open Database Explorer**.
2. In the SAP HANA Database Explorer, right-click on your HDI container database connection on the left-side panel and select **Open SQL Console**.



Note: It is necessary to use the DBADMIN user connected to the instance since you require the DBADMIN user privileges to create and assign the roles.

3. In the SQL Console, copy and paste the following code. Run it to create a role for the Object Owner (OO). The Object Owner, in this example, will be assigned to the user who owns the project and creates calculation views. This role will need Select privileges with grant option that will be assigned to it in the next steps.

```
create role "genericRoleForOO";
```

4. Next, copy and paste the following code in the SQL Console and run it to create a role for application users. The Application User, in this example, will be assigned to the user who uses the data preview on the calculation views created by the object owner. This role will need Select privileges that will be assigned to it in the next steps.

```
(AP). create role "genericRoleForAP";
```

5. Copy and Paste the following code in the SQL Console. Run the code to view the schema name for the role **inventory** created in the first session of the workshop.

```
select role_schema_name,* from roles where role_name='inventory';
```

6. Note the schema name as it is needed for the next statement.

The screenshot shows the SAP Business Application Studio interface with the SQL Console tab active. The SQL Editor window contains the following code:

```
SQL Console 1.sql x
1 create role "genericRoleForOO";
2 create role "genericRoleForAP";
3
4 select role_schema_name,* from roles where role_name='inventory';
```

The Result window shows the output of the last query:

ROLE_SCHEMA_NAME	ROLE_SCHEMA_NAME	ROLE_NAME	ROLE_ID	ROLE_MODE	GLOBAL_IDENTITY	CREATOR	CREA
WS3_1_HDI_DB_1	WS3_1_HDI_DB_1	Inventory	158226	LOCAL	NULL	WS3_1_HDI_DB_1#OO	2021-08-04 1

7. Next, you will assign the two roles you have created in session 1 ("inventory_with_grant" and "inventory") to the two new roles you have just created. Copy and Paste the following code in the SQL Console and make sure the schema name ("WS3_1_HDI_DB_1") matches the schema name you have just checked with the previous statement. Run the statement to grant roles to the Object Owner and Application user.

```
grant "WS3_1_HDI_DB_1"."inventory_with_grant" to "genericRoleForOO";
grant "WS3_1_HDI_DB_1"."inventory" to "genericRoleForAP";
```

8. Create a database user by running the following SQL statement. UPS_GRANTOR means User-Provided Service Grantor. In the next step, this user will be allowed to grant the roles created above.

```
create user UPS_GRANTOR password "DnATBG!1" NO FORCE_FIRST_PASSWORD_CHANGE;
```

9. Assign the roles you have created to this user UPS_GRANTOR by running the statements below.

```
grant "genericRoleForOO" to UPS_GRANTOR WITH ADMIN OPTION;
grant "genericRoleForAP" to UPS_GRANTOR WITH ADMIN OPTION;
```

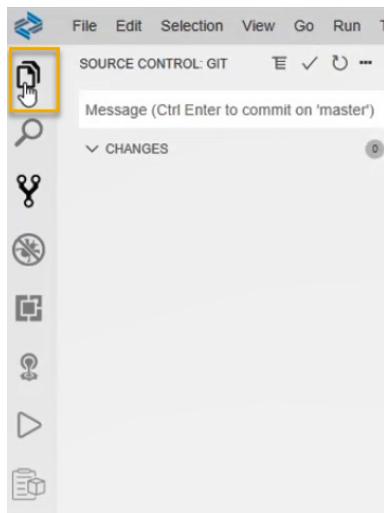
Now the user UPS_GRANTOR is assigned both roles that were created in session 1.

4. ENHANCE THE PROJECT WITH SERVICES, SYNONYMS AND GRANTS

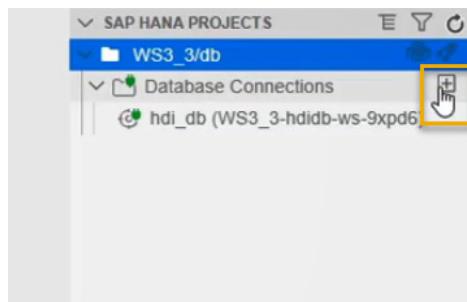
Create a User-Provided Service in SAP BAS

The user-provided service will be responsible for granting privileges to the project in the future.

1. Go back to your SAP Business Application Studio tab and to your project WS3_3. If it is not open already, you can find it on the Welcome screen.
2. Navigate to the **File Explorer** if you are still in the source control view. You can get there by clicking on the File Explorer icon at the top of the vertical menu bar on the left of the screen.

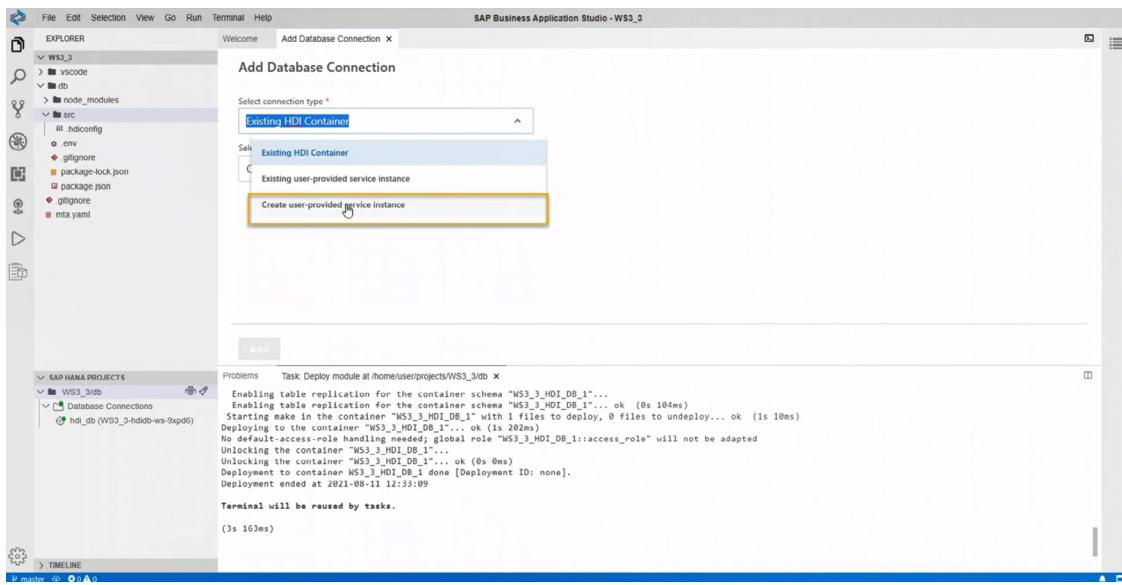


3. On the bottom left corner, you can see the project folder **WS3_3/db** in the SAP HANA Projects panel. Click on the deploy icon for the file to deploy the changes to your HDI container.
4. Next, click on the **plus icon** for adding Database Connections.

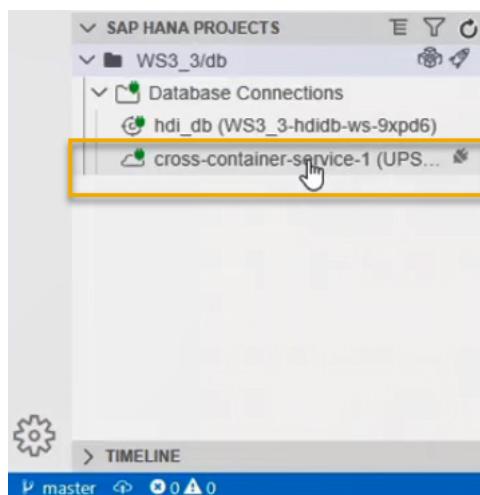


5. The **Add Database Connections** window will open.

- Select **Create a user-provided service instance** from the drop-down menu in the field **Select connection type**.



- Enter the name of the service instance as **UPS_GENERIC**.
- In the field 'Connect to database', select **Use deployment target container database**.
- In the field 'Enter user name', enter **UPS_GRANTOR**.
- In the field 'Enter password', enter the password **DnATBG!1** that you used while creating the user in the SAP HANA Database Explorer using SQL.
- Click on **Add** to complete the creation of the user-provided service instance. You will see this addition in the SAP HANA PROJECTS panel.

