

## Contents

What you will learn .....	1
Prerequisites .....	1
Simulator Setup.....	2
Simulator Test.....	5
Takeaway .....	13

## What you will learn: Simulator

Day 1 of 5. Topics covered :

SAP business objects simulator

- Publish SAP simulated events that represents mocked JSON structures
- Publish events from 5 different SAP object- Salesorders, Business Partners, Chart of Accounts, Material Master, and Notifications
- Test Simulated SAP events being published and subscribed too using the AEM broker and the SAP Ui5 dashboard.

## Prerequisites

### **1: BTP subaccount with a developer space enabled in Cloud Foundry**

### **2: CloudFoundry CLI installation (Only needed if upload doesn't work)**

To start with, we will be installing the CloudFoundry CLI for the deployment process. Follow the steps mentioned over here [Installing the CF CLI](#) for detailed instructions on this.

### **3 : Downloading the deployable artifacts**

Download the following files artefact files and save them in the same directory:

- capm-erp-simulation-exec.jar : <https://github.com/SolaceLabs/aem-sap-integration/blob/main/deployable/capm-erp-simulation-exec.jar>

- manifest.yml : <https://github.com/SolaceLabs/aem-sap-integration/blob/main/deployable/manifest.yml>

## SAP Simulator setup

The SAP Cloud Application Programming Model (CAP) is a framework of languages, libraries, and tools for building enterprise-grade services and applications. It guides developers along a 'golden path' of proven best practices and a great wealth of out-of-the-box solutions to recurring tasks. CAP-based projects benefit from a primary focus on domain. Instead of delving into overly technical disciplines, we focus on accelerated development and safeguarding investments in a world of rapidly changing cloud technologies.

For more information on SAP CAP, you can refer to the link : [SAP Cloud Application Programming Model](#)

To showcase the integration capability of SAP CAP and AEM, we have created a CAP based Java microservice which will publish different SAP business object events into your AEM instance. This application can be deployed in your SAP CloudFoundry space.

### 1 : Identify CF Domain address

In order to deploy the simulator to your CloudFoundry space, you need to identify the domain address which is a part of the API endpoint.

- Navigate to your SAP BTP Sub account Overview page
- Copy the specified section of the API Endpoint in the Cloud Foundry Environment as shown below :

The screenshot shows the SAP BTP Cockpit interface. On the left, there's a navigation sidebar with various options like Overview, Services, Cloud Foundry, HTML Applications, Connectivity, Security, Entitlements, and Usage Analytics. A red circle labeled '1' is around the 'Cloud Foundry' option. The main content area is titled 'Subaccount: trial - Overview'. It has tabs for General, Cloud Foundry Environment, Kyma Environment, and Entitlements. Under General, it shows 75 Entitlements and 2 Instances and Subscriptions. It also displays Subdomain: fe049ff2trial, Tenant ID: [REDACTED], Subaccount ID: [REDACTED], Provider: Amazon Web Services (AWS), Region: US East (VA), and Environment: Multi-Environment. Below this, under 'Cloud Foundry Environment', there's a table with one row for 'Spaces (1)'. The 'Name' column has 'dev', the 'Applications' column has '1', and the 'Service Instances' column has '0'. A red circle labeled '2' is around the 'API Endpoint' field, which contains the URL https://api.cf.us10-001.hana.ondemand.com. There are also buttons for 'Create Space', 'Manage environment instance', and 'Disable Cloud Foundry'.

