Rapid Pilot Day 1

Rapid Pilot Program 2024

Contents

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.

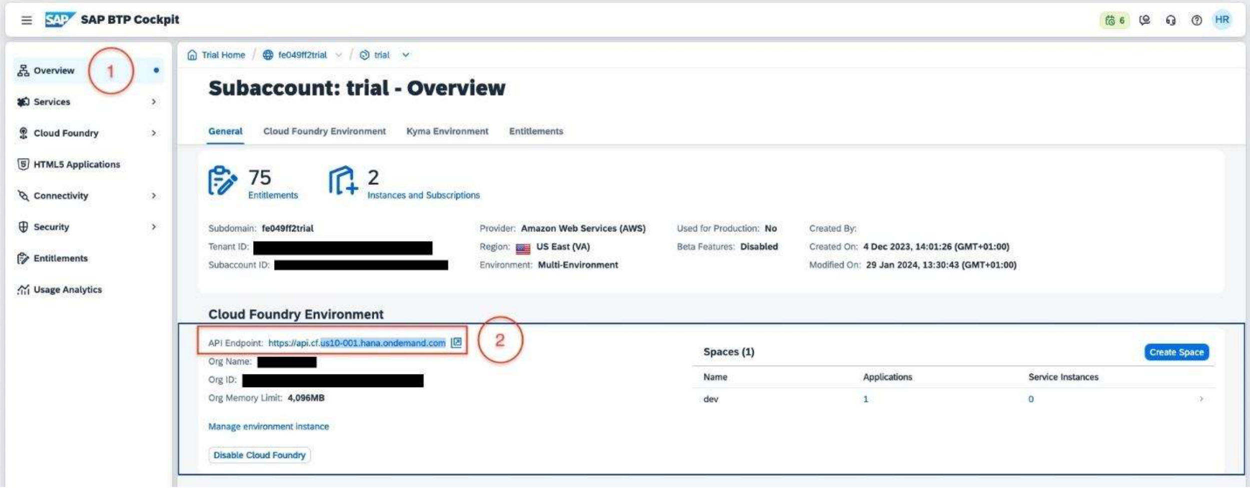
For more information on SAP CAP, you can refer to the link : SAP Cloud Application Programming Model

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 :



Subaccount: trial - Overview

Subdomain: fe049ff2trial
Tenant ID: [redacted]
Subaccount ID: [redacted]
Provider: Amazon Web Services (AWS)
Region: US East (VA)
Environment: Multi-Environment
Used for Production: No
Beta Features: Disabled
Created By:
Created On: 4 Dec 2023, 14:01:26 (GMT+01:00)
Modified On: 29 Jan 2024, 13:30:43 (GMT+01:00)

Cloud Foundry Environment

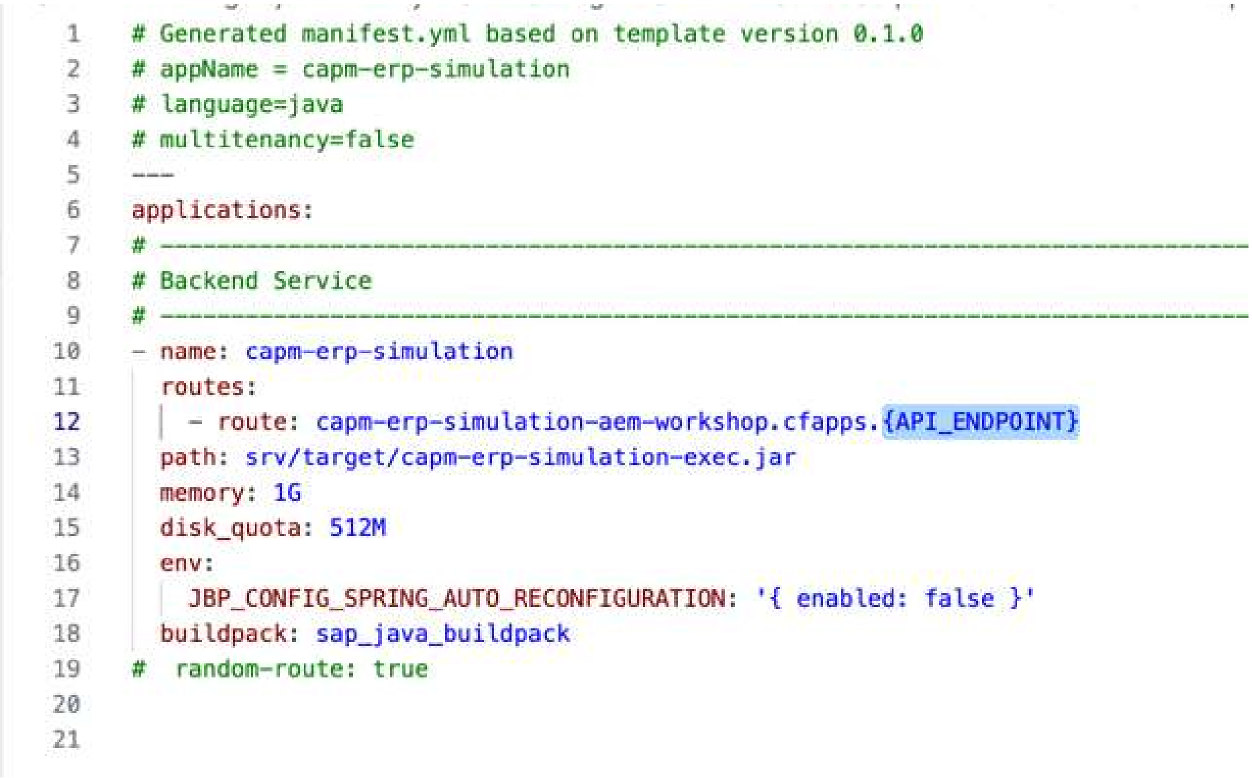
API Endpoint: https://api.cf.us10-001.hana.ondemand.com
Org Name: [redacted]
Org ID: [redacted]
Org Memory Limit: 4,096MB

Spaces (1)