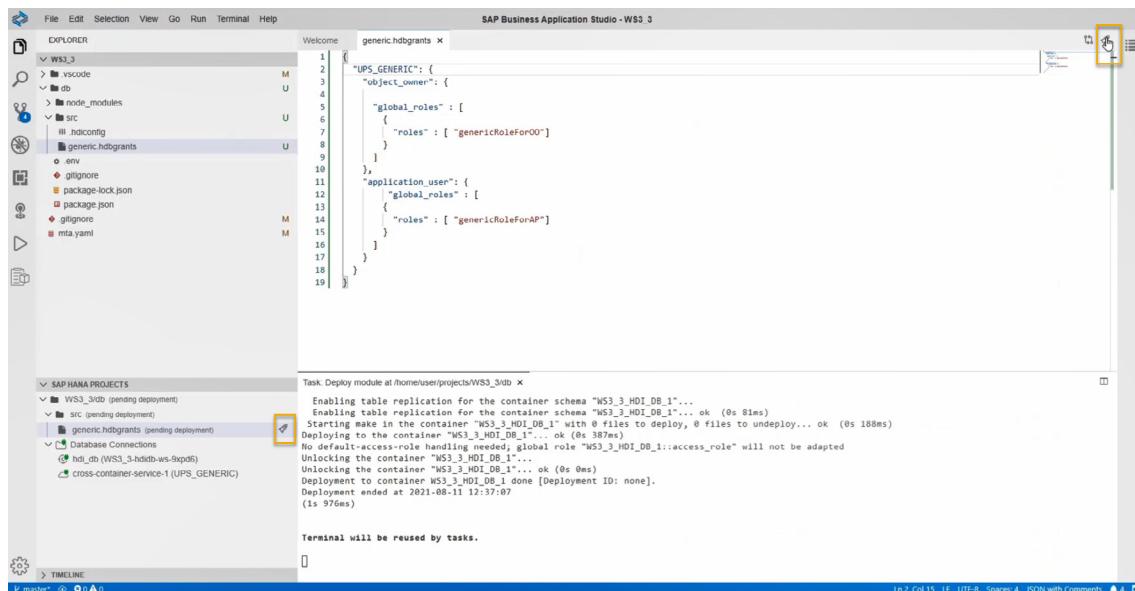


Grant privileges using User-Provided Service

1. Select the src folder from the File Explorer. Press the **F1 key** or click on **View – Find Command** at the top of the screen to open a prompt. Enter **HANA** in the field and from the dropdown list of functionalities, choose **SAP HANA: Create SAP HANA Database Artifact**.
2. In the window for creating a new SAP HANA Database Artifact, click on the folder icon in the first field ("Choose path where you want to create the artifact"). Navigate to your project, the **src** folder, and then click on **Open**.
3. Select the artifact type as **Grants (hdbgrants)**.
4. In the field 'Specify the artifact name' enter the name **generic**.
5. The database version should be **HANA Cloud**, by default. Without making any further changes, click on **Create**.
6. A new file **generic.hdbgrants** can be seen in the src folder in the File Explorer. Click on this file to open the file window.
7. The hdbgrants file comes pre-filled with templates you can use. In this session, we will only need to grant global roles to object owners and roles. Below you can see the code needed to do that. Remove all the default code and copy and paste the code below to the generic.hdbgrants file.

```
{
  "UPS_GENERIC": {
    "object_owner": {
      "global_roles": [
        {
          "roles": [
            "genericRoleForOO"
          ]
        }
      ],
      "application_user": {
        "global_roles": [
          {
            "roles": [
              "genericRoleForAP"
            ]
          }
        ]
      }
    }
  }
}
```

- Click on the **deploy icon** for the file. You can do this either from the SAP HANA Projects panel or from the deploy icon on the top right corner of the file window.

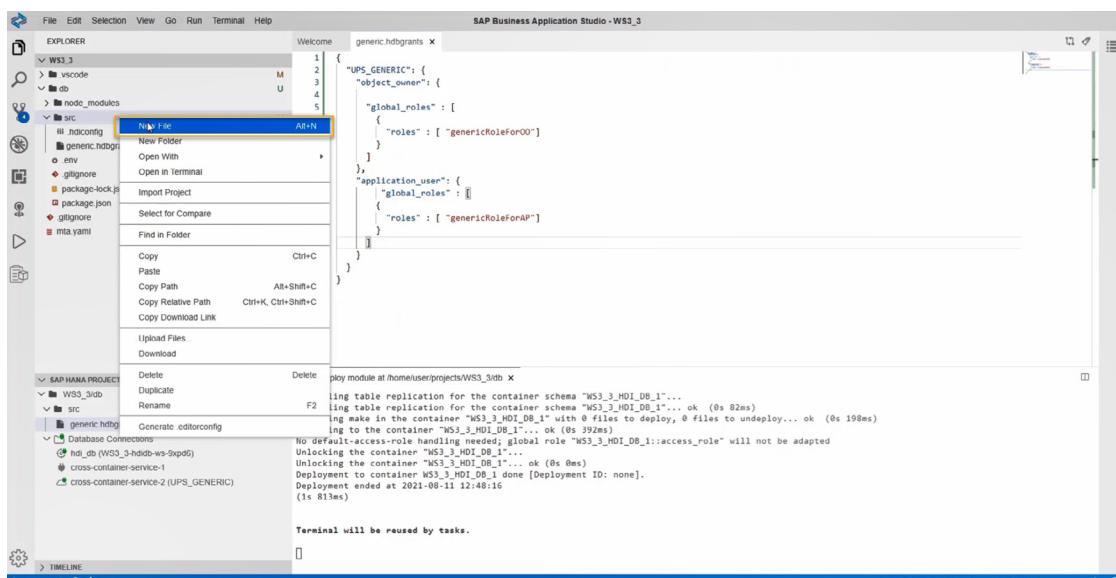


After successful deployment of the generic.hdbgrants file, object owner users and application users will be authorized to report on the data in the container created in the first session of this workshop series.

Create a synonym

Next, you will create a synonym that allows users of the template project to access the inventory table even if they do not have it present in their own HDI container.

- Right-click on the src folder and select **New File**.

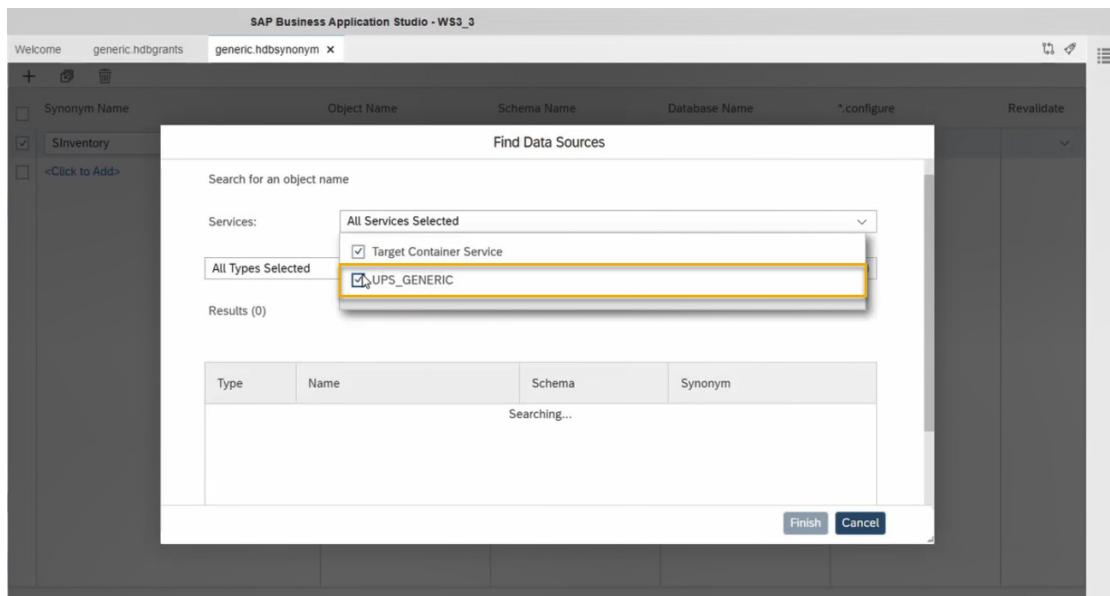


- Name the file as **generic.hdbsynonym**. Click **OK**.

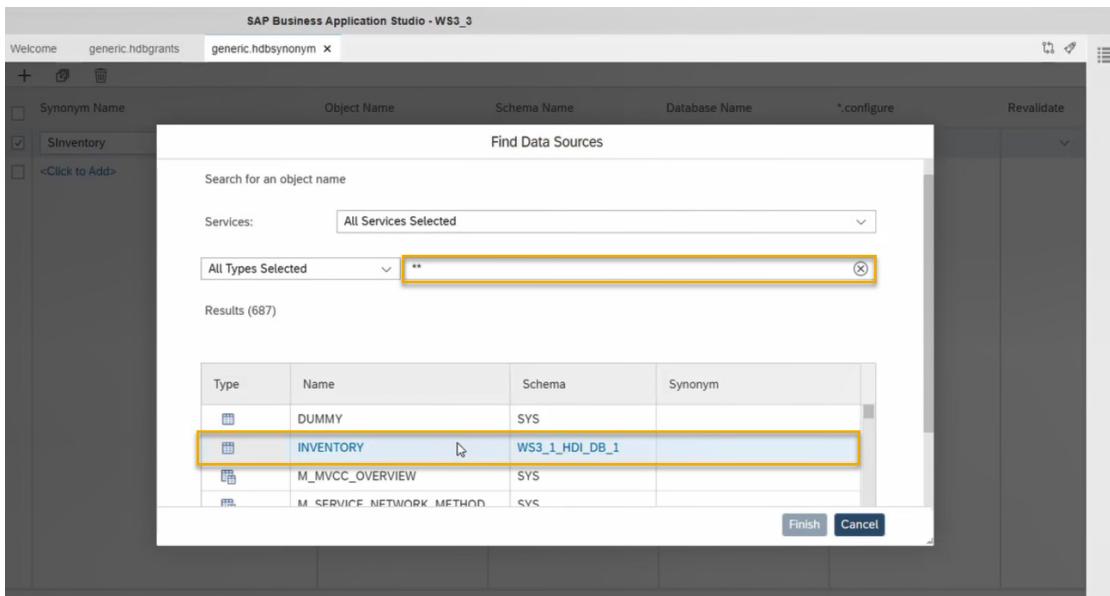
3. A new file named generic.hdbsynonym will be visible under the src folder. Select the file to open the window.
4. Click on **Click to Add** to add a new synonym.