## 2 : Update the manifest file

- Open the manifest.yml file which you downloaded earlier in a text editor
- Replace the placeholder text {API\_ENDPOINT} on line number 12 with the value copied from the API Endpoint
- Also in the route name "capm-erp-simulation-aem-workshop.cfapps.{API\_ENDPOINT}", change the name workshop to your company name
- Example capm-erp-simulation-aem-solace.cfapps.{API\_ENDPOINT},

```
1  # Generated manifest.yml based on template version 0.1.0
2  # appName = capm-erp-simulation
3  # language=java
4  # multitenancy=false
5  ---
6  applications:
7  # -----
8  # Backend Service
9  # -----
10 - name: capm-erp-simulation
11   routes:
12     - route: capm-erp-simulation-aem-workshop.cfapps.{API_ENDPOINT}
13       path: srv/target/capm-erp-simulation-exec.jar
14       memory: 1G
15       disk_quota: 512M
16   env:
17     JBP_CONFIG_SPRING_AUTO_RECONFIGURATION: '{ enabled: false }'
18   buildpack: sap_java_buildpack
19 # random-route: true
20
21
```

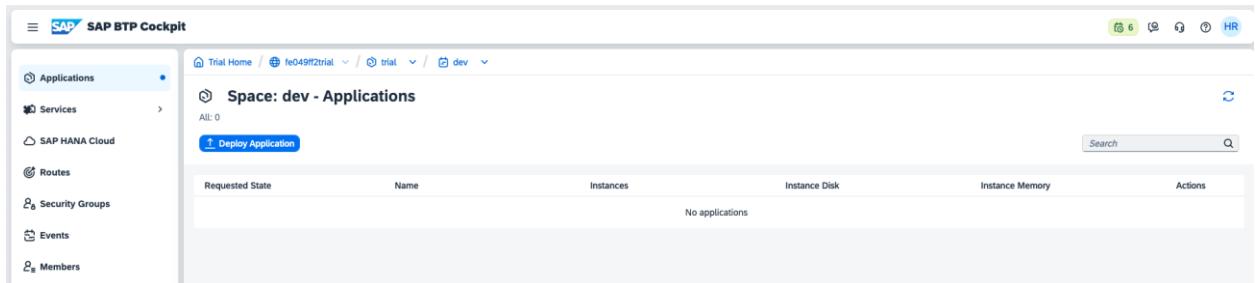
- After replacing your manifest file should look like this :

```
1  # Generated manifest.yml based on template version 0.1.0
2  # appName = capm-erp-simulation
3  # language=java
4  # multitenancy=false
5  ---
6  applications:
7  # -----
8  # Backend Service
9  # -----
10 - name: capm-erp-simulation
11   routes:
12     - route: capm-erp-simulation-aem-workshop.cfapps.us10-001.hana.ondemand.com
13       path: srv/target/capm-erp-simulation-exec.jar
14       memory: 1G
15       disk_quota: 512M
16       env:
17         JBP_CONFIG_SPRING_AUTO_RECONFIGURATION: '{ enabled: false }'
18       buildpack: sap_java_buildpack
19   # random-route: true
20
21
```

Save and close the file.

### 3 : Deploying the SAP Simulator application

- Navigate to the CloudFoundry space where you want to deploy the application and click on the **Deploy Application** button as below :



The screenshot shows the SAP BTP Cockpit interface. The left sidebar has a tree view with 'Applications' selected. The main content area is titled 'Space: dev - Applications' and shows a table with one row: 'No applications'. At the top right of the main area, there is a blue button labeled 'Deploy Application'.

- A **Deploy Application** modal window will be displayed where you can browse and select the **capm-erp-simulation-exec.jar** and **manifest.yml** as below

