

## Deploying the SAP ERP “Simulator”

During this workshop, we won't have the time to connect to a real SAP system. So, to simulate what would happen, we will use an iFlow to simulate the creation of a real business partner.

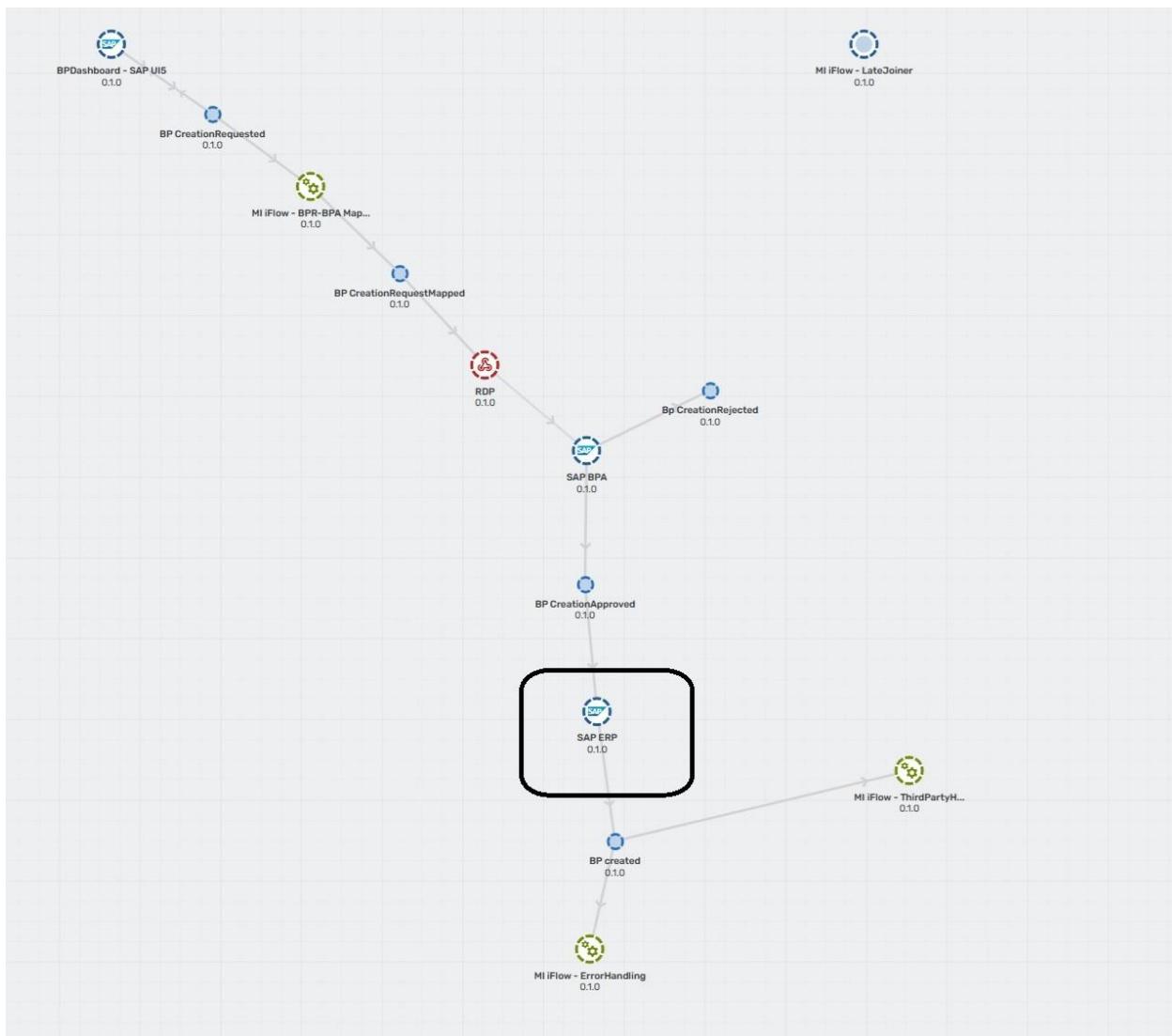
In a real environment, you would ingest the Event into the SAP system with one of many possible approaches (eg. IDOC, Rest Call, ASAPIO Ingestion) Message would be ingested by SAP, a business partner would be created and then SAP would publish a business partner event back to the mesh which is what we are simulating here. We will deploy a basic iflow, almost identical to the one you did earlier. An event comes in one side and gets sent back out with a different topic. Even though it's simple, we do recognize this a common requirement as events will need to be routinely converted from one format into another and will require different topics and/or payload.

The record that is being published by the SAP ERP system will represent the golden record and will be the one distributed to the various iFlows to simulate customer environments where Master Data needs to be shared with multiple systems. The goal will be to show how this can be done via eventing, providing a flexible and scalable way to do it without disruption.

## The Event Portal Model

The model below should now be starting to become familiar, and we will be deploying the “SAP ERP” application which in this case will be an iFlow. We will start by pushing the configuration for the “SAP ERP” application so that all the necessary artifacts are present on the AEM service. The key thing to remember here is that we will be pushing configurations for individual applications without affecting the already running applications.

From the Designer, bring up the Event Portal Model as shown below.



Double click on the SAP ERP Application to reveal the properties:

Design Details		Version Details	
Version	0.1.0	Version Name	None
State	<span>Draft</span>		
Description	Approved BusinessPartner record created into SAP ERP system.		
<span>Events</span> <span>Events</span> <span>Advanced Subscription Setup</span>			
Published Events			
<span>BP created</span> 0.1.0 <span>Draft</span> sap / businesspartner / created / V1 / {businessPartnerType} / {country} / {city} <span>...</span>			
Subscribed Events			
<span>BP CreationApproved</span> 0.1.0 <span>Draft</span> sap / businesspartner / creationApproved / V1 / {businessPartnerType} / {country} / {city} <span>...</span> <span>1 subscription</span>			

Close

Now we will create a configuration to be pushed for to the AEM instance. From here you can see the Events published and subscribed for this Application. In addition, if you click the “Advanced Subscription Setup”, you will see the subscription and the queue that will be used by this application.

SAP ERP 0.1.0 Draft

[Open Application](#)

		Design Details	Version Details
<b>Design Details</b> Set version details, event flow, and consumers	Version	0.1.0	
	Version Name	None	
	State	<span>Draft</span>	
<b>Configurations</b> Add broker configuration and connection details	Description	Approved BusinessPartner record created into SAP ERP system.	
<b>Promotions</b> Decide where the application gets deployed	Events	<a href="#">Advanced Subscription Setup</a>	
	Consumers represent means of sending events to applications from event brokers. <a href="#">Learn about consumers</a>		
	<b>erp.bp_creation_approved</b> Solace Event Queue		
	<b>Subscriptions (1)</b> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">           sap/businesspartner/creationApproved/V1/&gt;           <span style="float: right;">1 Event</span> </div>		

From here, click on “Open Application” in the top right corner. Now select “Create Configurations”.

Designer > RPPV2\_MDM > SAP ERP

SAP ERP

Event Access All Approved		Verify your application is ready for promotion
Configure for Event Brokers		<a href="#">Create Configurations</a>
Application has global default configurations Start creating configurations to customize the global configurations for specific event brokers. <a href="#">Create Configurations</a>		

**Design Details**  
Set version details, event flow, and consumers

**Configurations**  
Add broker configuration and connection details

**Promotions**  
Decide where the application gets deployed

Now you must select the Broker to which the configuration will be pushed. Select the Broker and the “Next”.