5. Click on the new synonym created under the column Synonym Name and rename it as **SInventory**.
6. In the **Object Name** column, click on the **three dots** icon to search for the object.
7. A new dialog box **Find Data Sources** will open. Make sure to select **UPS_GENERIC** in the services filter.



8. Type ****** in the search bar to view the list of all objects available. Double-click on the object **INVENTORY** contained in the schema used in the first session of this workshop series.



9. Select the synonym **SInventory**.

Note: The **Schema Name** column should have the appropriate schema of the object by default (in this case, **WS3_1_HDI_DB_1**). Make sure this matches the name of your container. If this is not the case for you, please add the name of the schema you obtained in **step 5** of the section [Generate Users and Roles for assigning privileges](#).

10. Click on the **deploy icon** to deploy this synonym.



- 11.** You can check if your synonym was successfully deployed to your database by opening your HDI container icon from the SAP HANA Projects panel.
- 12.** In the SAP HANA Database Explorer select your HDI container from the list of database connections.
- 13.** In the catalog, scroll down to **Synonyms**, clicking on it should reveal the synonym SInventory.
- 14.** Right-click on **SInventory** seen at bottom left- side explorer panel and then select **Open Data**.

You will now view the data from the first session of the workshop series.

And with that, your template project is already done and ready for collaborators to use. Before you can give others access to this project, you will need to synchronize your project with your git repository.

HOW TO COLLABORATE ON THE TEMPLATE PROJECT

The steps in this section illustrate how the template project can be shared with a colleague or a collaborator who works with the project to create a calculation view. The following steps are for the project template owner, the subsequent section outlines the steps for the collaborator, some of which may also be simulated by the template owner.

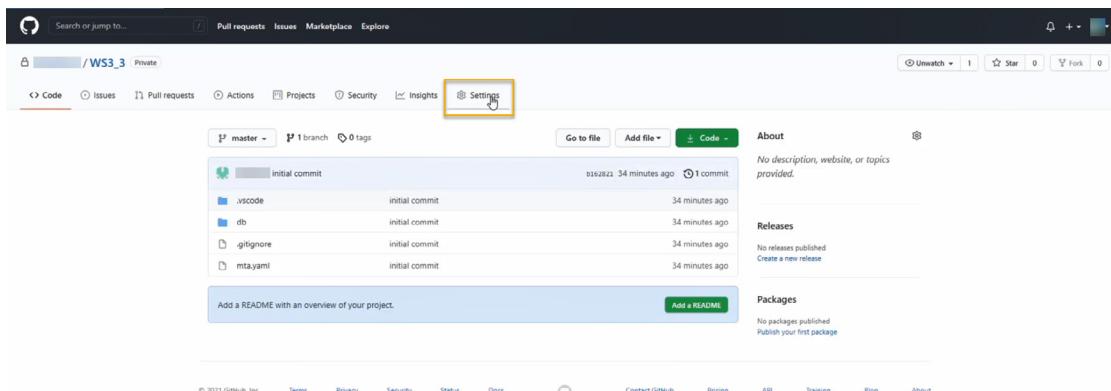
Steps for Project Template Owner:

Synchronize the project with git

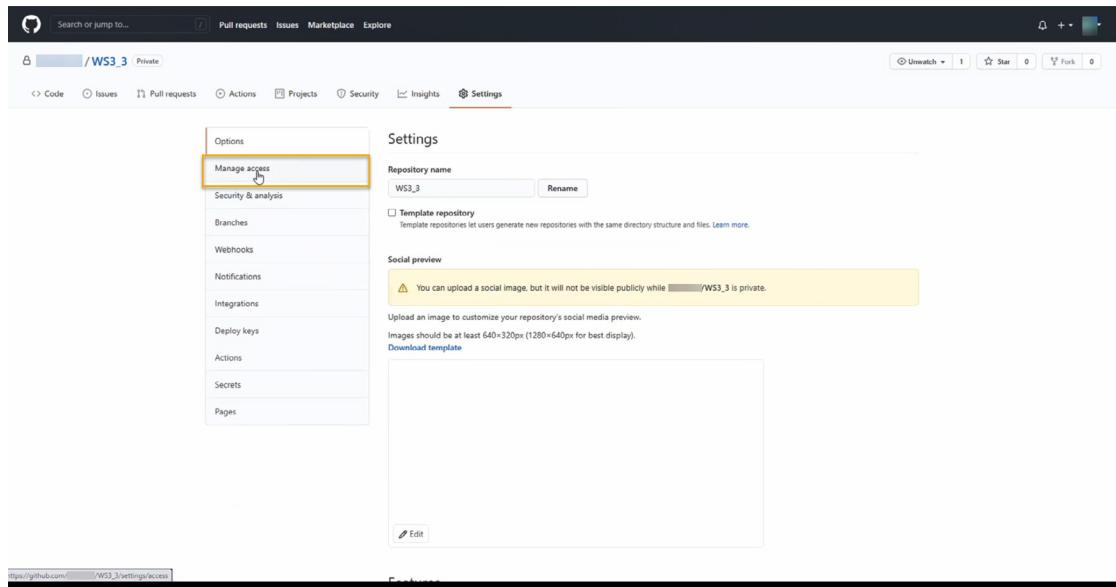
1. Go back to your project WS3_3 in SAP Business Application Studio.
2. On the left side menu of your SAP BAS environment, click on the  icon to go to the **Source Control** panel. Here, you will see all the recent updates to your project that have not been pushed to GitHub yet.
3. In the Source Control panel, you can see all the recent changes made. Click on the plus icon to stage these changes.
4. Enter a commit message in the field at the top (for example. template) and click on tick mark icon next to the field to commit the changes.
5. In the Source Control panel, click on the **three dots** icon. Select **Pull**, **Push** and then **Push to...** option.
6. A search prompt asking for a remote to publish the master branch appears on the horizontal menu bar. Select **main**.
7. You can verify the synchronization by checking your GitHub repository WS3_3. You should now see all the objects you created in the src folder there, for example, the synonyms and hdbgrants files.

Invite a collaborator to your git repository

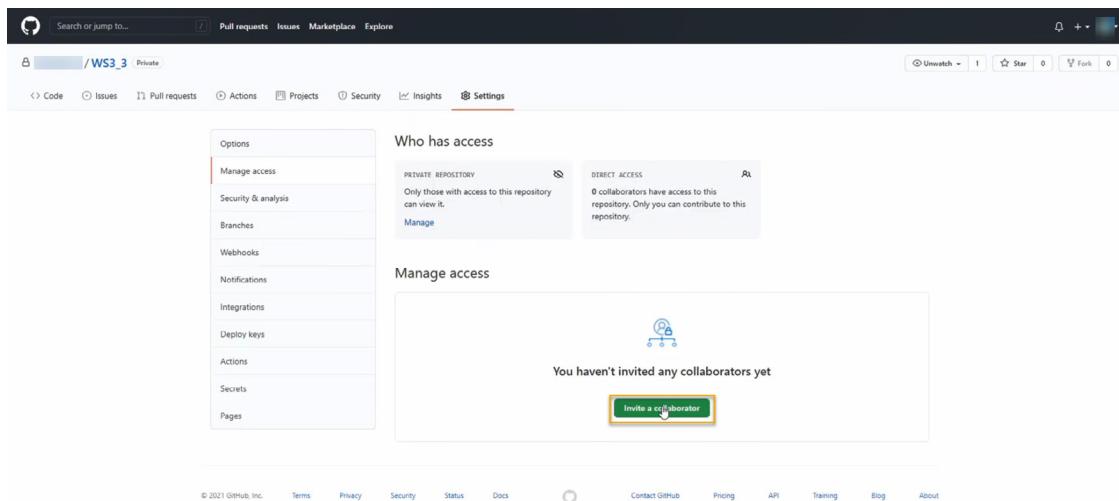
1. Go back to your GitHub browser tab and navigate to your repository **WS3_3**.
2. Select **Settings** option from the horizontal menu bar at the top.



3. Select **Manage access** from the panel on the left side of the screen.



4. You might need to sign in again with your GitHub credentials before you can continue.
 5. Click on **Invite a collaborator** in the Manage access section.