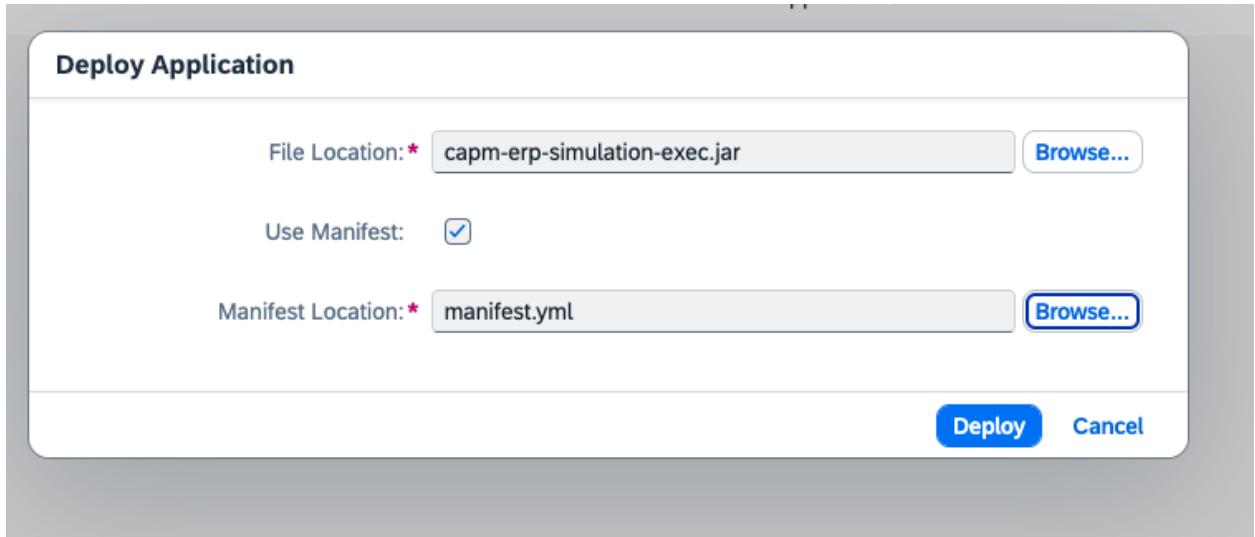
**Deploy Application**

File Location: \*  [Browse...](#)

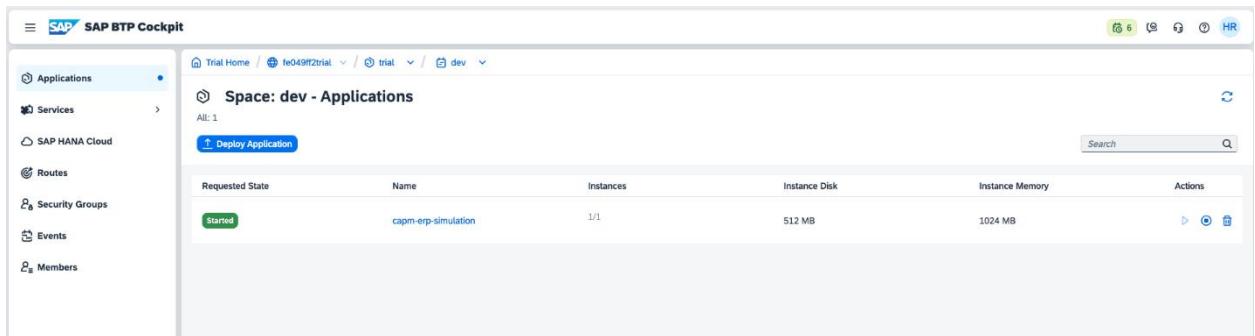
Use Manifest:

Manifest Location: \*  [Browse...](#)

[Deploy](#) [Cancel](#)



- Click on the deploy button **Note : this action will take some time to completely execute as it uploads the jar deployable and also start the application.**
- Once the upload is completely executed, you should be able to see the application deployed and running as below :



Requested State	Name	Instances	Instance Disk	Instance Memory	Actions
Started	capm-erp-simulation	1/1	512 MB	1024 MB	<a href="#">D</a> <a href="#">C</a> <a href="#">E</a>

## \*CMD Line Deployment Option\*

### Login to CloudFoundry space

You can log in to the SAP CloudFoundry space in your account as below :

Use the command : cf login to log in, which will prompt for your SAP login credentials.

Once authenticated, the details of the default cloudfoundry space will be displayed.

### Deploying the SAP Simulator application

Navigate to the directory where the above deployable artifact files are saved.