## Configure Application: SAP ERP 0.1.0 Draft



What event broker do you want to configure for?

Default

Event Broker

AEM\_Broker X ▼

[Cancel](#)

[Next: Define Event Handling >](#)

The next section should be pre-populated for you based on the portal if the design work has been done properly. We can see that one queue will be created. Press “Next”.

## Configure Application: SAP ERP 0.1.0 Draft



How do you want your events treated?

Show Advanced Options

**erp.bp\_creation\_approved**

Solace Event Queue **1 Event** | **1 Subscription**

Queue Name \*

erp.bp\_creation\_approved

Queue Configuration

Default configuration applied

[Back: Set Context](#)

[Cancel](#)

[Save & Close](#)

[Next: Define Access Control >](#)

Here you will define the user that will be deployed onto the broker with the corresponding permissions. In this case the user is “user5” and the password is “user123\$”.

\*\*\* The user will be used on the iFlow and it’s password will be a secure parameter that is mentioned later\*\*\*

Hit Next....

## Configure Application: SAP ERP 0.1.0 Draft



Which clients can access the events?

### Client Credentials

Manage the credentials that the runtime client application uses to connect to the event broker. [Learn more about managing credentials.](#)

Access Type *	Basic	Client Username and Password <span>▼</span>
Client Username *	user5	
Password *	user123\$	

[Back: Define Event Handling](#)

[Cancel](#)

[Save & Close](#)

[Next: Review >](#)

In this next step , you **could** hit “Start Promotion” but we would suggest for clarity, hit “Save and Promote Later” so you can fully review and see what is being deployed.

## Configure Application: SAP ERP 0.1.0 Draft



Review your configuration choices

Show Advanced Options

Configuring for AEM\_Broker

[Define Event Handling](#) [Define Access Control](#)

**erp.bp\_creation\_approved**

Solace Event Queue [1 Event](#) | [1 Subscription](#)

Queue Name \* erp.bp\_creation\_approved

Queue Configuration Default configuration applied

When you're ready, promote this object to the selected event broker

[Start Promotion](#)

[Back: Define Access Control](#)

[Cancel](#)

[Save & Promote Later](#)

Click on “Configurations”

Designer > RPPV2\_MDM > SAP ERP

SAP ERP

Solace Standard Application  
Click to view in graph

Version  
0.1.0 Draft Latest +

Version Actions  
Manage Lifecycle State  
Download AsyncAPI  
Delete Version

Design Details  
Set version details, event flow, and consumers

Configurations  
Add broker configuration and connection details

Promotions  
Decide where the application gets deployed

Event Access All Approved

Configure for Event Brokers

Event brokers with configurations created

Default  
AEM\_Broker

Configuration Details

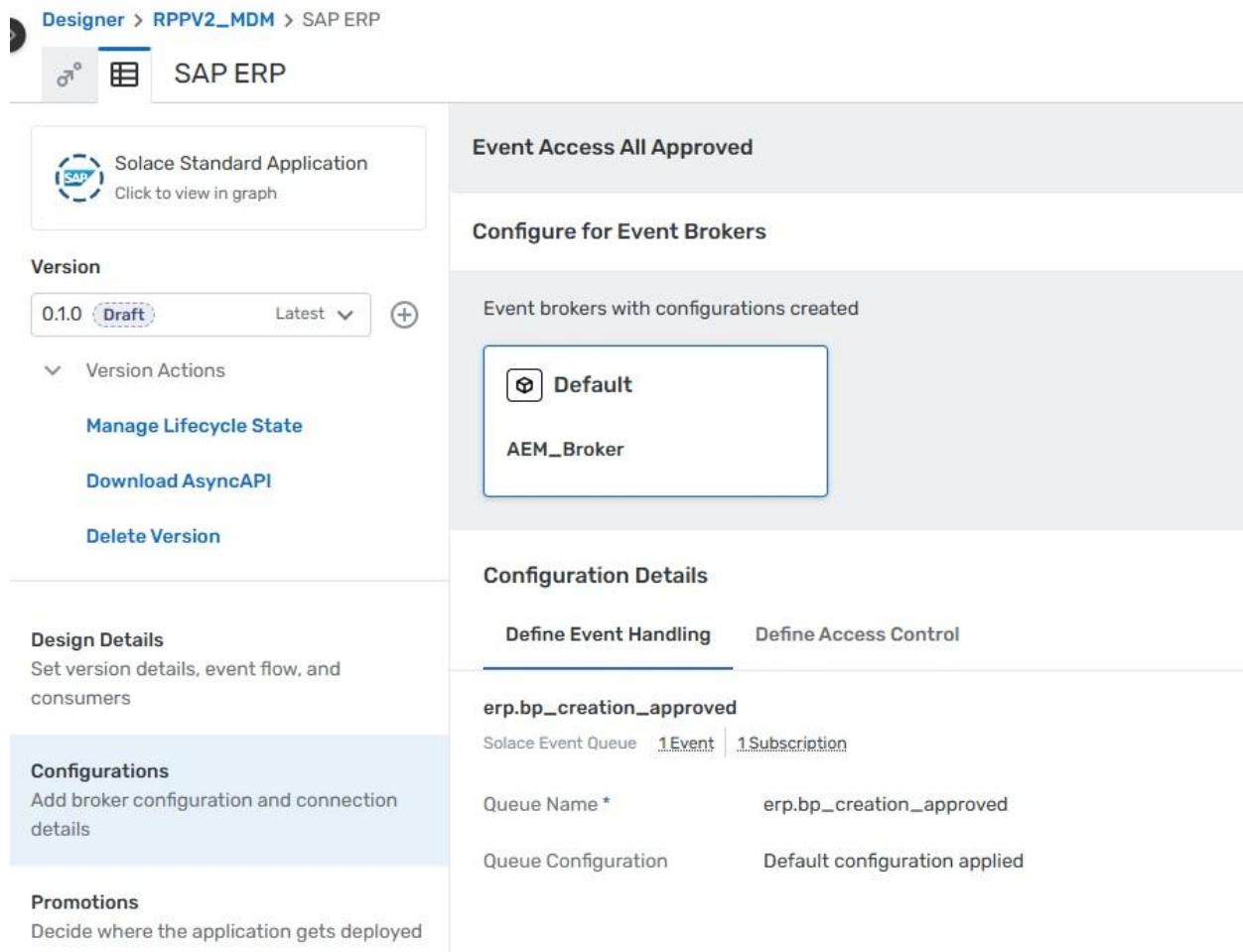
Define Event Handling Define Access Control

erp.bp\_creation\_approved

Solace Event Queue 1 Event | 1 Subscription

Queue Name \* erp.bp\_creation\_approved

Queue Configuration Default configuration applied



Click "Promote Version"