6. In the new dialogue box that appears, search for the GitHub username of your collaborator. Select the name and click on **Add <username> to this repository**.

Your collaborator will now receive an invitation to collaborate on your project. Ideally, we would recommend you to first fork your repository and only grant your collaborator the permission to work on the fork, thereby preventing any unwanted changes to your main project repository. The procedure used in this workshop is only for ease of illustration.

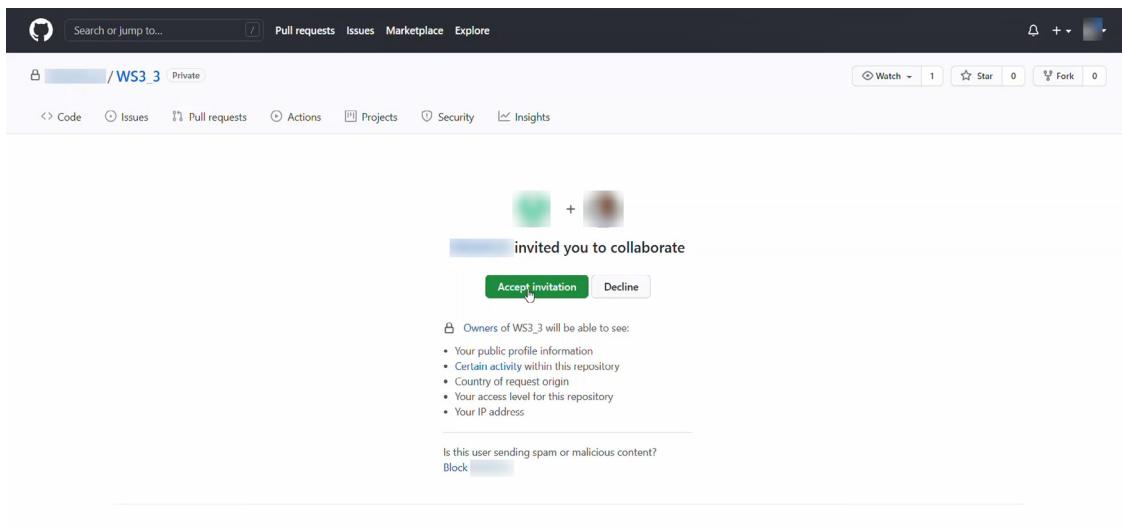
Steps for template User:

The steps in this section are an example of how a collaborator could use the template project to create a simple calculation view. You as the project template owner can simulate the process by using the same dev space and creating a new project by cloning from git. Some instructions in this section only apply to others using your template project. Simply skip the steps not needed in your case.

Clone the template project from git

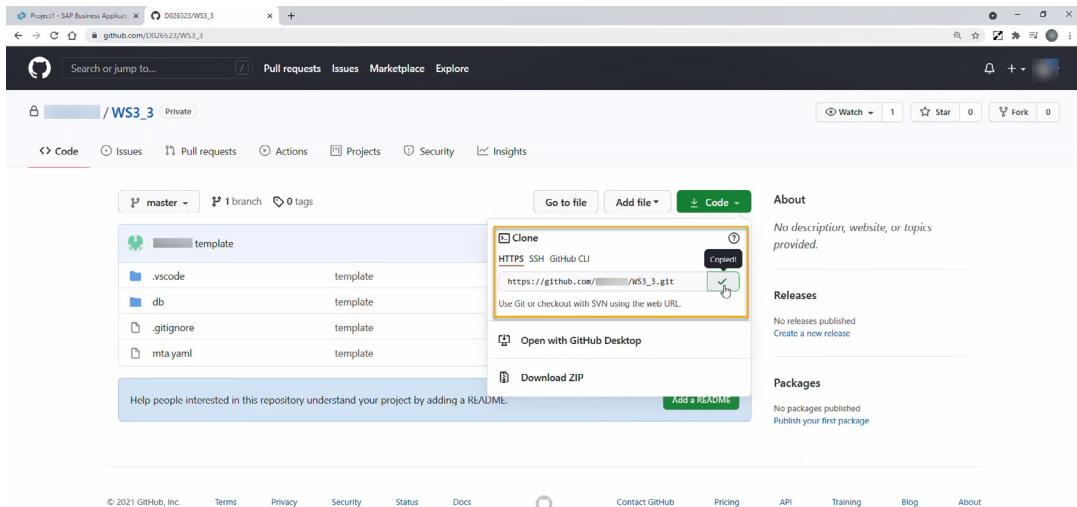
Note: If someone wants to use a template project by cloning it from the project repository, they have to enter the repository HTTPS URL, their username and their GitHub password or token.

- As a collaborator, you need to accept the invitation to work on the template project. To do this, go to your GitHub account and check the notifications for the invitation. Click on **Accept invitation**.



Now, you have access to the WS3_3 project folders in GitHub.

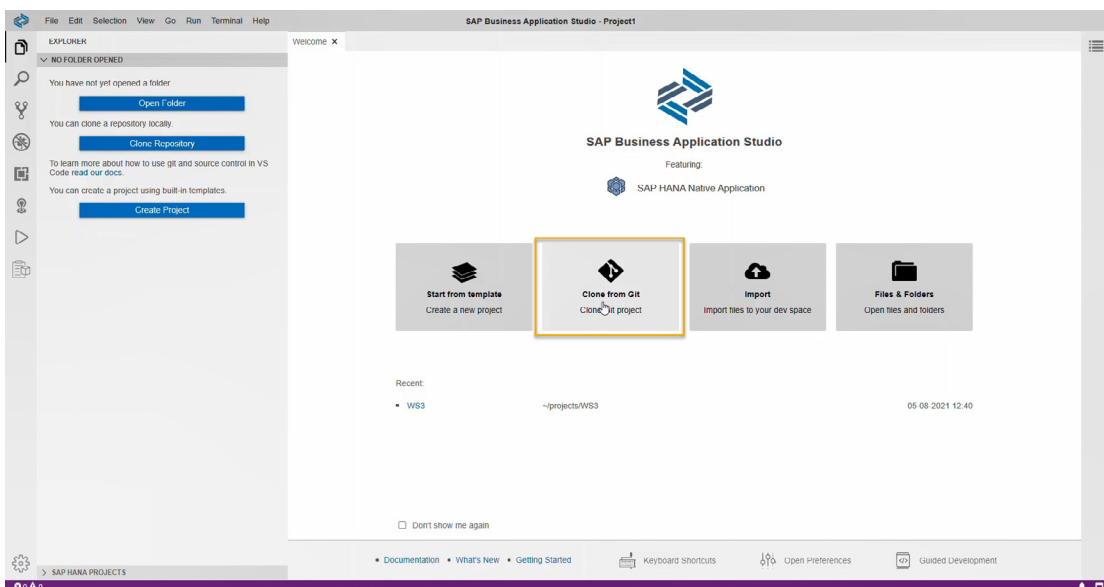
2. To clone from the project repository, you need the HTTPS connection link. In the repository, click on the green **Code** button at the top right and the clipboard icon next to the HTTPS URL.



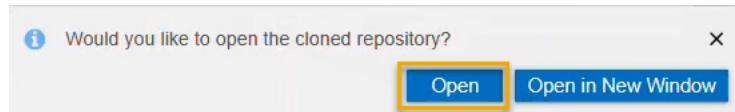
3. Go to your SAP Business Application Studio environment and open the **Welcome** page.

Note: To view the welcome window, click on **Help** at the top navigation bar and select **Welcome**.

4. Click on **Clone from Git**.



5. A prompt will open at the top center of the screen. Paste the HTTPS URL you just copied from GitHub into the dialogue box and press **Enter**.
6. Enter your **GitHub username** in the dialogue box and press **Enter**.
7. Next, enter your **Git account password** or a token. If you have already generated a token in GitHub, use that token here. Press **Enter**.
8. You will receive a notification at the bottom right corner of the screen asking how you want to open the cloned repository. Click **Open**.



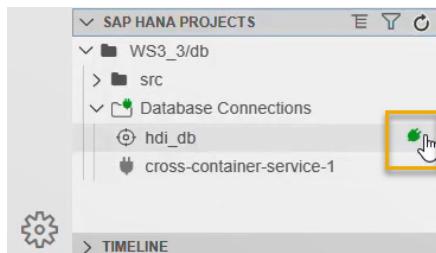
Now, you can see all the folders present in project WS3_3 from the GitHub repository in your file explorer in this new project.

Deploy the template project to an HDI container

If you are the project template owner, you cannot simulate this step in a cloned project in your development space and you can skip this step.

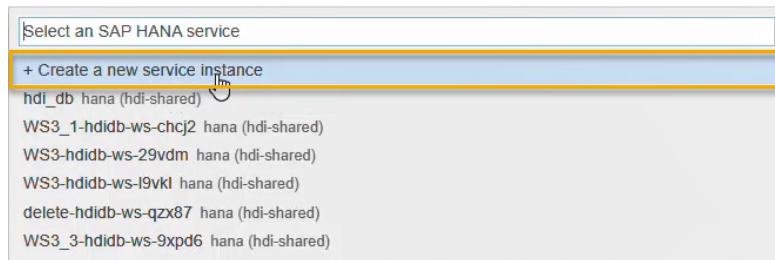
Next, you will create your own HDI container in the database containing the project and make use of the user provided service UPS_GENERIC created in the template project.