Run the command cf push --random-route which will upload the jar file and use the manifest.yml for properties. Note : this command will take some time to completely execute as it uploads the jar deployable and also start the application.

Once the command is completely executed, run the command cf apps to view a listing of the apps in your cloudfoundry space

Verify that the app capm-erp-simulation is deployed and started

## Testing the Simulator

### 1 : Accessing the SAP Simulator application

- Navigate to the Cloud Foundry environment in your SAP BTP Cockpit
- You should see a screen like below

The screenshot shows the SAP BTP Cockpit interface for the 'hari\_cf\_space' space. The 'Applications' section displays a single application named 'capm-erp-simulation'. The application details are as follows:

Requested State	Name	Instances	Instance Disk	Instance Memory	Actions
Started	capm-erp-simulation	1/1	512 MB	1024 MB	

- Click on the application name : **capm-erp-simulation** and enter the application overview screen.

The screenshot shows the 'Application Overview' page for the 'capm-erp-simulation' application. The page includes sections for Application Routes, Application Information, Instance Details, and Instances.

**Application Routes:** https://capm-erp-simulation.cfapps.us10-001.hana.ondemand.com

**Application Information:**

Instances	Package Uploaded	Buildpack	Stack
1	27 Sept 2023, 13:11:25 (GMT+02:00) (STAGED)	sap_java_buildpack	Cloud Foundry Linux-based filesystem (Ubuntu 22.04) (cflinuxfs4)

**Instance Details:**

Instance Memory	Instance Disk
1024 MB (available memory 3072 MB)	512 MB

**Instances:**

#	State	Since	CPU	Memory	Disk
0	RUNNING	29 Sept 2023, 15:48:21 (GMT+02:00)	0.6%	181.2 MB	474.5 MB

- Click on the application route as highlighted below. Note : this route url will differ from for different SAP BTP accounts.

## Rapid Pilot Day 1

The screenshot shows the SAP Cloud Foundry Application Overview page for the application 'capm-erp-simulation'. At the top, there's a toolbar with buttons for 'Restart', 'Start', 'Stop', 'Instance' (with '+' and '-' icons), 'Delete', and a refresh icon. Below the toolbar, the 'Application Routes' section displays the URL <https://capm-erp-simulation.cfapps.us10-001.hana.ondemand.com>. The 'Application Information' section provides details about the instance: Instances: 1, Package Uploaded: 27 Sept 2023, 13:11:25 (GMT+02:00) (STAGED), Buildpack: sap\_java\_buildpack, Stack: Cloud Foundry Linux-based filesystem (Ubuntu 22.04) (cflinuxfs4). The 'Instance Details' section shows Instance Memory: 1024 MB (available memory 3072 MB) and Instance Disk: 512 MB. The 'Instances' section lists one instance: # 0, State RUNNING, Since 29 Sept 2023, 15:48:21 (GMT+02:00), CPU 0.6%, Memory 181.2 MB (green bar), and Disk 474.5 MB (orange bar).

## 2 : Connecting to SAP AEM and running the simulator

- As you click on the above application route url, you will be redirected to the simulator screen as below

The screenshot shows the SAP Advanced Event Mesh - ERP Simulator connection interface. It features a header with the SAP logo and the text 'Welcome to the Advanced Event Mesh - ERP Simulator'. Below the header is a form with fields for 'Host URL', 'VPN Name', 'Username', and 'Password', each with a red 'Required' validation icon. A 'Connect to broker' button is at the bottom of the form.

Here you can connect to your SAP AEM instance to publish events.

As long as both of your SAP AEM services are connected to the event mesh, messages will flow freely between the two of them. Due to this intelligent routing, you can connect the simulator to either of your AEM services created earlier.