Name	Applications	Service Instances
dev	1	0

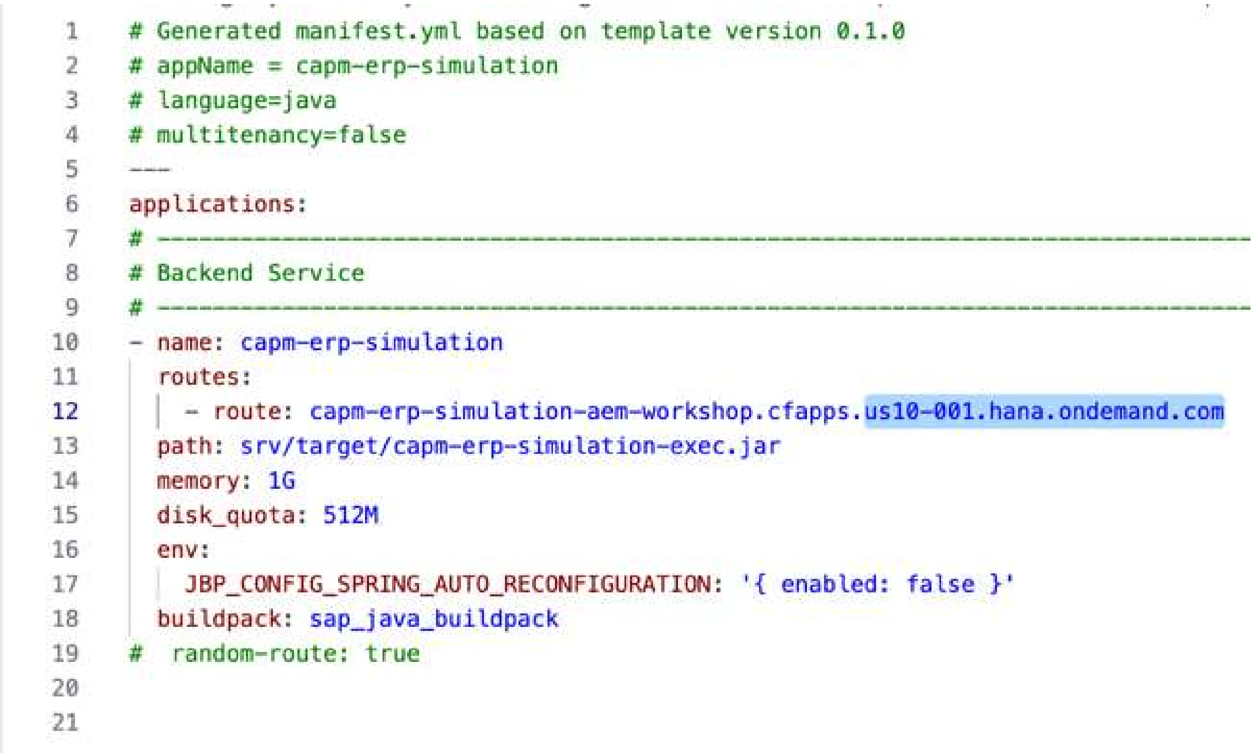
2 : Update the manifest file

- 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 :

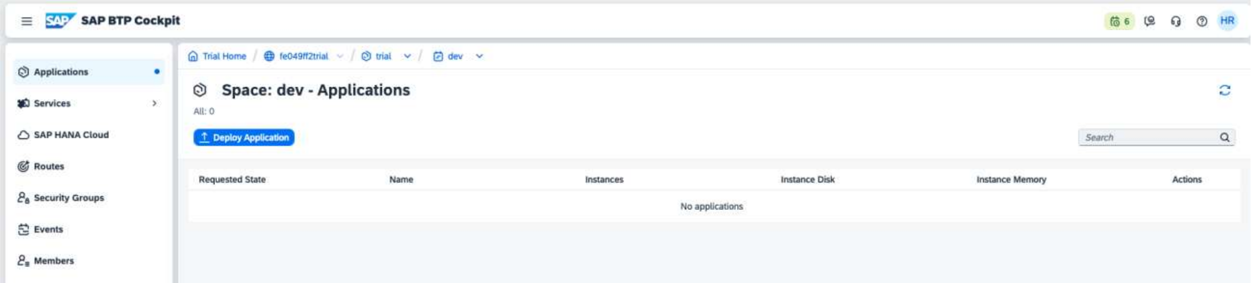


```
# Generated manifest.yml based on template version 0.1.0
# appName = capm-erp-simulation
# language=java
# multitenancy=false
---
applications:
# -----------------------------------------------------------------------------
# Backend Service
# -----------------------------------------------------------------------------
- name: capm-erp-simulation
  routes:
    - route: capm-erp-simulation-aem-workshop.cfapps.us10-001.hana.ondemand.com
  path: srv/target/capm-erp-simulation-exec.jar
  memory: 1G
  disk_quota: 512M
  env:
    JBP_CONFIG_SPRING_AUTO_RECONFIGURATION: '{ enabled: false }'
  buildpack: sap_java_buildpack
#  random-route: true
```

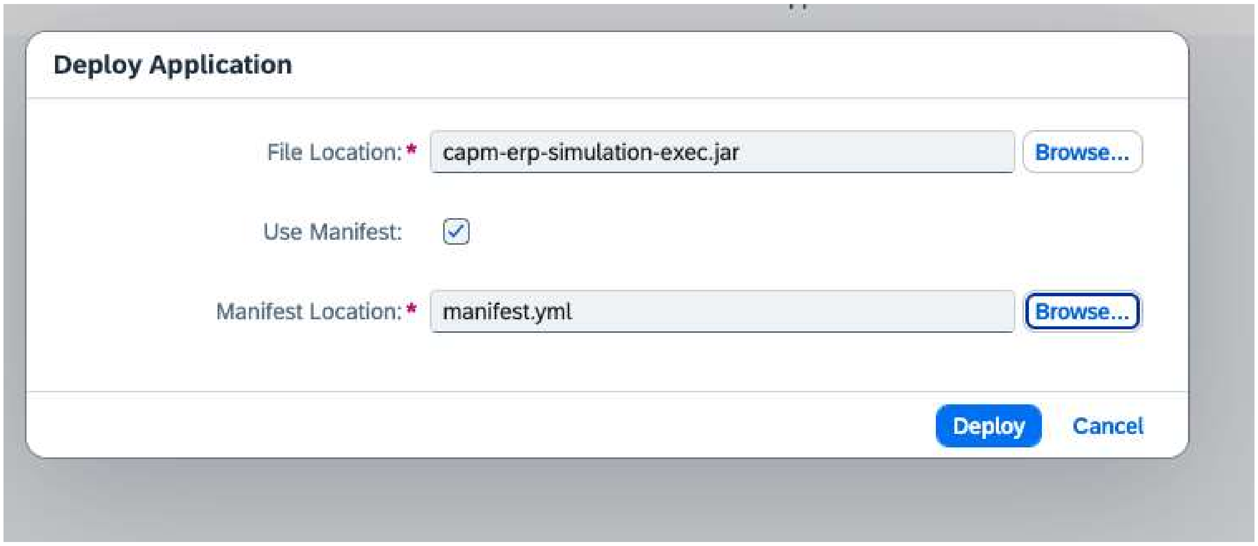
Save and close the file.