Designer > RPPV2\_MDM > SAP ERP

SAP ERP

Solace Standard Application  
Click to view in graph

Version  
0.1.0 Draft Latest +

Version Actions  
Manage Lifecycle State  
Download AsyncAPI  
Delete Version

Design Details  
Set version details, event flow, and consumers

Configurations  
Add broker configuration and connection details

Promotions  
Decide where the application gets deployed

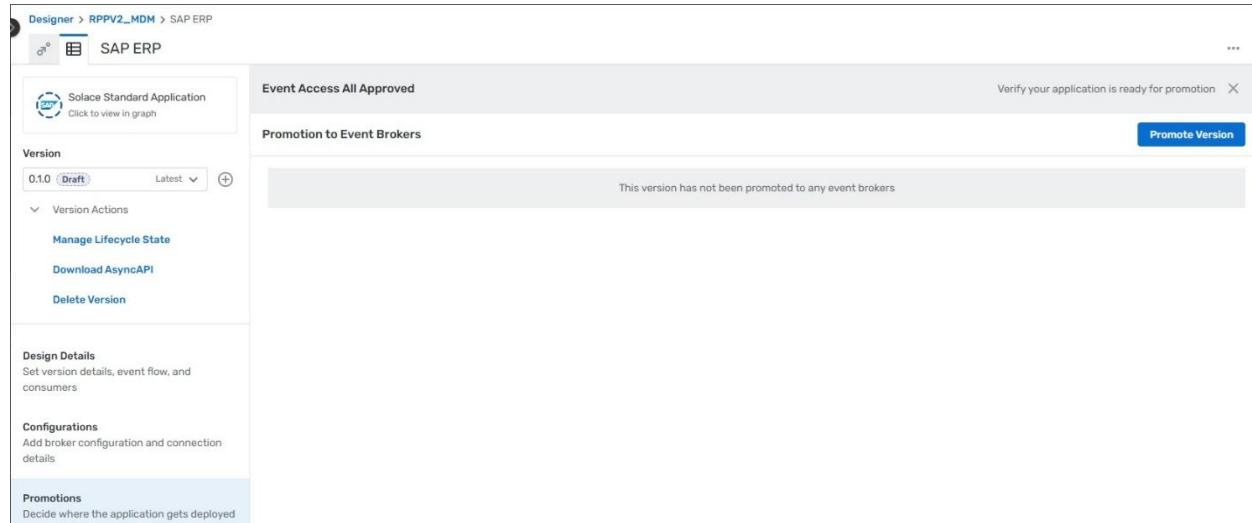
Event Access All Approved

Promotion to Event Brokers

Verify your application is ready for promotion X

Promote Version

This version has not been promoted to any event brokers



## Select Preview Promotion

Promote: SAP ERP 0.1.0 Draft

<input checked="" type="checkbox"/> Default	X
AEM_Broker	

### What does promotion do?

Application will be added to MDM\_Event\_Manager in Runtime Event Manager.

Configuration will be deployed to event broker AEM\_Broker.

Application is ready for promotion.

[Cancel](#)

[Preview Promotion](#)

From here, you can expand all the objects that are “to be added” on the right to see what is exactly being pushed and the configuration associated with each of the objects. Once you are ready, hit the “promote” button.

Promote SAP ERP 0.1.0 (Draft)

**Application**

Name	SAP ERP
Version	0.1.0 <span>(Draft)</span>
<b>Consumers</b>	
Event Queues	1

**Event Broker**

Name	AEM_Broker <span>MDM_Event_Manager</span>
Configuration	Enabled <span>(?)</span>

**Client Credentials**

Access Type	Basic
Client Username	user5
Password	***** <span>(?)</span>
Client Profile	default

**Preview** (?)

**Access Control**

- > ACL Profile To be added
- > Client Credentials To be added

**Queues**

- > erp.bp\_creation\_approved To be added

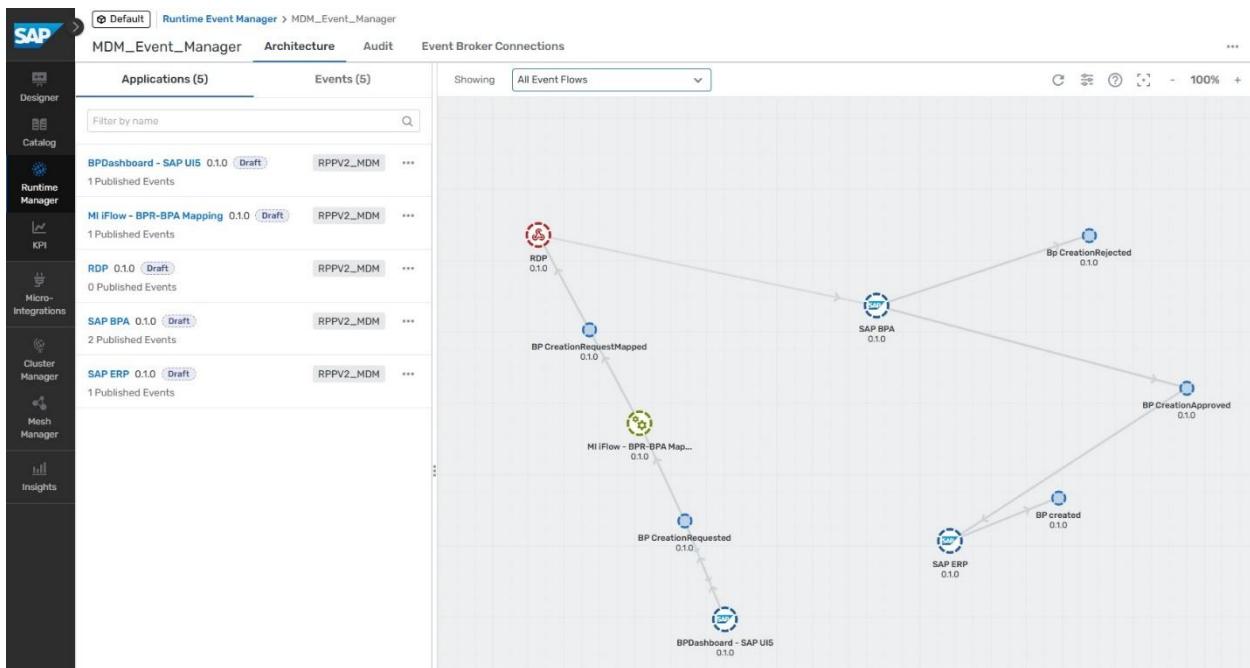
Cancel Promote

It should be that simple but now we can validate what has been pushed so far. Select the “Runtime Manager” from the left side menu.

The screenshot shows the SAP Runtime Event Manager interface. The left sidebar has a navigation menu with icons for Designer, Catalog, Runtime Manager (which is selected), KPI, Micro-Integrations, Cluster Manager, Mesh Manager, and Insights. The main content area is titled "Runtime Event Manager: Modeled Event Meshes". It features a search bar with a filter dropdown set to "Default". Below the search bar is a table with the following data:

Name	Event Brokers	Broker Type	Environment	Last Updated
MDM_Event_Manager	1	Solace	Default	2 days ago

From here, you can see each of the applications that has been deployed.



Now that the configuration has been pushed, we can deploy the iFlow.