1. Click on the **bind icon** from the SAP HANA Projects panel.

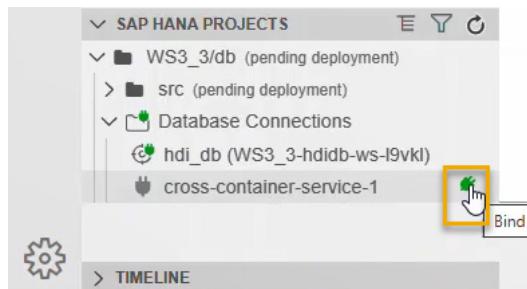


2. Press **Enter** to confirm your Cloud Foundry Endpoint in the dialogue box that appears on the horizontal menu bar.
3. Next, enter your SAP HANA Cloud account credentials and press **Enter**.
4. Select the Organization and the Space of the Project Owner. (Project Owner refers to the one who created the project and gave you access to the template.)

5. Next, select **Create a new service instance**.

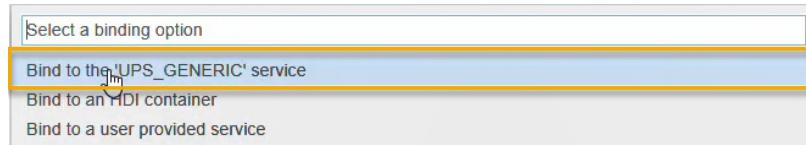


6. Press **Enter** to confirm the new service instance name provided by default.
7. Once the creation of the HDI container is complete, click on the **bind icon** for cross-container-service-1.



This will bind your project to the previously created UPS_GENERIC service.

8. Select **Bind to the 'UPS_GENERIC' service** as the binding option in the dialogue box.



9. Once the binding is complete, deploy the project to your HDI container. Click on the **deploy icon** for the project WS3_3 in the SAP HANA PROJECTS panel.

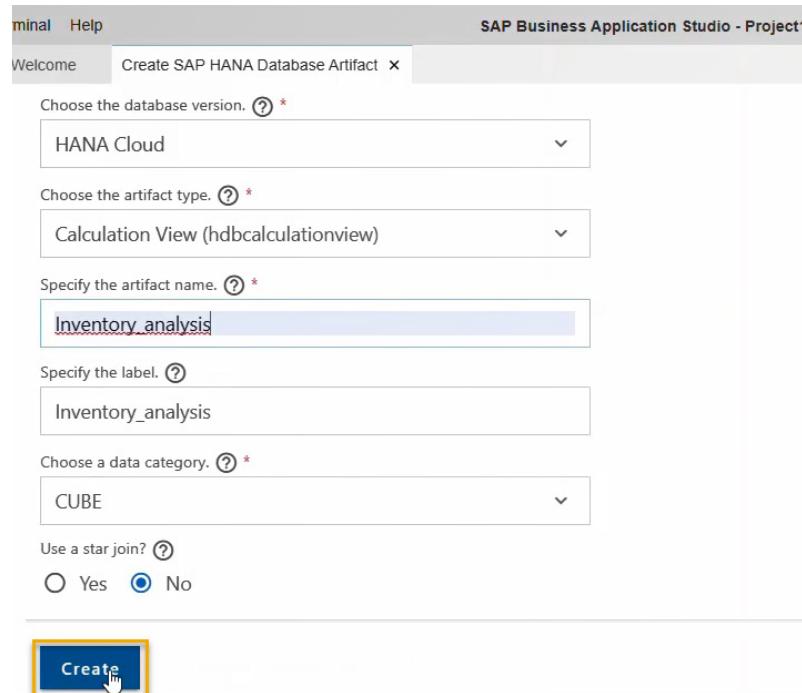
And with that, you are ready to use the template project and access the data. In the next sections, we will show you how to create a simple calculation view making use of the services, roles and synonyms in the project.

Create a simple calculation view

If you are the project owner, you can replicate the following sections regarding the calculation view in a cloned project in the same space or in the same project you have created the template project in.

Our goal is to create a calculation view on the inventory data that visualizes what products are low on stock. We will use a calculated column that indicates if a certain product stock is below its target value. To start, we first need to create a calculation view artifact in our project.

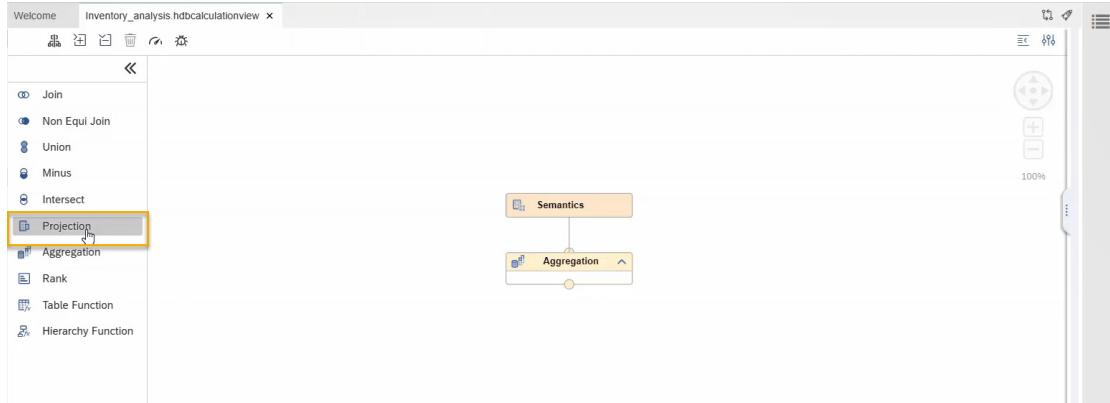
1. In the SAP Business Application Studio environment, click on **View – Find Command** at the top of the screen to open a search prompt. Enter **HANA** in the field and from the dropdown list of functionalities and choose **SAP HANA: Create SAP HANA Database Artifact**.
2. In the window for creating a new SAP HANA Database Artifact, click on the folder icon in the first field (Choose path where you want to create the artifact). Navigate to your project, the **src** folder, and then click on **Open**.
3. Select the artifact type as **Calculation View (hdbcalculationview)**.
4. In the field ‘Specify the artifact name’ option, enter the name **Inventory_analysis**.
5. In the field ‘Specify the label’ option, enter the name **Inventory_analysis**.
6. In the field ‘Choose a data category’ option, select **CUBE**.
7. Select the choice ‘**No**’ for the option ‘Use a start join?’.
8. The database version should be **HANA Cloud**, by default. Without making any further changes, click on **Create**.



Add a data source to the calculation view

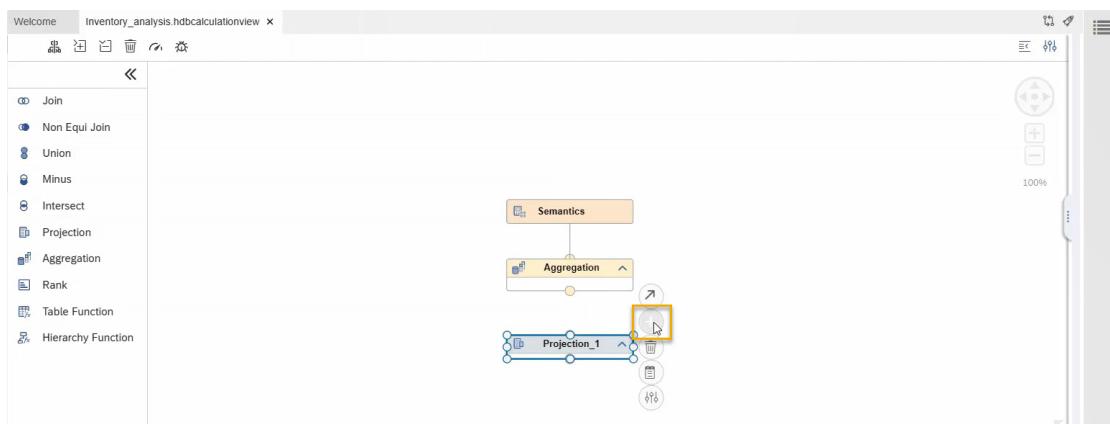
The first step in the calculation view graphical editor is to add a data source and define what columns should be mapped to the output. To do this, we will add a projection node to the view.

1. Select the new file **Inventory_analysis.hdbcalculationview** created in the src folder of your project.
2. Drag and drop a **Projection** node from the vertical tool bar into the calculation view window.