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 :



- 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



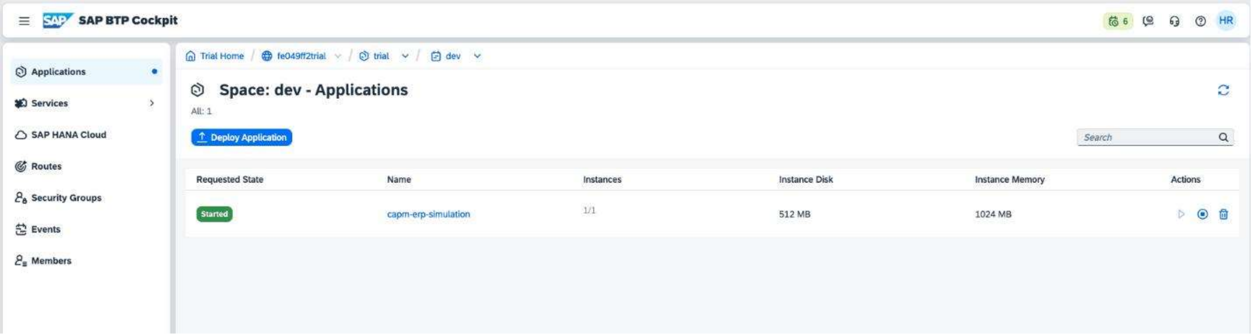
Deploy Application

File Location:* capm-erp-simulation-exec.jar

Use Manifest: ☑

Manifest Location:* manifest.yml

- 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 :



Space: dev - Applications

All: 1

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

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.

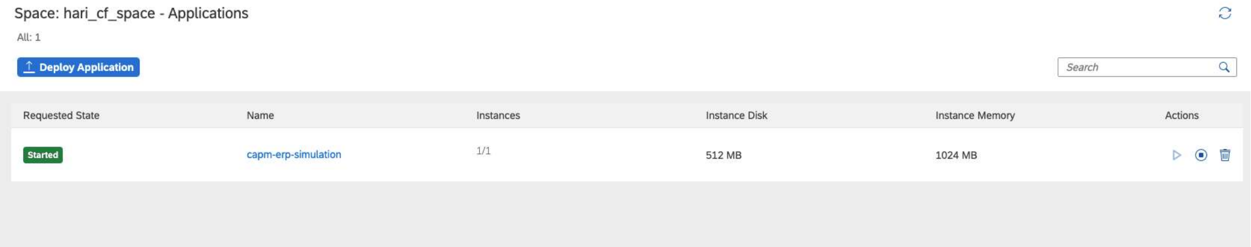
Once the command is completely executed, run the command cf apps to view a listing of the apps in your cloudfoundry space

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
  :

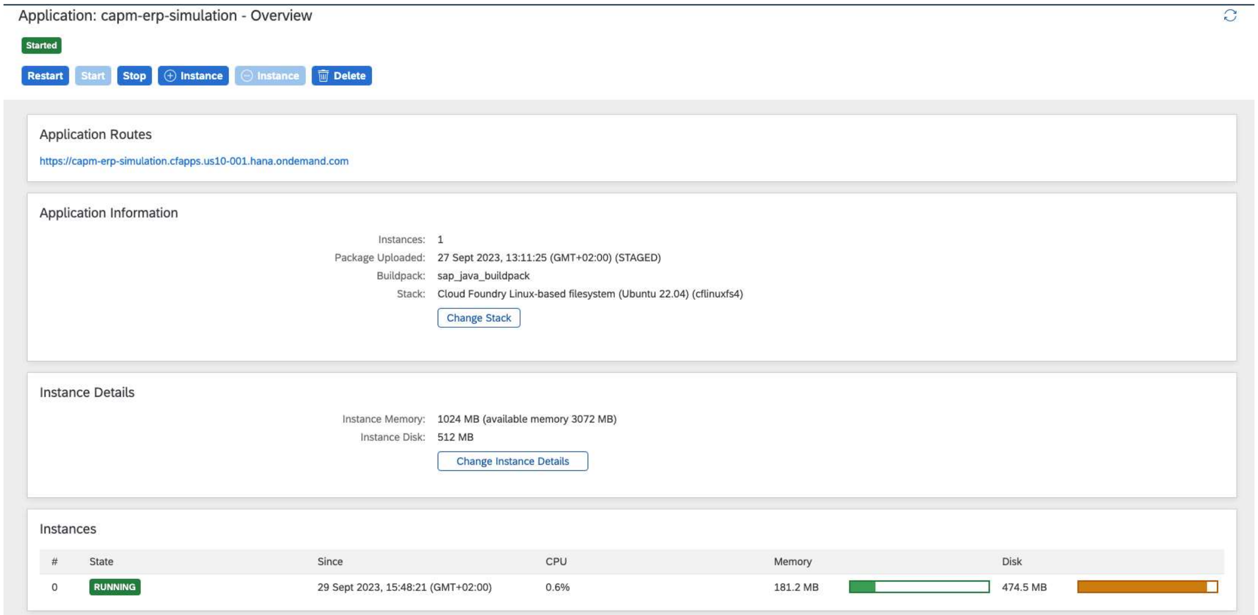


Space: hari_cf_space - Applications

All: 1

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.