## Deploying the iFlow

From the GitHub deployable directory, within the iFlow sub folder [GITHUB](#)

You will find 2 artifacts to deploy....a Message Mapping and an iFlow:

- ERPSIM.zip – This is your message mapping

Integrations and APIs / AEM-Rapid-Pilot-V2 / ERPSIM /

**ERPSIM**

Message Mapping

References

**Local (1) Global (0)**

Name	Type	Size	Actions
erpbpcreator	JSON	2 KB	<a href="#">Download</a>

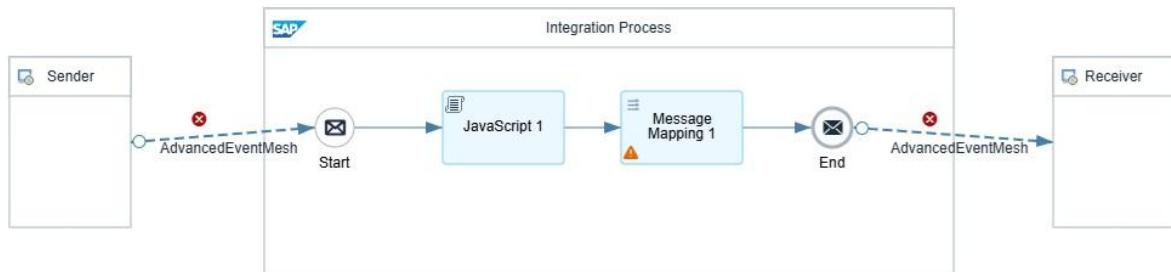
/erpcreate-POST-Request

Structure	Occurrence
partnerId	1..1
validFrom	1..1
validTo	1..1
addressNumber	1..1
businessPartnerType	1..1
firstName	1..1
lastName	1..1
city	1..1
postalCode	1..1
street	1..1
houseNumber	1..1
country	1..1

/erpcreate-POST-Request

Structure	Occurrence
partnerId	1..1
validFrom	1..1
validTo	1..1
addressNumber	1..1
businessPartnerType	1..1
firstName	1..1
lastName	1..1
city	1..1
postalCode	1..1
street	1..1
houseNumber	1..1
country	1..1

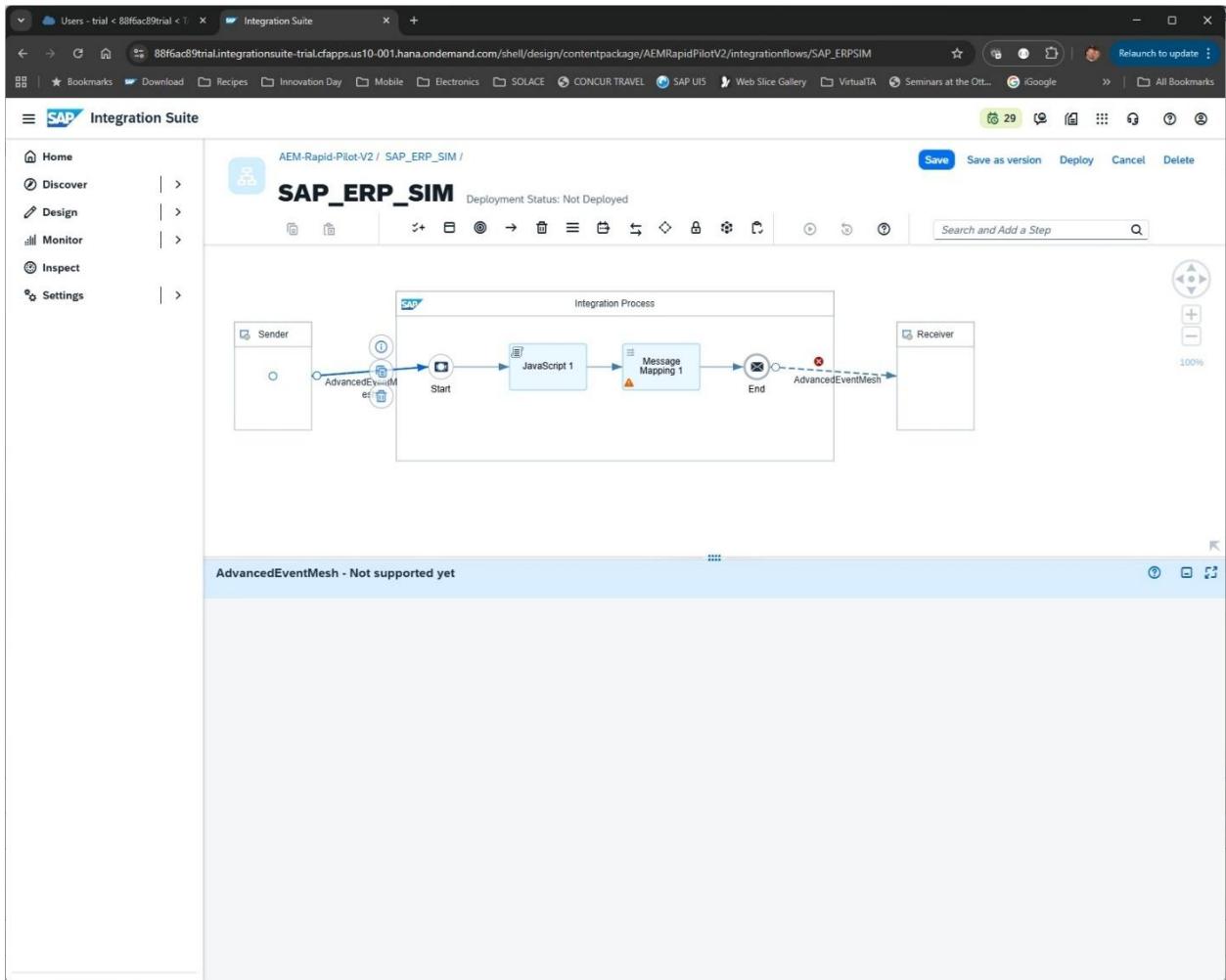
- SAP\_ERP\_SIM.zip – This is your iFlow



This is a basic iFlow. If you open up the Javascript file, you will see the small bit of code that is used to generate the topic which is then referenced in the properties for the Receiver AEM Configuration.

\*\*\*Yes, I realize there are a few red marks but wanted everyone to see this because this is completely normal after uploading the iFlow from scratch as we have not configured anything yet.

In addition, please check out the following screenshot:



If you see this after you click on the Advanced Event Mesh Adapter, it's because you have yet to use the AEM adapter in your environment. Please drop the connection between the sender and the iFlow and re-add it back in by dragging the line between the Sender and the start block and select "Advanced Event Mesh". This will install the adapter and that message will disappear.

Now will install the 2 artifacts. From the business package within Cloud Integration, you will start by importing the message mapping. Make sure you are in Edit mode and proceed to "Add" and select "Message Mapping".

Integrations and APIs / AEM-Rapid-Pilot-V2 /

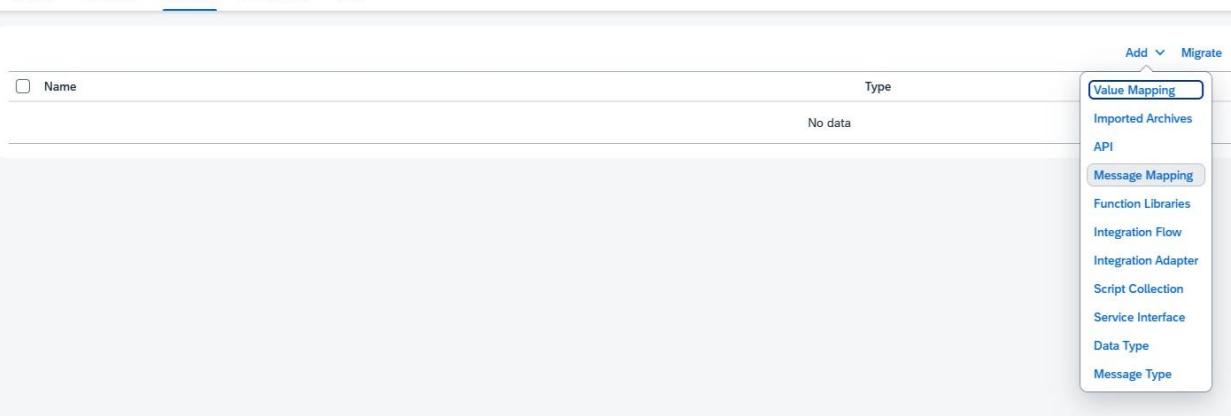
## AEM-Rapid-Pilot-V2

Header Overview Artifacts Documents Tags

Add Migrate

Name	Type
	No data

Value Mapping  
Imported Archives  
API  
**Message Mapping**  
Function Libraries  
Integration Flow  
Integration Adapter  
Script Collection  
Service Interface  
Data Type  
Message Type



Select the “ERPSIM” zip file that you downloaded from GITHUB.