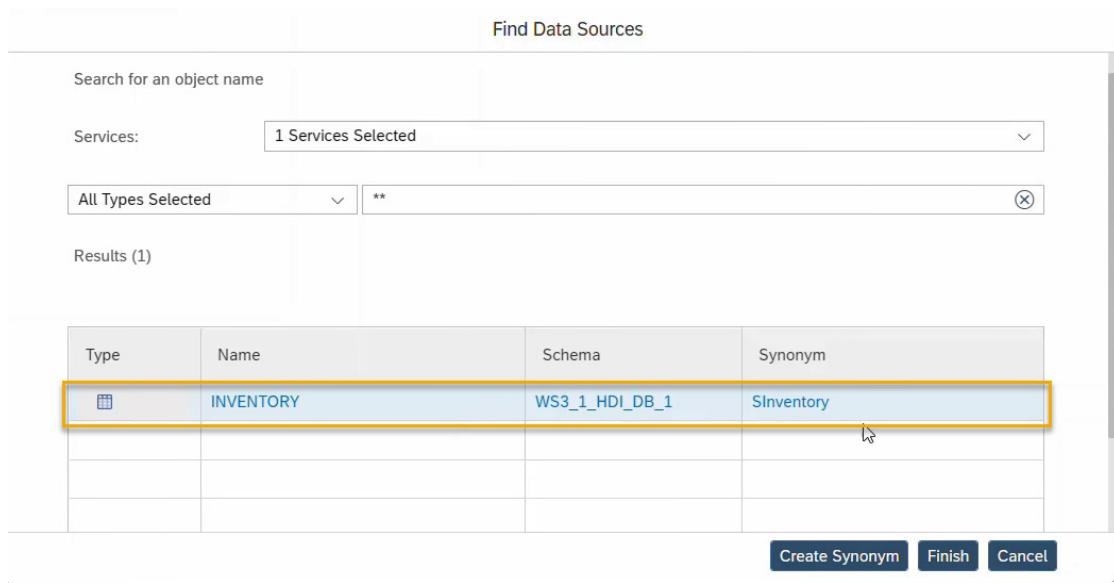


Note: To see the labels of different nodes available in the calculation view graphical editor, you can expand the side-bar menu by clicking on the arrows at the top left corner of the editor.

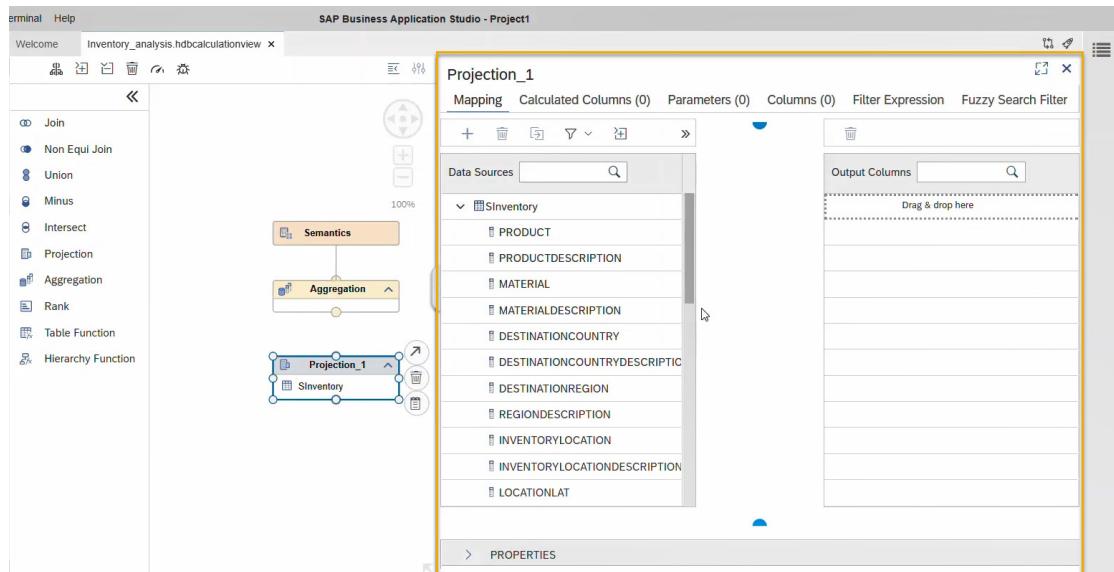
3. Click on the **plus icon** to Add Data Sources.



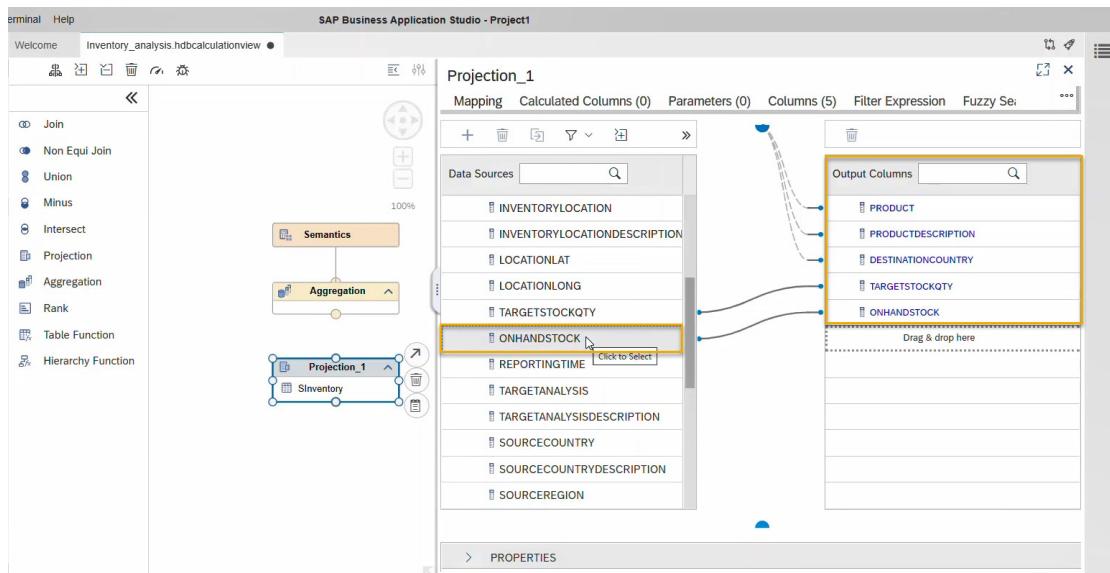
4. In the dialogue that opens, type ** in the search bar to view all the objects available. Select the synonym **SInventory** from the list and click on **Finish**.



5. Double-click on the **Projection** node to configure it.



6. In the tab **Mapping**, you can now select the **columns** from the Data Sources that you want in the Output Columns. Clicking on a column on the left will map it to the output on the right.

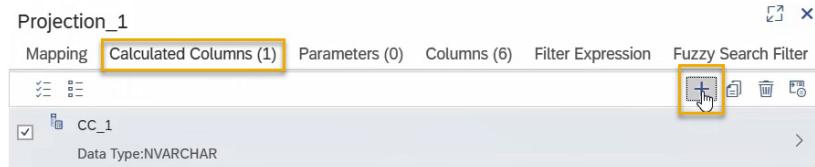


7. In this example, you select the columns PRODUCT, PRODUCTDESCRIPTION, DESTINATIONCOUNTRY, TARGETSTOCKQTY and ONHANDSTOCK.

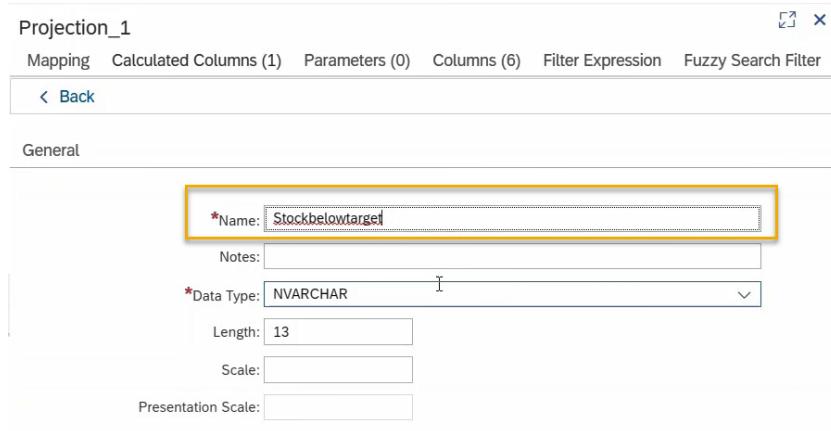
Add a calculated column to the projection node

To calculate what products' stock values are below a target stock value, we will next add a calculated column to the projection node.

1. In the projection node menu, select the **Calculated Columns** tab.
2. Click on the **plus icon** at the top right to add a calculated column.



3. Once the calculated column is generated, click on the arrow (>) icon to take you the General properties.
4. You can now rename the calculated column as "**Stockbelowtarget**".
5. Select the Data Type as **INTEGER**.