- The connection parameters for the simulator can be captured from below:

The screenshot shows the SAP MontrealBroker-10.1 monitoring interface. The top navigation bar includes 'MontrealBroker-10.1', 'Status' (highlighted with a red box), 'Connect', 'Manage', 'Monitoring', 'Configuration', and 'Try Me!'. On the left, there's a sidebar with 'Mission Control' (selected), 'Event Portal' (with a 'Beta' badge), and other icons. The main area is titled 'Messaging Activity' and displays three circular progress bars: 'Active Connections' (20 %), 'Guaranteed Messaging Endpoints' (53 %), and 'Queue Usage' (1 %). Below these are tables for 'AMQP' (0), 'MQTT' (0), 'SMF' (20), 'REST' (0), 'Web' (0), 'Queues' (52), 'Topic Endpoints' (1), and 'Messages Queued' (161721). The bottom section is titled 'Availability and Versioning'.

## Rapid Pilot Day 1

The screenshot shows the MontrealBroker-10.1 interface. At the top, there's a navigation bar with links for Status, Connect, Manage, Monitoring, Configuration, Try Me!, Open Broker Manager, and more. A red circle labeled '1' highlights the 'View by:' dropdown menu, which is set to 'Language'. Below it, a link 'Expand all' is visible.

The main area is titled 'Connect Using a Supported Client Library' and instructs the user to select a connection point and supported client library to start messaging. A red circle labeled '2' highlights the 'Connect with Spring' option, which includes 'Spring Cloud Stream, Spring Boot'.

A table below lists supported libraries and protocols:

Library	Protocol
Spring Cloud Stream	SMF
Spring Boot Java API	SMF

Red circles labeled '3' and '4' highlight the 'Spring Boot Java API' row and the 'Get Started' button respectively.

Enter the appropriate value as specified below :

- Host URL : Public Endpoint
- VPN Name : Message VPN
- Username : Username
- Password : Password

## Rapid Pilot Day 1

Select another API

**Spring**

[API Docs](#) [Download](#)

### Spring Boot Java API

**1 Get API** **2 Connect to Service** **3 Learn More**

Here are a few easy ways to get the API. If your environment differs then adjust the build instructions appropriately.

**Get the API: I already have a Maven Spring Boot Project**

Add the Solace Java Spring Boot Starter to your POM if using Maven.

```
<dependency>
  <groupId>com.solace.spring.boot</groupId>
  <artifactId>solace-java-spring-boot-starter</artifactId>
  <version>4.1.0</version>
</dependency>
```

**Get the API: I already have a Gradle Spring Boot Project**

Add the Solace Java Spring Boot Starter to your build.gradle if using Gradle.

```
compile group: 'com.solace.spring.boot', name: 'solace-java-spring-boot-starter'
```

**Get the API: Starting from Scratch - Use Spring Initializr to bootstrap my project!**

It is recommended to bootstrap your Spring Boot project using [Spring Initializr](#). Spring Initializr will generate your Spring Boot project skeleton for you based on a few simple choices, such as your preferred language, Spring Boot version, and then dependencies you need to build your microservice.

Once you bootstrap your Spring Boot project add the Solace Java Spring Boot Starter as shown in the previous section.

**Connection Details**

Username  Password  Message VPN

**Host URIs**

Secured SMF URI  Public Internet:   
TrustStore [Download PEM](#)

- Once the broker is successfully connected, you will be displayed a screen as below :

**Welcome to the Advanced Event Mesh - ERP Simulator**

**Broker connection status**

**SAP Event simulator schedules**

tcp://  solace-cloud-client  Password   
**Connect to broker**

**Success! Broker connected successfully**

Sales Order Create event frequency:  0 minute  
Sales Order Change event frequency:  0 minute  
Business Partner Create event frequency:  0 minute  
Business Partner Change event frequency:  0 minute  
Material Master Create event frequency:  0 minute  
Material Master Change event frequency:  0 minute  
Chart of Accounts Create event frequency:  0 minute  
Chart of Accounts Change event frequency:  0 minute  
Notification Create event frequency:  0 minute  
Notification Change event frequency:  0 minute

**Submit**

- You can choose which events to simulate and its frequency by using the sliders. As you change a schedule, the submit button in the bottom will be enabled.