Application: capm-erp-simulation - Overview

Started

Application Routes

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

Application Information

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)

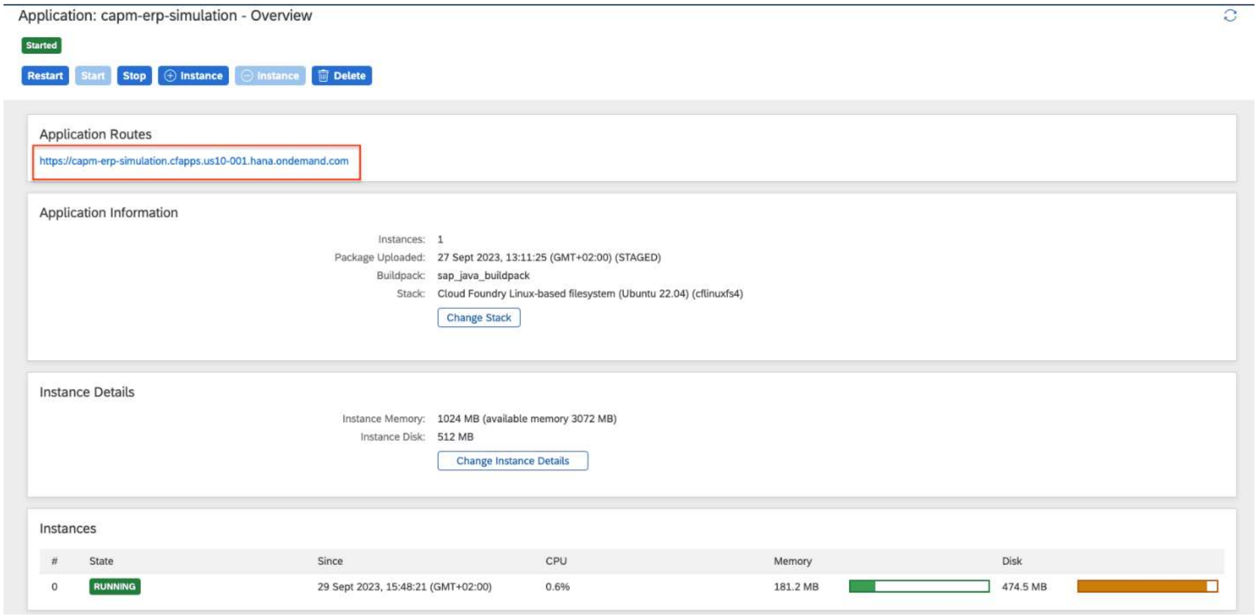
Instance Details

Instance Memory: 1024 MB (available memory 3072 MB)
Instance Disk: 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.



Application: capm-erp-simulation - Overview

Application Routes

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

Application Information

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)

Instance Details

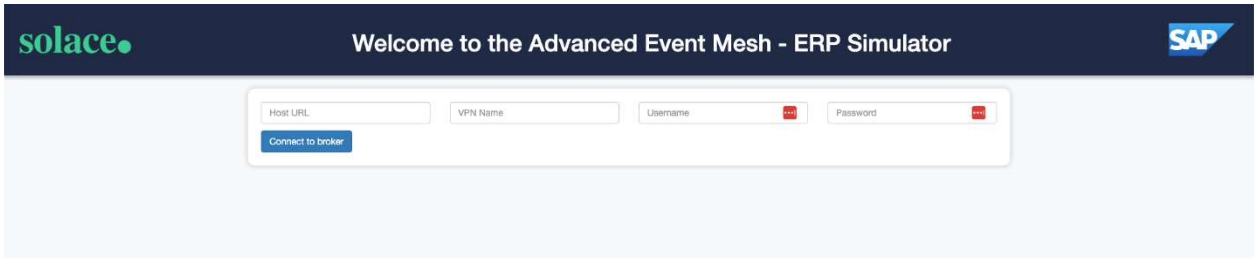
Instance Memory: 1024 MB (available memory 3072 MB)
Instance Disk: 512 MB

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

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

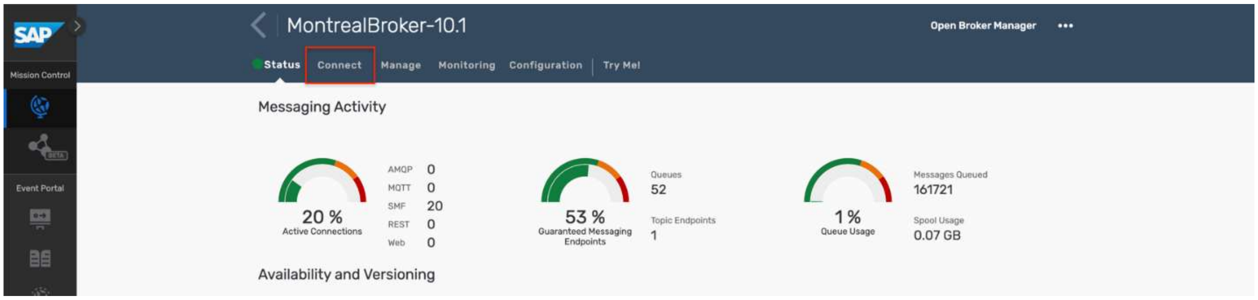


Welcome to the Advanced Event Mesh - ERP Simulator

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:



MontrealBroker-10.1

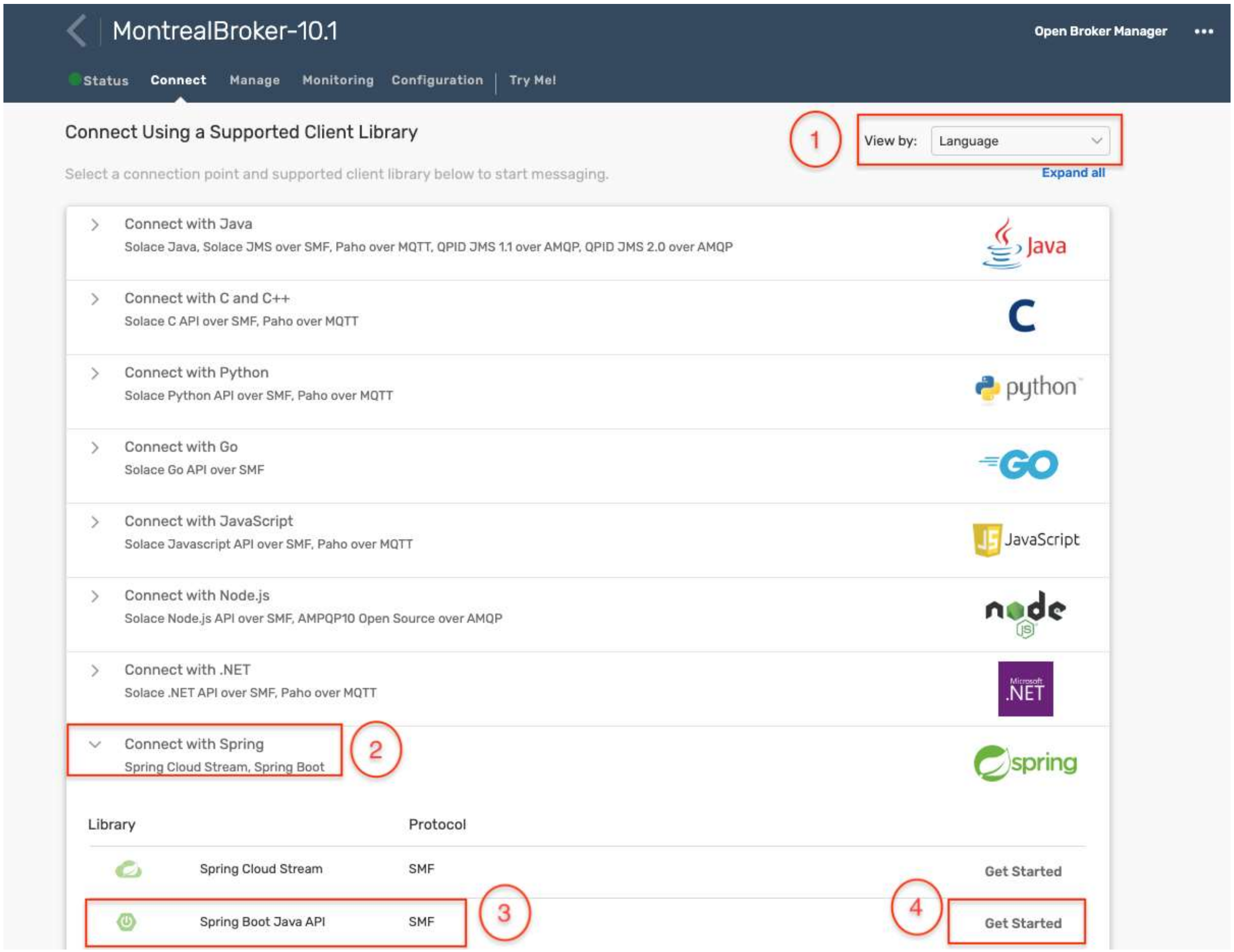
Messaging Activity

20 % Active Connections
AMQP 0
MQTT 0
SMF 20
REST 0
Web 0

53 % Guaranteed Messaging Endpoints
Queues 52
Topic Endpoints 1

1 % Queue Usage
Messages Queued 161721
Spool Usage 0.07 GB

Availability and Versioning



Connect Using a Supported Client Library

Select a connection point and supported client library below to start messaging.

View by: Language

Connect with Java
Solace Java, Solace JMS over SMF, Paho over MQTT, QPID JMS 1.1 over AMQP, QPID JMS 2.0 over AMQP

Connect with C and C++
Solace C API over SMF, Paho over MQTT

Connect with Python
Solace Python API over SMF, Paho over MQTT

Connect with Go
Solace Go API over SMF

Connect with JavaScript
Solace Javascript API over SMF, Paho over MQTT

Connect with Node.js
Solace Node.js API over SMF, AMPQP10 Open Source over AMQP

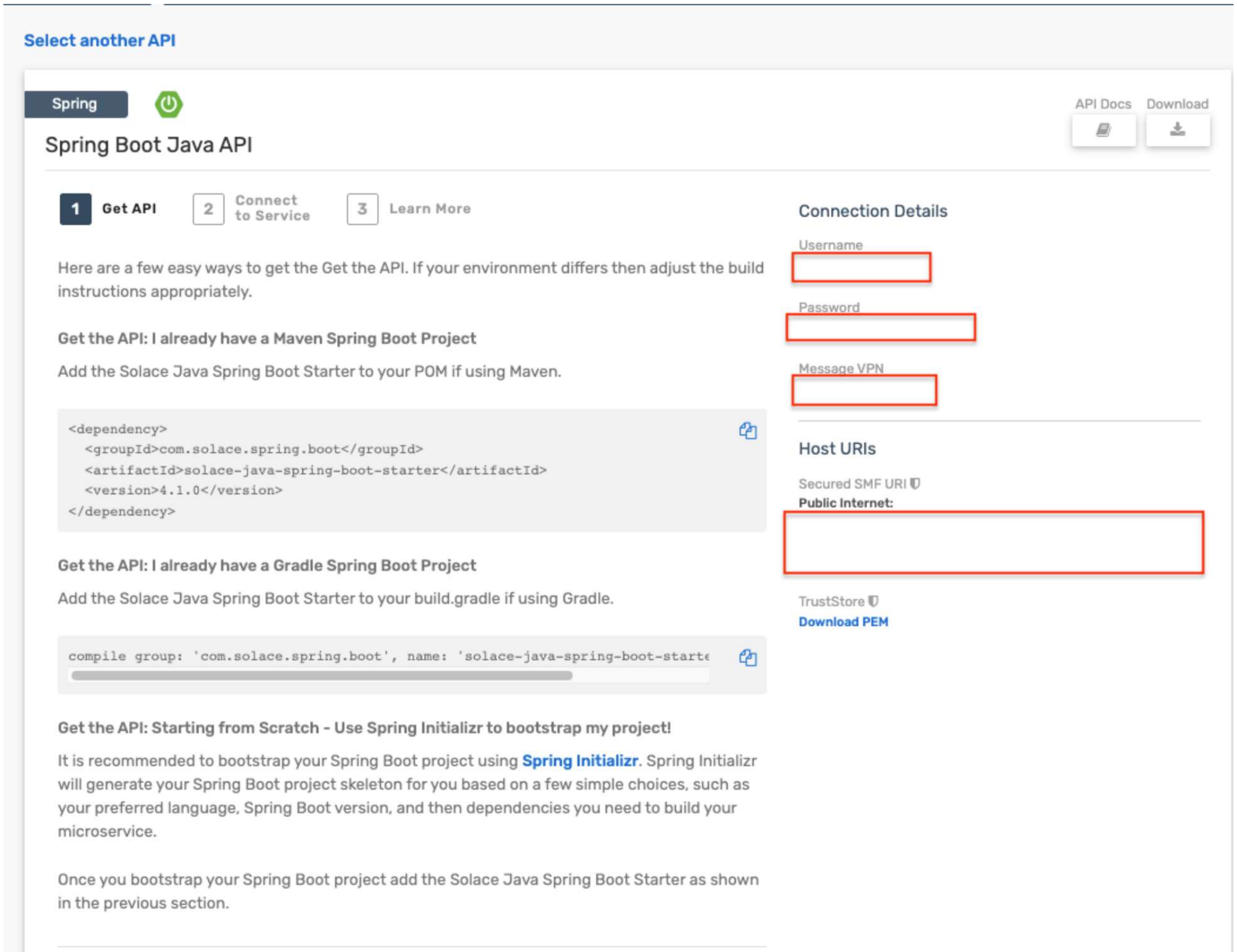
Connect with .NET
Solace .NET API over SMF, Paho over MQTT