Add Message Mapping

1 Add

1. Add

Create  Upload  ES Repository

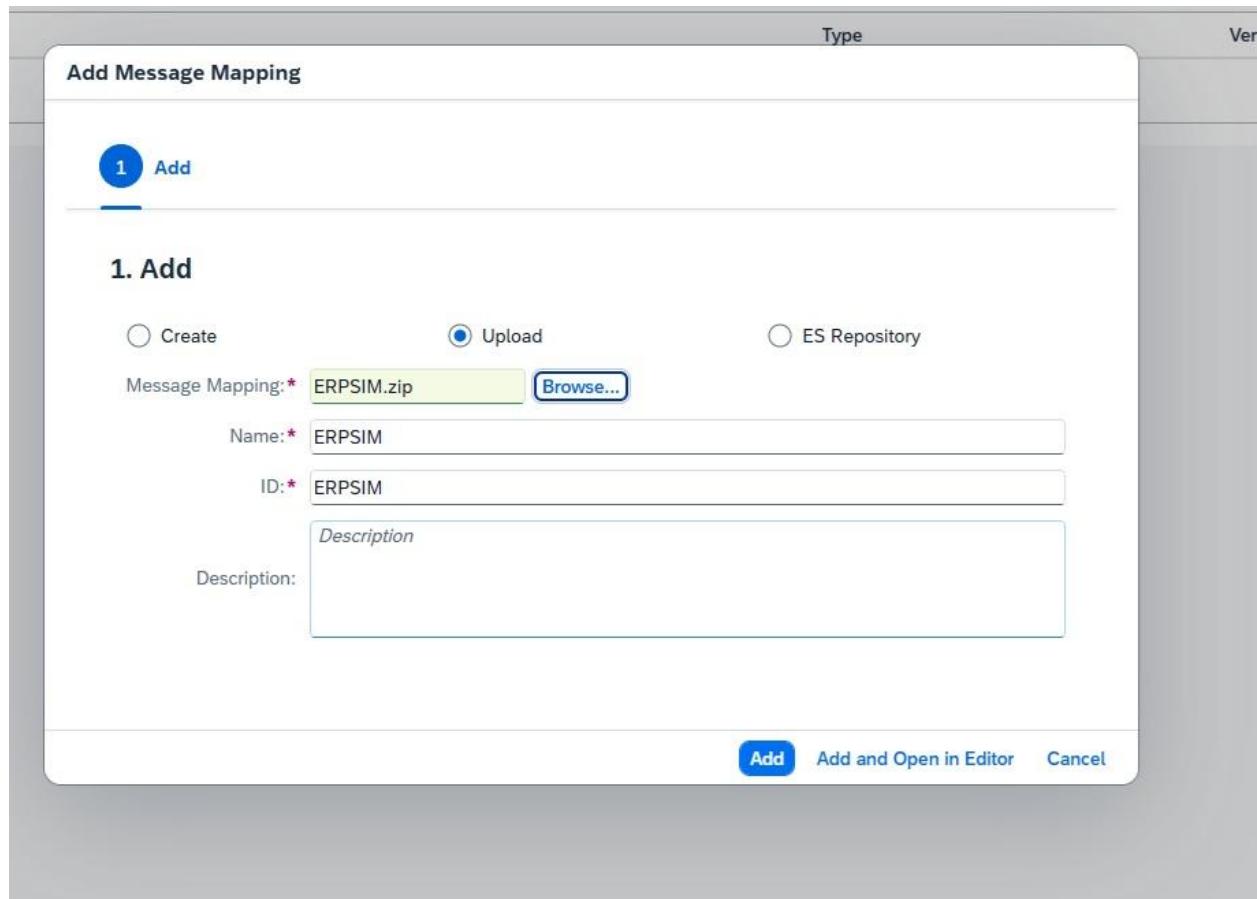
Message Mapping: \* ERPSIM.zip

Name: \* ERPSIM

ID: \* ERPSIM

Description:

Add Add and Open in Editor Cancel



Hit the “Add” button and you should now see the mapping. This mapping will be referenced as a Global asset in the iFlow.

The screenshot shows the AEM-Rapid-Pilot-V2 interface. At the top, there are buttons for Save, Export, Cancel, and Delete Package. Below this is a navigation bar with Header, Overview, Artifacts (1), Documents, and Tags. The Artifacts (1) tab is selected. The main area displays a table with one row. The row contains a checkbox labeled 'Name', the value 'ERPSIM', a 'Type' column showing 'Message Mapping', a 'Version' column showing '1.0.2', and an 'Actions' column with three dots and a right arrow. There is also a 'Filter Artifacts' search bar at the top of the table.

Now we will repeat the process for the iFlow:

From the Add dropdown, select “Integration Flow”.

The screenshot shows the AEM-Rapid-Pilot-V2 interface. The Artifacts (1) tab is selected. The main area displays a table with one row. The row contains a checkbox labeled 'Name', the value 'ERPSIM', a 'Type' column showing 'Message Mapping', and an 'Actions' column with three dots and a right arrow. An 'Add' button is visible above the table. A dropdown menu is open from the 'Add' button, listing several options: Value Mapping, Imported Archives, API, Message Mapping, Function Libraries, Integration Flow (which is highlighted with a blue border), Integration Adapter, Script Collection, Service Interface, Data Type, and Message Type.

Now select the “SAP\_ERP\_SIM” zip file that you downloaded earlier and hit “Add”.

Message Mapping

### Add Integration Flow

Create       Upload

Integration Flow: \*

Name: \*

ID: \*

Description:

Sender:

Receiver:

You should now have the 2 artifacts that you need for the next section

Integrations and APIs / AEM-Rapid-Pilot-V2 / Save Export

### AEM-Rapid-Pilot-V2

Header Overview Artifacts (2) Documents Tags

<input type="checkbox"/> Name	Type	Version	Actions
<input type="checkbox"/> ERPSIM Created	Message Mapping	1.0.2	<input type="button" value="..."/>
<input type="checkbox"/> SAP_ERP_SIM Created	Integration Flow	1.0.0	<input type="button" value="..."/>

From here, we only need to configure the iFlow, there is nothing to change on the mapping so click to open the “SAP\_ERP\_SIM” iflow and click the Edit button.

From here, you will click on the “Dotted Line” for the Sender where it says “Advanced Event Mesh” and then proceed to click the “Connection” tab. Here you will see all the properties that you need to specify in order for the Adapter to connect to your instance of “AEM”.

\*\*\*\*\* Quick Glance to where you can find connection information \*\*\*\*\*

In case you need a reminder of where to find those parameters, check out the following screenshot from the management console.