Welcome to the Advanced Event Mesh - ERP Simulator

SAP

tcp://montrealbroker.messaging.solace.com  
montrealbroker-10-1  
solace-cloud-client  
Password

**Connect to broker**

Success! Broker connected successfully

Event Type	Event Frequency
Sales Order Create	10 minute
Sales Order Change	6 minute
Business Partner Create	15 minute
Business Partner Change	4 minute
Material Master Create	1 minute
Material Master Change	0 minute
Chart of Accounts Create	0 minute
Chart of Accounts Change	0 minute
Notification Create	0 minute
Notification Change	0 minute

**Submit**

- In case you want to disable any of the events, then pull the slider to **0** and click submit and the event will be disabled immediately.

Welcome to the Advanced Event Mesh - ERP Simulator

SAP

tcp://montrealbroker.messaging.solace.com  
montrealbroker-10-1  
solace-cloud-client  
Password

**Connect to broker**

Success! Broker connected successfully

Event Type	Event Frequency
Sales Order Create	0 minute
Sales Order Change	6 minute
Business Partner Create	15 minute
Business Partner Change	4 minute
Material Master Create	1 minute
Material Master Change	0 minute
Chart of Accounts Create	0 minute
Chart of Accounts Change	0 minute
Notification Create	0 minute
Notification Change	0 minute

**Submit**

### 3 : Test the incoming events

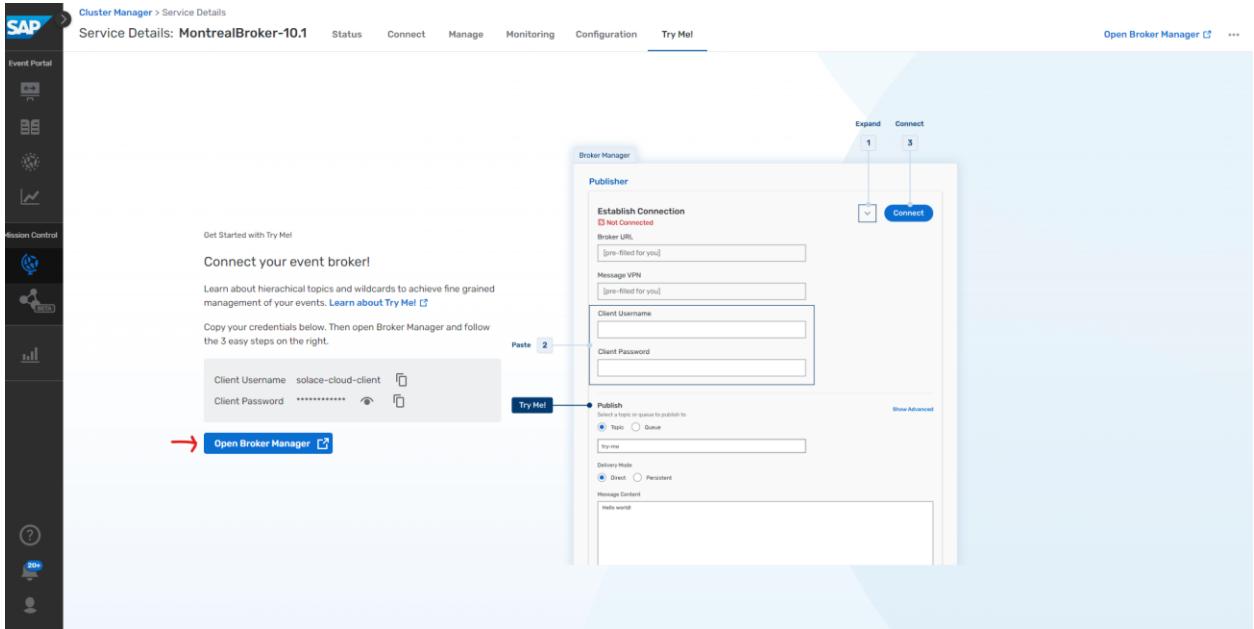
You can easily test the simulator by using the **Cluster Manager - Try-Me** as below:

- > aside negative
- > As mentioned earlier due to the intelligent routing in the event mesh, you can connect the simulator and try-me to either of the two SAP AEM services in the event mesh and see the messages flowing freely.