Connect with Spring
Spring Cloud Stream, Spring Boot

Library	Protocol
Spring Cloud Stream	SMF
Spring Boot Java API	SMF

Enter the appropriate value as specified below :

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



Spring Boot Java API

Here are a few easy ways to get the 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-starte
```

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

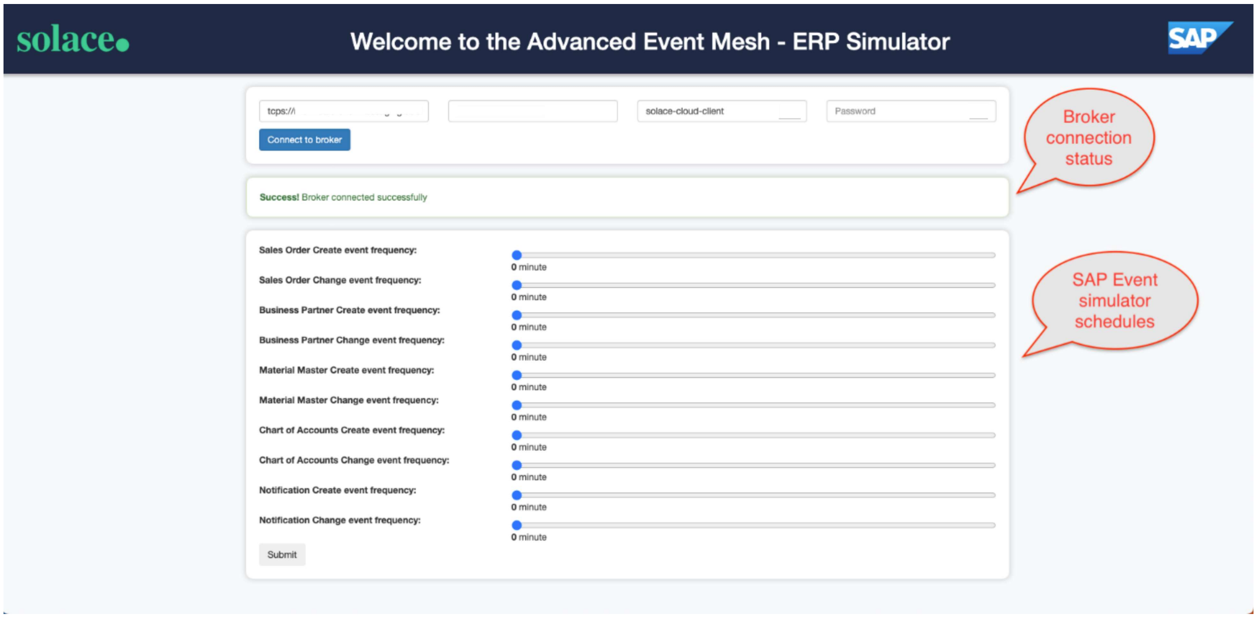
Secured SMF URI

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

tcps://

solace-cloud-client

Password

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

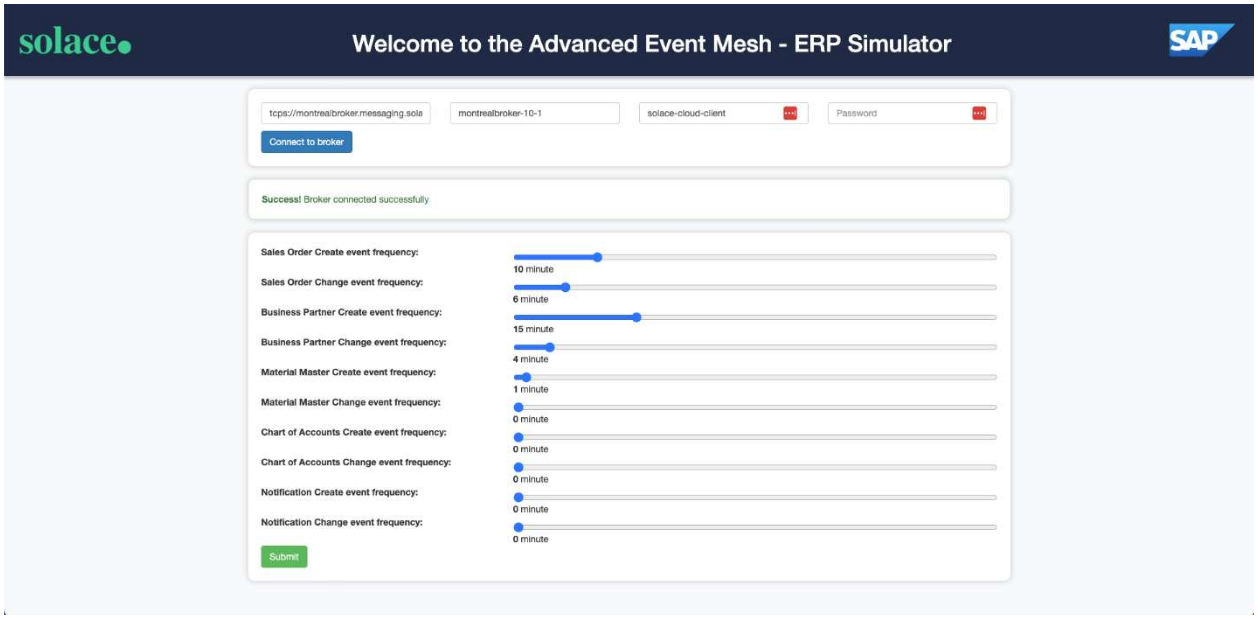
Chart of Accounts Create event frequency: 0 minute

Chart of Accounts Change event frequency: 0 minute

Notification Create event frequency: 0 minute

Notification Change event frequency: 0 minute

- 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

tcps://montrealbroker.messaging.sola | montrealbroker-10-1 | solace-cloud-client | Password

Connect to broker

Success! Broker connected successfully

Sales Order Create event frequency: 10 minute

Sales Order Change event frequency: 6 minute