6. Expand the **Expression** section below the **General** settings.

7. Click on Expression Editor.

The screenshot shows the SAP Business Application Studio interface. A window titled 'Projection_1' is open, specifically the 'Calculated Columns (1)' tab. On the left, there's a tree view with 'Elements' selected, showing columns like PRODUCT, PRODUCTDESCRIPTION, DESTINATIONCOUNTRY, TARGETSTOCKQTY, and ONHANDSTOCK. The main area is the 'Expression Editor' for a calculated column named 'Stockbelowtarget'. The 'Expression' tab is active, and the expression 'TARGETSTOCKQTY > ONHANDSTOCK' is typed into the input field. A yellow box highlights the 'Expression Editor' tab in the top navigation bar.

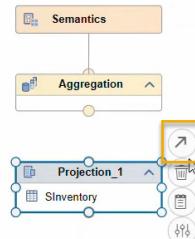
8. From the Elements section on the left, select the column **TARGETSTOCKQTY** with a double-click. Then enter ">" and select the column **ONHANDSTOCK** from the left.
9. The final Expression should look like this. You can also copy and paste it to the expression editor from here:

"TARGETSTOCKQTY" > "ONHANDSTOCK"

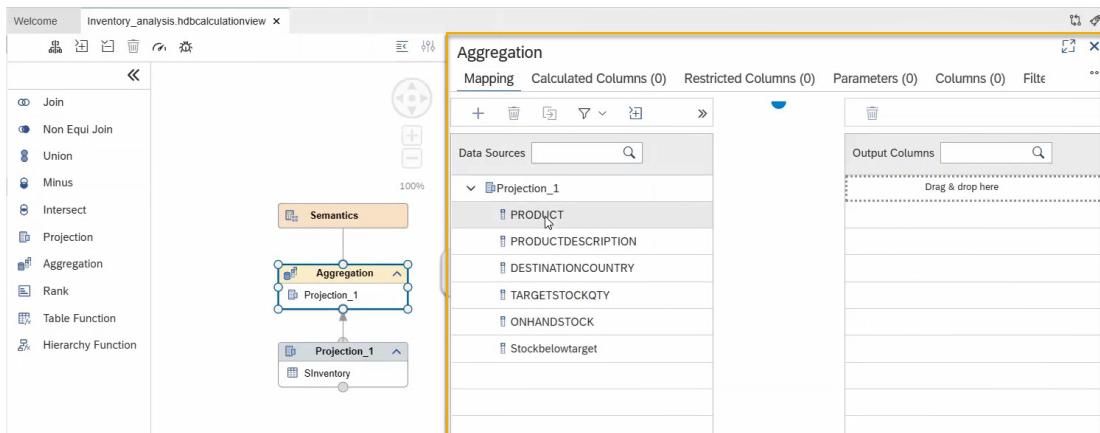
The screenshot shows the SAP Business Application Studio interface. A window titled 'Projection_1' is open, specifically the 'Calculated Columns (1)' tab. On the left, there's a tree view with 'Elements' selected, showing columns like PRODUCT, PRODUCTDESCRIPTION, DESTINATIONCOUNTRY, TARGETSTOCKQTY, and ONHANDSTOCK. The main area is the 'Expression Editor' for a calculated column named 'Stockbelowtarget'. The 'Expression' tab is active, and the expression 'TARGETSTOCKQTY > ONHANDSTOCK' is typed into the input field. A yellow box highlights the expression in the input field.

10. Close the Projection node window.

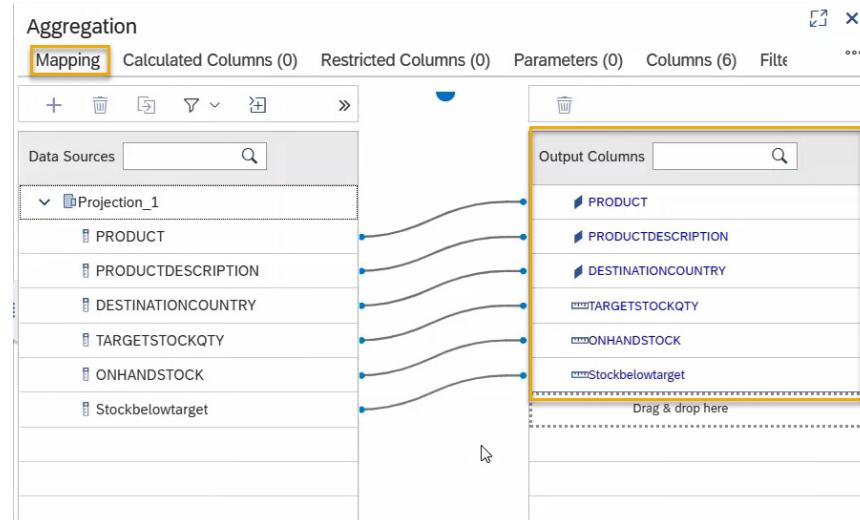
11. Connect the Projection node with Aggregation node by dragging the **arrow icon** from the Projection node to the Aggregation node above.



12. Double-click on the **Aggregation** node to configure it.



13. In the tab **Mapping**, select the columns from the Data Sources that want in the Output Columns. Map all the columns present in the Presentation node to Output Columns. Double-clicking on the top level on the left "Projection_1" will map all columns to the output.

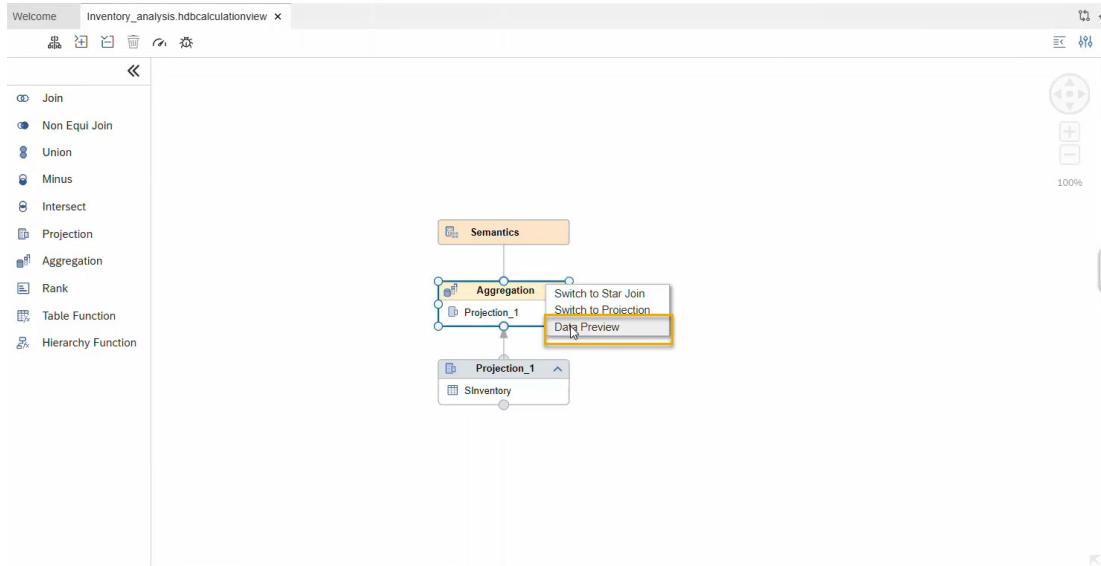


14. Close the Aggregation node window.

And with that, the view is complete. Lastly, we will deploy it to the database and preview the results.

Deploy the view and preview the data

1. Click on the **deploy** icon on the top right corner of the Calculation View editor to deploy it to your database.
2. Right-click on the **Aggregation** node and select **Data Preview** to analyze the data.

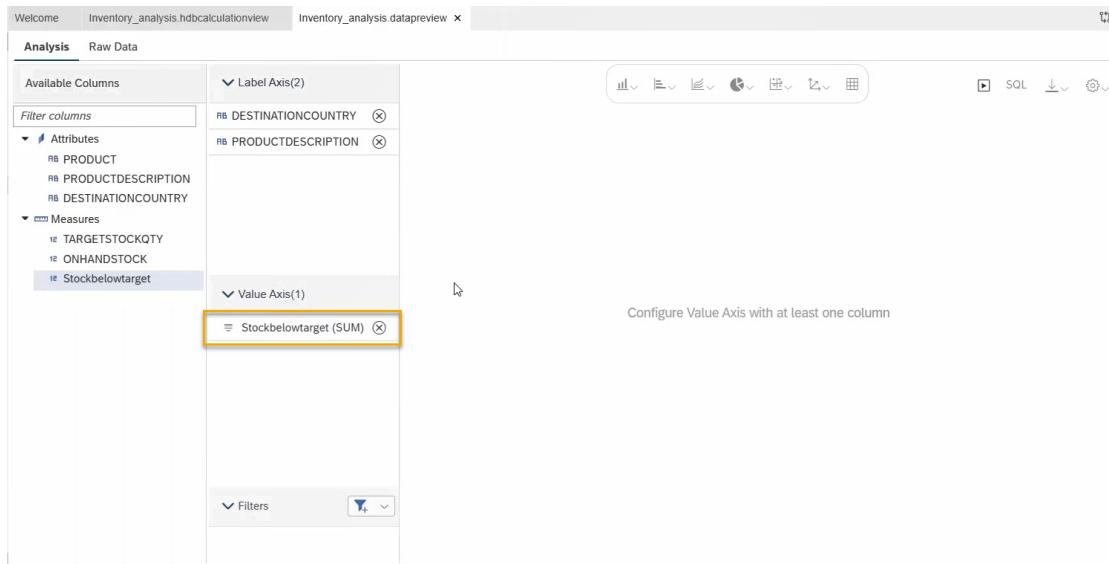


Note: You may also open the view and preview the data in the SAP HANA Database Explorer.