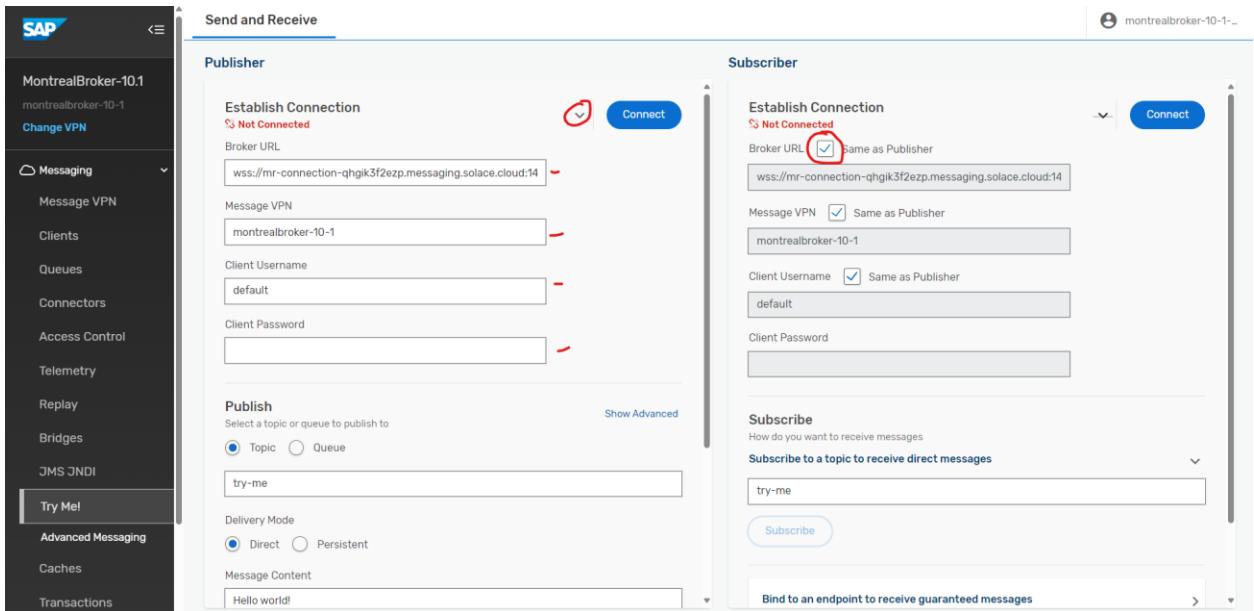
## Rapid Pilot Day 1

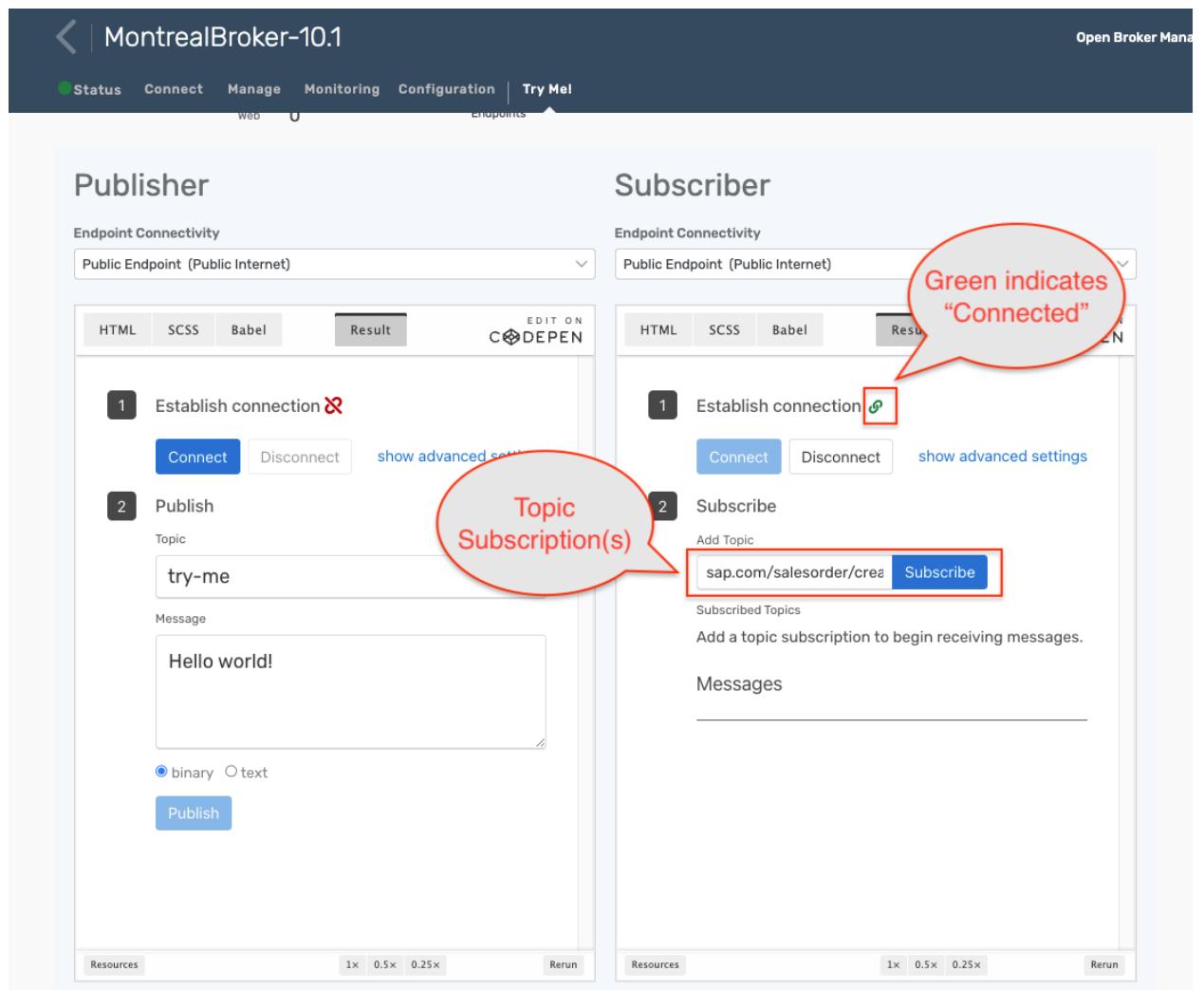


- Click on the Open Broker Manager button as below :



- Select the drop arrow on the **Publisher** side and enter the credentials of your Brokers. On the **Subscriber** select the "Same as Publisher" box and click **Connect**:





- You can use the below topic structures for different event types :

- **Sales Order :**
  - Create : sap.com/salesorder/create/>
  - Change : sap.com/salesorder/change/>
- **Business Partner :**
  - Create : sap.com/businesspartner/create/>
  - Change : sap.com/businesspartner/change/>
- **Chart of Accounts :**
  - Create : sap.com/chartofaccounts/create/>
  - Change : sap.com/chartofaccounts/change/>
- **Material Master :**
  - Create : sap.com/material/create/>

## Rapid Pilot Day 1

- Change : sap.com/material/change/>
- **Notifications :**
  - Create : sap.com/notification/create/>
  - Change : sap.com/notification/change/>
- As the simulator publishes events to the broker you should see events appearing in the subscribed topic(s)

## Takeaways

- Deploy SAP Simulator in BTP
- Test Events with AEM Try Me Tab