Business Partner Create event frequency: 15 minute

Business Partner Change event frequency: 4 minute

Material Master Create event frequency: 1 minute

Material Master Change event frequency: 0 minute

Chart of Accounts Create event frequency: 0 minute

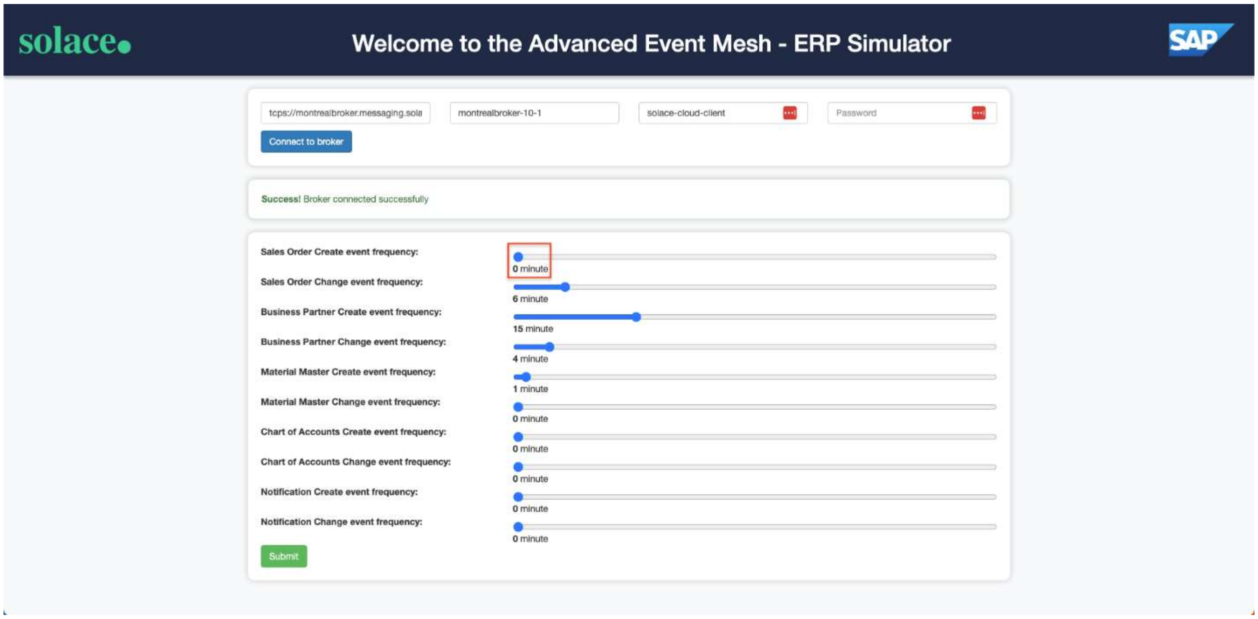
Chart of Accounts Change event frequency: 0 minute

Notification Create event frequency: 0 minute

Notification Change event frequency: 0 minute

Submit

- 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

tcps://montrealbroker.messaging.sola | montrealbroker-10-1 | solace-cloud-client | Password

Connect to broker

Success! Broker connected successfully

Sales Order Create event frequency: 0 minute

Sales Order Change event frequency: 6 minute

Business Partner Create event frequency: 15 minute

Business Partner Change event frequency: 4 minute

Material Master Create event frequency: 1 minute

Material Master Change event frequency: 0 minute

Chart of Accounts Create event frequency: 0 minute

Chart of Accounts Change event frequency: 0 minute

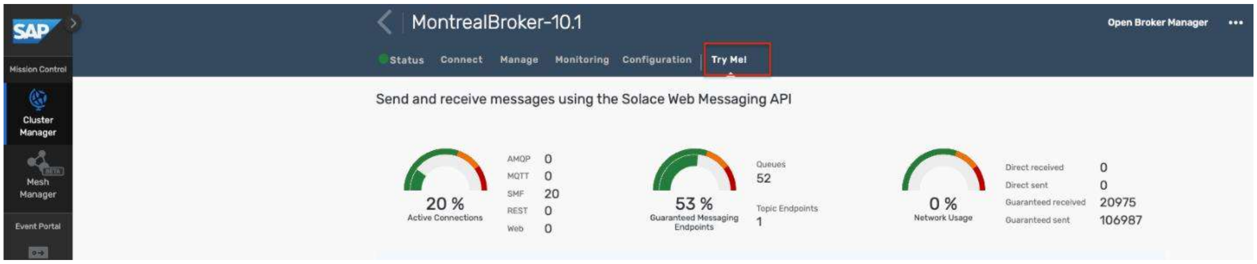
Notification Create event frequency: 0 minute