From here, you will take the VPN name and also the host. Notice I have selected the first category which is "Solace Messaging"...if you select Solace Web Messaging it will be a different port and you won't get very far 😊



The username and password will be highlighted below but this information will be pushed by the Event Portal to the AEM instance. You however will need to create this security artifact on Cloud Integration

The screenshot shows the SAP BPCreate interface with the 'Cluster Manager' selected. The main area displays 'Service Details: AEM\_Broker'. Under the 'Connect' tab, there's a section for 'Solace Messaging' with a table of client libraries:

Technology	Library	Language	Action
solace•	Solace Java API	Java	Get Started
solace•	Solace JCSMP API	JCSMP	Get Started
solace•	Solace JavaRTO	Java	Get Started
solace•	Solace JMS API	Java	Get Started
solace•	Solace C	C	Get Started
solace•	Solace Python	Python	Get Started
solace•	Solace Go API	Go	Get Started
solace•	Solace .NET	.NET	Get Started
Spring Boot	Spring Boot Java API	Spring	Get Started
Spring Boot	Spring Boot JMS API	Spring	Get Started
Spring Cloud Stream			Get Started

Below this, there's a section for 'Solace Web Messaging' with links for AMQP and MQTT.

On the right side, a sidebar titled 'Solace Java API' shows connection details:

- Username: solace-cloud-client
- Password: (redacted)
- Message VPN: scott-s-demo (highlighted with a red box)
- Host URIs: `tcp://mr-connection-zzdg8xd3fa9.messaging.solace.cloud:55443` (highlighted with a red box)
- TrustStore: Download Root CA PEM, Download Root G2 PEM

\*\*\*\*\*

So, in this case, you will be changing the Host/VPN name to match your instance but the Username should be “user5” and the “Authentication Type” should be Basic and you will use the same name for the Password Secure Alias. “aem-rpp-erp-user5-password” which we will create in a few minutes. Next we will configure the Processing Tab.

Integrations and APIs / AEM-Rapid-Pilot-V2 / SAP\_ERP\_SIM /

## SAP\_ERP\_SIM

Deployment Status: Deployed on Sep 28, 2025, 09:01:46, Runtime Status: Error

Integration Process

```
graph LR; Sender[Sender] -- "AdvancedEventMesh" --> Start((Start)); Start --> JavaScript1[JavaScript 1]; JavaScript1 --> MessageMapping1[Message Mapping 1]; MessageMapping1 --> End((End)); End -- "AdvancedEventMesh" --> Receiver[Receiver]
```

The diagram illustrates the integration process. It starts with a 'Sender' node on the left, connected via an 'AdvancedEventMesh' link to a 'Start' event. This leads to a 'JavaScript 1' step, followed by a 'Message Mapping 1' step, and finally an 'End' event. From the 'End' event, an 'AdvancedEventMesh' link connects to a 'Receiver' node on the right.

**AdvancedEventMesh**

**Connection** (selected)

**SENDER CONNECTION DETAILS**

Host:	<input type="text" value="ENTER YOUR HOST HERE"/>
Message VPN:	<input type="text" value="ENTER YOUR VPN HERE"/>
Username:	<input type="text" value="user5"/>
Authentication Type:	<input type="select" value="Basic"/>
Password Secure Alias:	<input type="text" value="aem-rpp-erp-user5-password"/>

Select the Processing Tab and here, we will be specifying the details for where the iFlow will be sourcing the events from AEM. Please configure your iFlow as per the details below and use “erp.bp\_creation\_approved” as the queue name. The queue will also be provisioning via the Config/Push from the Portal. Also pay particular attention to the “Automatic on Exchange Complete” option for the “**Acknowledgement Mode**”. \*\*\*To be discussed further during the workshop”. Once you have that configured, Hit the Save button and the red associated with that Adapter should disappear.

Integrations and APIs / AEM-Rapid-Pilot-V2 / SAP\_ERP\_SIM /

**SAP\_ERP\_SIM** Deployment Status: Not Deployed

Integration Process

```

graph LR
    Start((Start)) --> JS1[JavaScript 1]
    JS1 --> MM1[Message Mapping 1]
    MM1 --> End((End))
    subgraph AdvancedEventMesh [AdvancedEventMesh]
        direction LR
        Sender[Sender] --- Start
        Start --- JS1
        JS1 --- MM1
        MM1 --- End
        End --- Receiver[Receiver]
    end

```

**AdvancedEventMesh**

**Processing**

SENDER PROCESSING DETAILS

Consumer Mode.*:	Guaranteed
Run on a single worker node?:	<input checked="" type="checkbox"/>
Parallel Consumers.*:	1
Queue Name.*:	erp.bp_creation_approved
Selector:	
Acknowledgment Mode.*:	Automatic On Exchange Complete
Settlement Outcome After Maximum Attempts.*:	Failed
Maximum Message Processing Attempts.*:	5
Retry Interval (in ms).*:	1000
Maximum Retry Interval (in ms).*:	30000
Exponential Backoff Multiplier.*:	3.0

A quick detour to the Javascript component. You won't be changing it but just one thing to point out. You will see that there is some small parsing to take the values from the body of the incoming message and these are used to set a header called "NewDestination". This will be referenced in the next configuration of the AEM Adapter. Also interesting to note that these "Topics" are being created dynamically with no requirement to pre-define them on the AEM instance.



The screenshot shows the AEM script editor interface. At the top, there's a breadcrumb navigation: Integrations and APIs / AEM-Rapid-Pilot-V2 / SAP\_ERP\_SIM / script1.js /. Below the breadcrumb is a blue icon of a document with a pencil. The main area contains the code for script1.js:

```
1 importClass(com.sap.gateway.ip.core.customdev.util.Message);
2
3 function processData(message) {
4
5     // Get body as string
6     var body = message.getBody(java.lang.String);
7
8     // Parse JSON
9     var json = JSON.parse(body);
10
11    // Extract fields
12    var bpType = json.businessPartnerType ? json.businessPartnerType : "";
13    var country = json.country ? json.country : "";
14    var city = json.city ? json.city : "";
15
16    // Concatenate with forward slashes
17    var combined = "sap/businesspartner/created/V1/" + bpType + "/" + country + "/" + city;
18
19    // Store as single header
20    message.setHeader("NewDestination", combined);
21
22    return message;
23 }
```

Now we will configure the Receiver AEM Adapter. Please configure the connection details the same as before as it's exactly the same on the Sender Adapter. Select the "Processing" Tab and modify as below. Note the reference to "\${header.NewDestination}". This is the reference to the header created in the Javascript that we showed above.