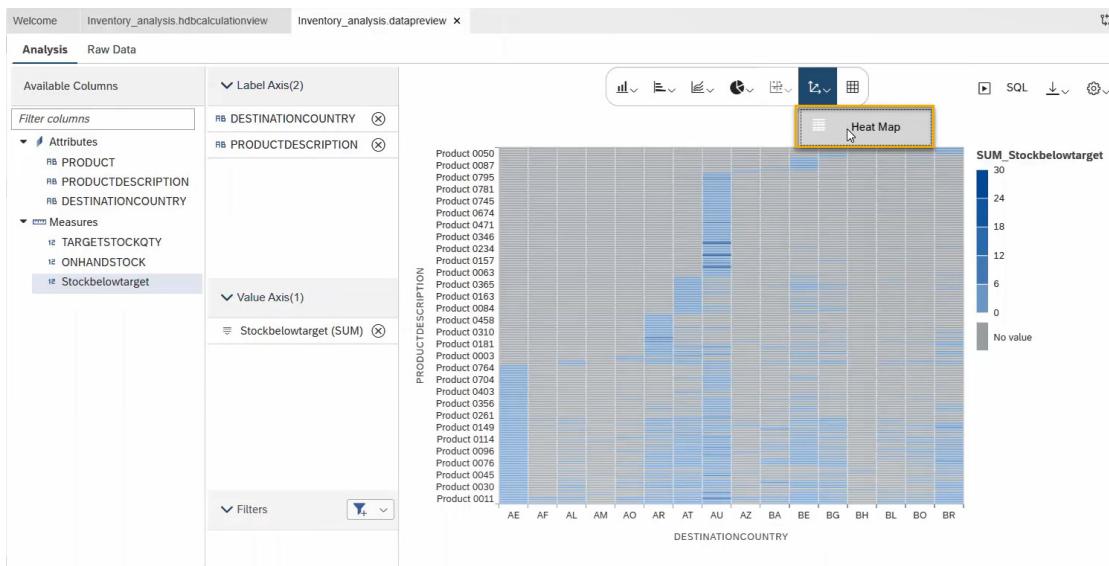
3. In the Data Preview window of the Calculation View, select the **Analysis** tab.
4. From the list of Attributes on the left, drag and drop the columns **DESTINATIONCOUNTRY** and **PRODUCTDESCRIPTION** to the **Label Axis**.

The screenshot shows the 'Inventory_analysis.datapreview' window. The 'Analysis' tab is selected. On the left, under 'Available Columns', 'DESTINATIONCOUNTRY' and 'PRODUCTDESCRIPTION' are selected and dragged to the 'Label Axis(2)' section. The 'Value Axis(0)' section is visible below, with a message stating 'Configure Value Axis with at least one column'. At the bottom, there is a 'Filters' section. The top of the window shows tabs for 'Welcome', 'Inventory_analysis.hdbculationview', and 'Inventory_analysis.datapreview'.

5. From the list of Measures, drag and drop the calculated column **Stockbelowtarget** to the Value Axis.



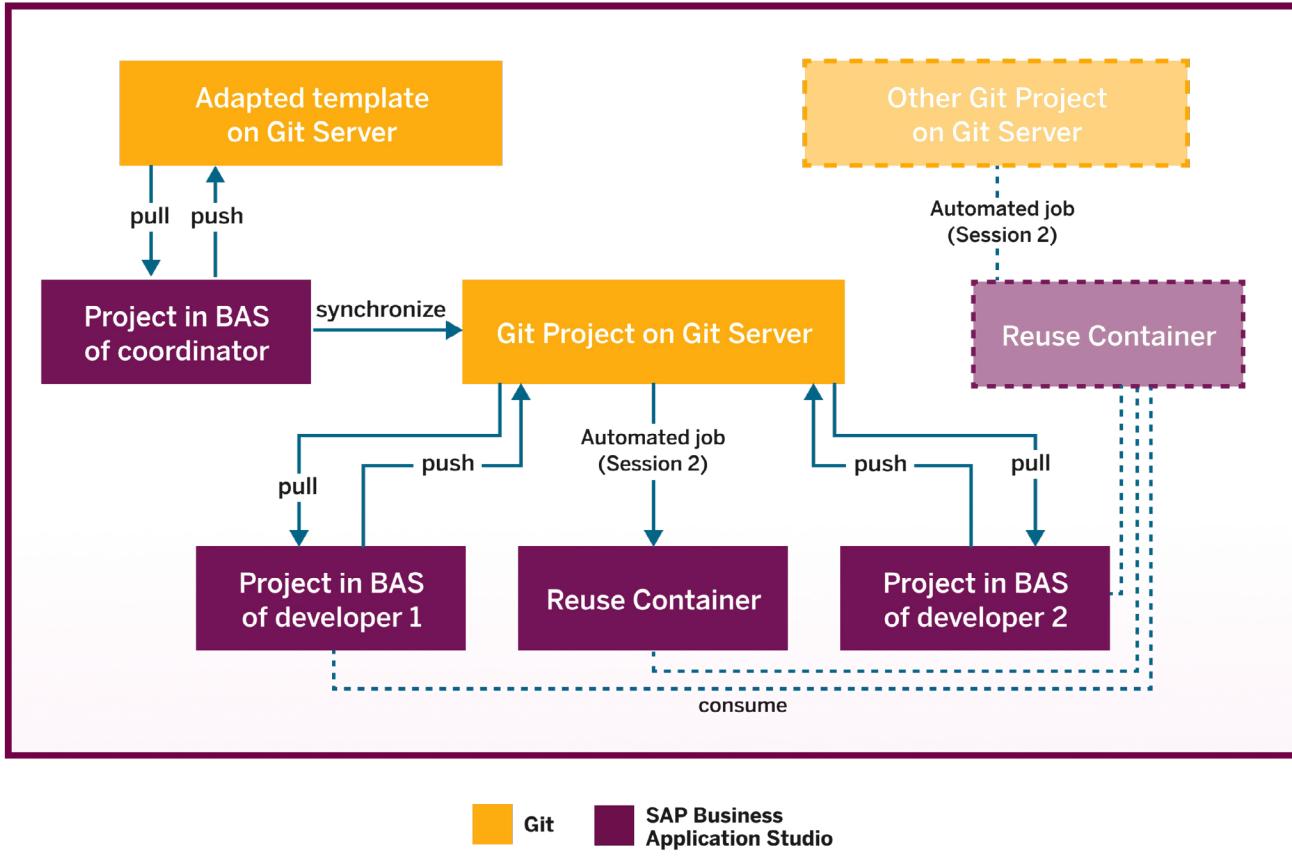
6. Using the menu bar for graphs, select the option for **Heat Map** to view what products have stock values below their target value in different countries.



And this concludes the data analysis in the calculation view.

This example shows how the template project created by the project owner can be used by a collaborator to create a calculation view and derive insights from the data in their own HDI container while easily accessing the data and necessary database objects. The advantage of using the template is that the collaborator can have access to the available data without managing roles and privileges.

SUMMARY



We started this session by creating a template project in SAP Business Application Studio and synchronizing it with the Git project.

This allows other developers in the team to work on their own project copies and use the objects and data in the project. The template creator can set up the project and its services and privileges once in the beginning making the development process more efficient for multiple collaborating developers.

In addition to the template project shown, reuse containers can further help make sure each developer may use their own isolated environment for minimal disruption. A reuse container acts as an interface for developers to safely share the objects they create without a concern on its future alterations.

Excellent! You've completed Session 3 of the Digital Hands-on Workshop!

We hope you learned a lot in this workshop and had as much fun following along as we had in creating it!

To stay up-to-date about upcoming events, follow the tag [SAP HANA Cloud](#) in the [SAP Community](#).

If you have any issues with the steps described in this workbook, please ask your questions there and we will get back to you as soon as we can.

See you at our next workshop and enjoy your trial of SAP HANA Cloud!

Additional Resources

- [SAP HANA Cloud product page](#)
- [Upcoming SAP HANA Cloud events](#)
- [Explore more SAP HANA Cloud tutorials](#)
 - [Mission: Jump start your SAP HANA Cloud, SAP HANA Database Trial](#)
 - [Mission: Get Started with a Standalone SAP HANA Cloud, Data Lake](#)
 - [Mission: Extend Your SAP HANA On-Premise to SAP HANA Cloud, SAP HANA Database](#)
- Technical Documentation
 - [SAP HANA Cloud](#)
 - [SAP Business Application Studio](#)
 - [SAP Continuous Integration and Delivery](#)