Notification Change event frequency: 0 minute

Submit

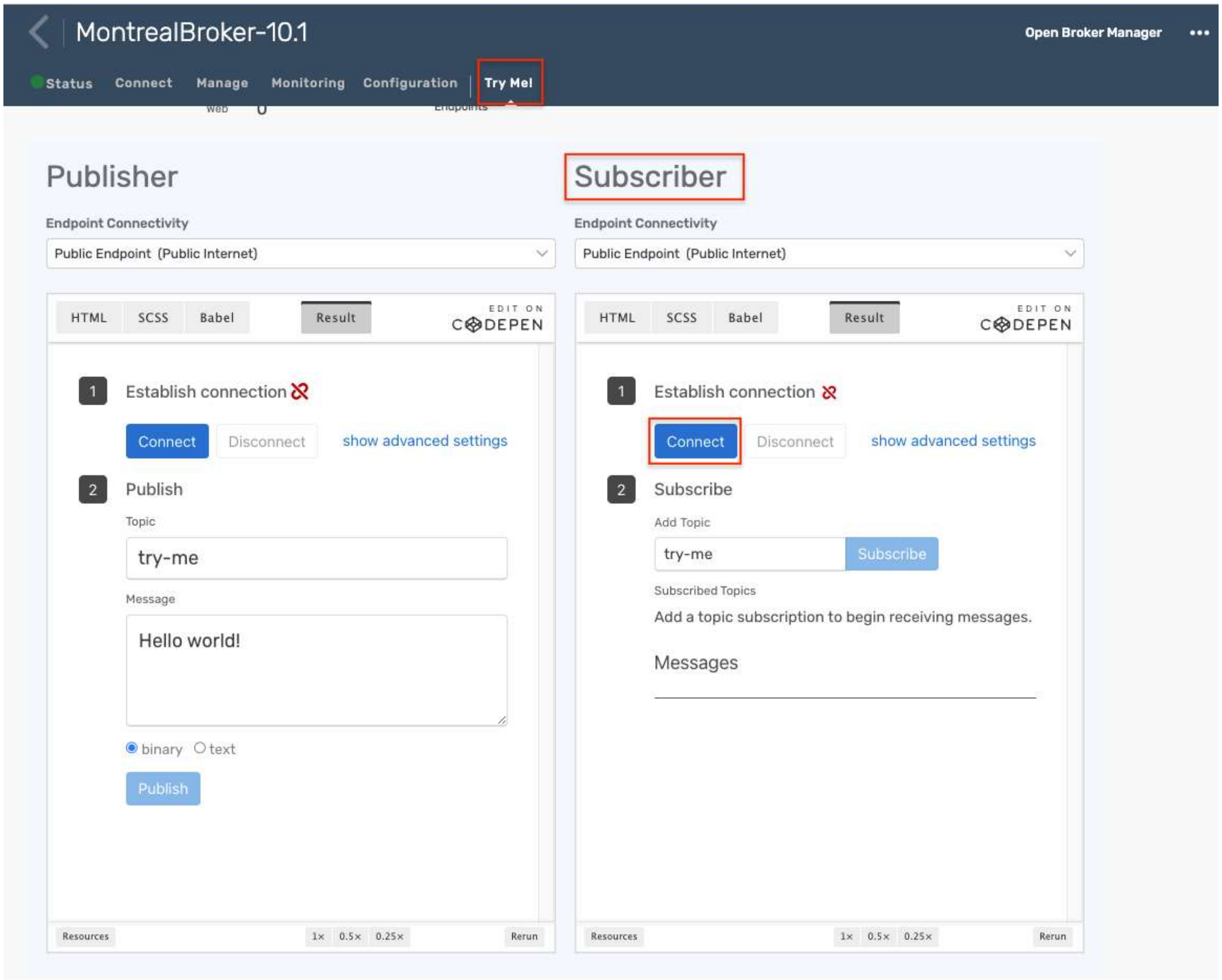
3 : Test the incoming events

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.



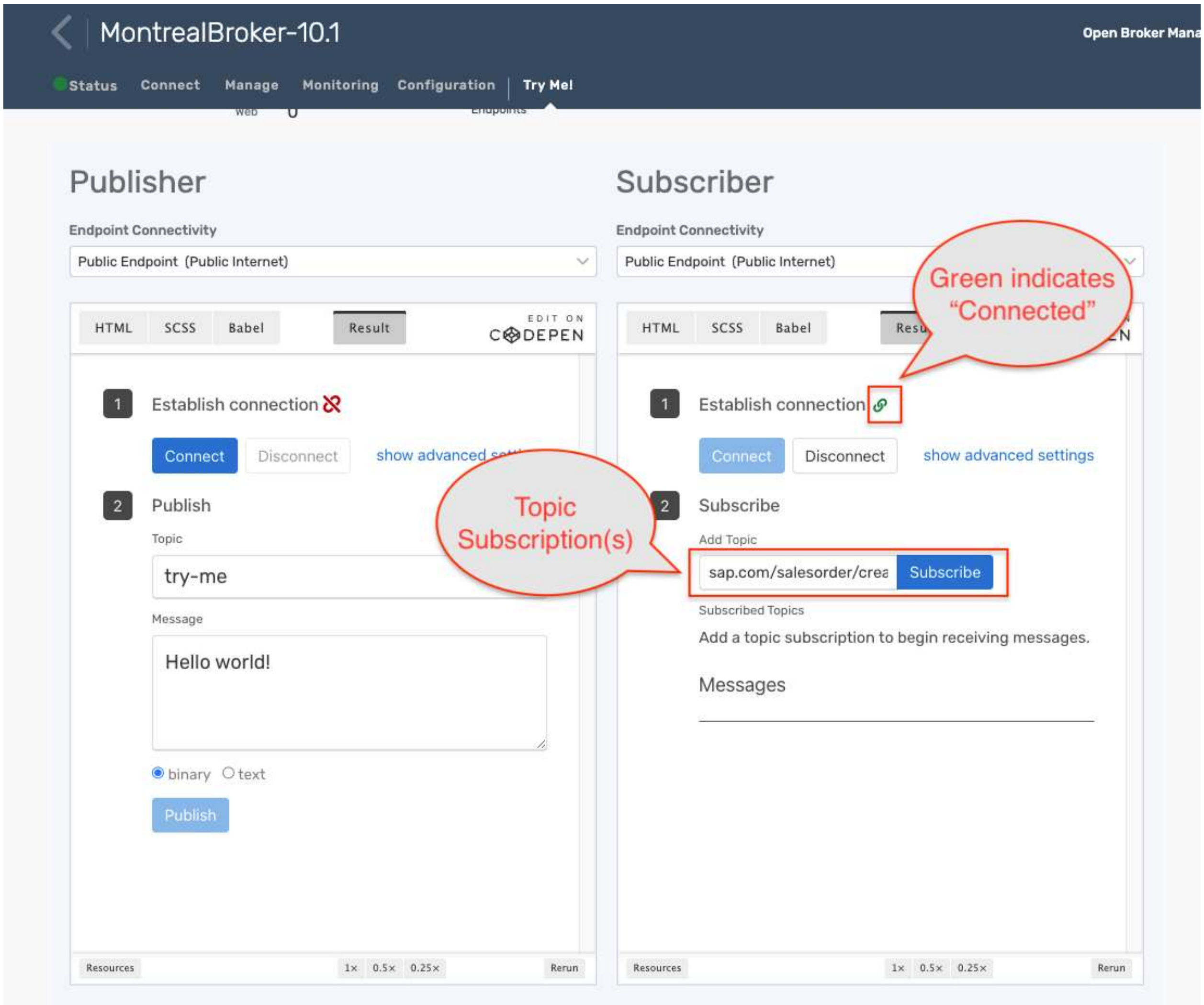
Send and receive messages using the Solace Web Messaging API

- Click on the **Connect** button in the **Subscriber** side of the panel as below
  :



Add a topic subscription to begin receiving messages.

- Add topic subscription(s) to view incoming messages on the topic as below:



Publisher

Endpoint Connectivity: Public Endpoint (Public Internet)

1 Establish connection

Connect | Disconnect | show advanced se...

2 Publish

Topic: try-me

Message: Hello world!

● binary ○ text

Publish

Subscriber

Endpoint Connectivity: Public Endpoint (Public Internet)

1 Establish connection

Connect | Disconnect | show advanced settings

2 Subscribe

Add Topic: sap.com/salesorder/crea | Subscribe

Subscribed Topics

Add a topic subscription to begin receiving messages.

Messages

Green indicates "Connected"

Topic Subscription(s)

- 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/>`
    - 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