Integrations and APIs / AEM-Rapid-Pilot-V2 / SAP\_ERP\_SIM /

**SAP\_ERP\_SIM** Deployment Status: Not Deployed

Integration Process

```

graph LR
    Start((Start)) --> JS1[JavaScript 1]
    JS1 --> MM1[Message Mapping 1]
    MM1 --> End((End))
    subgraph "AdvancedEventMesh"
        direction LR
        Sender[Sender] --> Start
        Start --> JS1
        JS1 --> MM1
        MM1 --> End
        End --> Receiver[Receiver]
        style AdvancedEventMesh fill:none,stroke:none
    end

```

**AdvancedEventMesh**

- General
- Connection
- Processing**
- Message Properties

PUBLISHER PROCESSING DETAILS

Delivery Mode:	* Direct
Endpoint Type:	* Topic
Destination Name:	* \${header.NewDestination}
Message Type:	Automatic

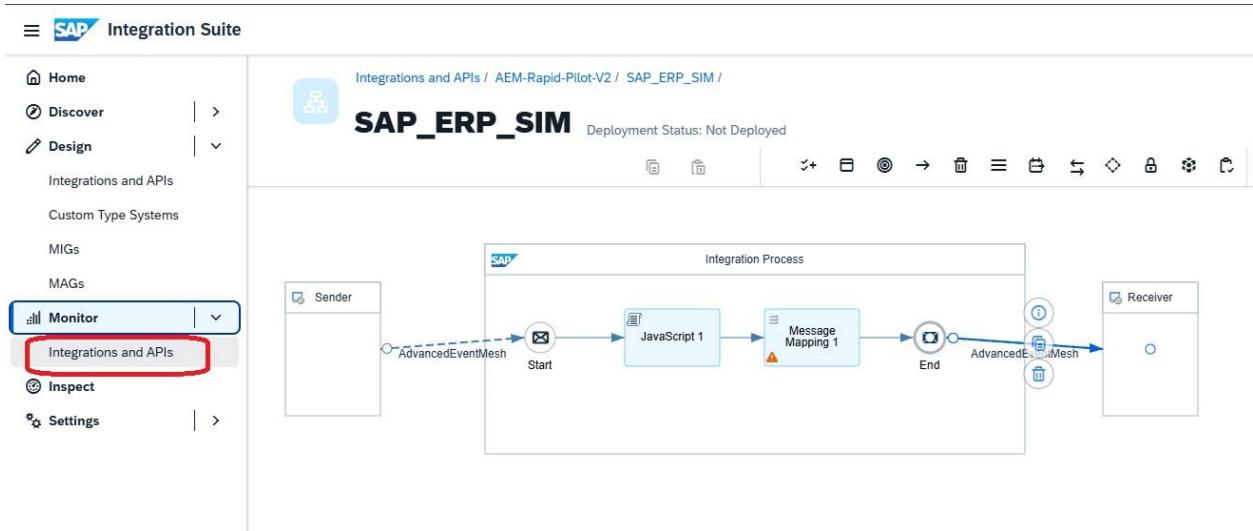
SYNCHRONOUS REQUEST PROCESSING DETAILS

Convert Publish Into Synchronous Requestor?:

Now that we have done all the configuration we need to create the secure parameter as the last step before deploying the iFlow.

Please note the name of the Secure Parameter is “aem-rpp-erp-user5-password”.

Head over to the monitoring area by selecting the “Integrations and APIs” under the Monitor category.



### Select the Security Material Tile

Manage Security

Security Material	Keystore	PGP Keys	Access Policies	JDBC Material	User Roles	Connectivity Tests
0 Artifacts	13 Entities	0 PGP Keys	0 Artifacts			

Manage Stores

Data Stores	Variables	Message Queues	Number Ranges	Partner Directory
0 Stores	0 Variables	0 Queues	0 Artifacts	

Access Logs

Manage Locks

Message Locks	Designtime Artifact Locks
0 Locks	0 Locks

Usage Details

On the far right, select the “Create” and choose “Secure Parameter” from the dropdown menu

Overview / Manage Security Material

Security Material (0)

Name	Type	Status	Deployed By	Deployed On
No data available				

Create

Here you will use the UserID as specified above and the password will be “user123\$”...of course you can change the password if you want but these will be pushed from Event Portal so for simplicity, we suggest you keep it as specified 😊 . Hit the “Deploy” button to activate the artifact.

The dialog box has a title 'Create Secure Parameter'. It contains four input fields: 'Name:' with value 'aem-rpp-erp-user4-password', 'Description:' (empty), 'Secure Parameter:' with value '\*\*\*\*\*', and 'Repeat Secure Parameter:' with value '\*\*\*\*\*'. At the bottom are 'Deploy' and 'Cancel' buttons.

At this point, you will likely have 3-4 other secure parameters. Just ensure that you see the status of “Deployed”.

Security Material (1)			
Name	Type	Status	Deployed By
aem-rpp-erp-user5-password	Secure Parameter	Deployed	scottdillon1@gmail.com

Now we are ready to deploy the iFlow.

Return back to the Design Area by selecting “Integrations and APIS” and then select your package.

The screenshot shows the SAP Integration Suite interface. On the left, there is a navigation sidebar with the following items:

- Home
- Discover
- Design
- Integrations and APIs** (this item is highlighted with a red rectangle)
- Custom Type Systems
- MIGs
- MAGs
- Monitor
- Integrations and APIs
- Inspect
- Settings

The main content area is titled "Integrations and APIs / Design". It displays a list of "Packages (2)":

- Advanced Event Mesh Adapter for SAP Integration Suite
- AEM-Rapid-Pilot-V2

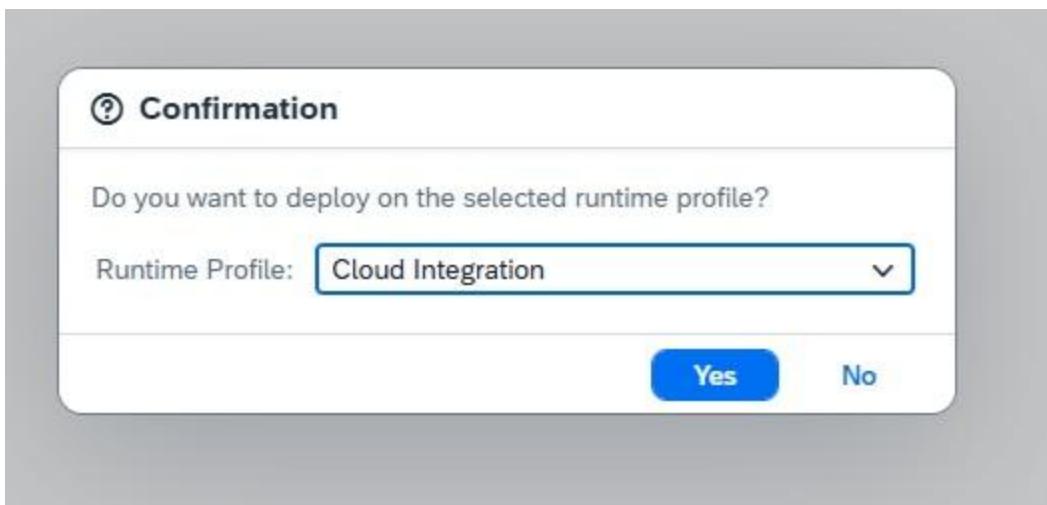
Once in the package, we will deploy the Message Mapping first. Select the ERPSIM artifact and Select the deploy option.

The screenshot shows the details page for the "AEM-Rapid-Pilot-V2" package. At the top, it says "AEM Rapid Pilot Program Version 2" and provides vendor and mode information. Below this, there are tabs for "Overview", **Artifacts (2)**, "Documents", and "Tags".

The "Artifacts (2)" tab is selected, showing a list of artifacts:

Name	Type	Version	Actions
ERPSIM	Message Mapping	1.0.2	<a href="#">View metadata</a> <a href="#">Download</a> <a href="#">Deploy</a> <a href="#">...</a>
SAP_ERP_SIM	Integration Flow	Draft	<a href="#">View metadata</a> <a href="#">Download</a> <a href="#">Deploy</a> <a href="#">...</a>

Then you will select “Cloud Integration” and press Yes.



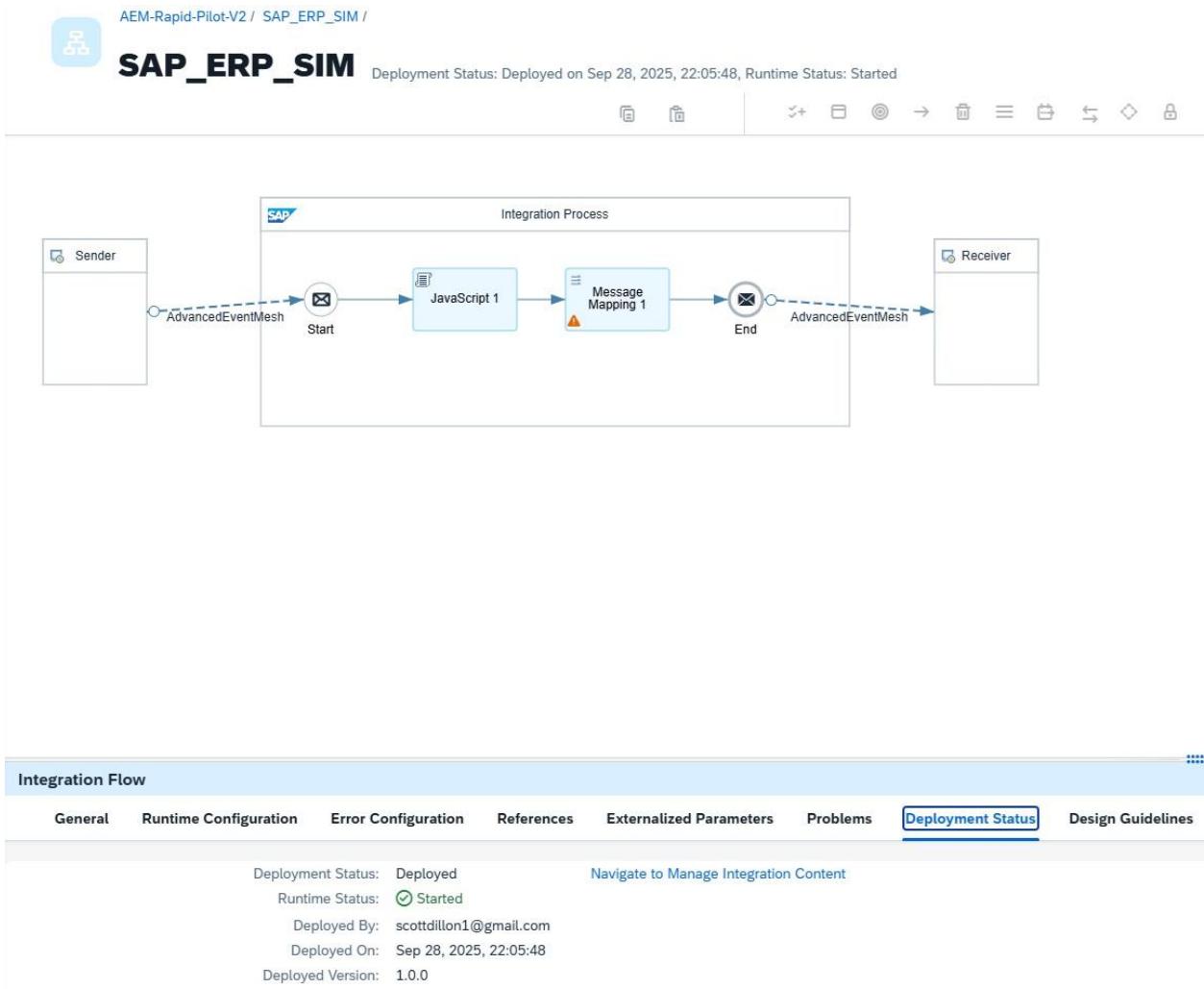
Once the message mapping is deployed, you can repeat the process for the iFlow called "SAP\_ERP\_SIM".

The screenshot shows the SAP Integration Suite interface. On the left, there's a navigation sidebar with options like Home, Discover, Design, Integrations and APIs, Monitor, Inspect, and Settings. The "Integrations and APIs" option is currently selected. In the main area, there's a card for "AEM-Rapid-Pilot-V2" which is described as "AEM Rapid Pilot Program Version 2". It shows vendor information and mode as "Editable". Below this, there are tabs for Overview, Artifacts (2), Documents, and Tags. The Artifacts tab is selected, displaying a table of artifacts. The table has columns for Name, Type, Version, and Actions. The artifacts listed are:

Name	Type	Version	Actions
ERPSIM	Message Mapping	1.0.2	<a href="#">Copy</a> <a href="#">View metadata</a> <a href="#">Download</a> <a href="#">Configure</a> <a href="#">Deploy</a>
SAP_ERP_SIM	Integration Flow	Draft	<a href="#">Copy</a> <a href="#">View metadata</a> <a href="#">Download</a> <a href="#">Configure</a> <a href="#">Deploy</a>

Below the table, there's another confirmation dialog box identical to the one at the top of the page.

Once the iFlow is deployed, you can double check the Status by clicking on the iFlow and then navigating to the "Deployment Status" Tab.



Now that we have the iFlow deployed, we should use the TRY ME Tab within the broker to test the conversion and generation of the new Topic.

## Testing the iFlow

Navigate to the TRY-ME tab within the AEM Service. From the Cluster Manager, you will select your broker and then Navigate to the Try Me Tab located at the Top of the Screen.

The screenshot shows the SAP Cluster Manager interface. On the left, there is a vertical sidebar with icons for Designer, Catalog, Runtime Manager, KPI, Micro-Integrations, and Cluster Manager. The Cluster Manager icon is highlighted. At the top, there is a dropdown menu set to 'Default' and a 'Cluster Manager' button. Below this, the title 'Cluster Manager: Services' is displayed. A search bar with the placeholder 'Filter by service name' and a magnifying glass icon is present. To its right is a checkbox labeled 'Only show my services'. The main area is titled 'All services' and contains two service entries:

- AEM\_Broker** (aws icon) - eks-ca-central-1a. Status: Running. Includes a user icon and a refresh icon.
- Broker 100** (Broker icon) - niketh.kala@solace.com kala. Status: Running. Includes a user icon and a refresh icon.

Once you are here, you will select the “open Broker Manager” and another screen will launch in a different browser tab.

The screenshot shows the 'Service Details: AEM\_Broker' page. The top navigation bar includes 'Default', 'Cluster Manager > Service Details', and tabs for Status, Connect, Manage, Monitoring, Configuration, and Try Me!. The 'Try Me!' tab is currently selected. The left sidebar has icons for Designer, Catalog, Runtime Manager, KPI, Micro-Integrations, Cluster Manager (highlighted), Mesh Manager, and Insights. The main content area starts with a 'Get Started with Try Me!' section. Below it is a 'Connect your event broker!' section with instructions about hierarchical topics and wildcards. It also says to copy credentials and follow three steps on the right. There are input fields for 'Client Username' (solace-cloud-client) and 'Client Password' (redacted). A 'Paste' button with a '2' is next to the password field. A 'Try Me!' button is located at the bottom of this section. To the right is a 'Broker Manager' panel with sections for 'Publisher' (Establish Connection, Broker URL [pre-filled for you], Message VPN [pre-filled for you], Client Username, Client Password), 'Connect' (Expand, Connect button with a '1'), and 'Publish' (Topic selected, Queue option, Publish button, Show Advanced). Below the Broker Manager is a 'Think Events!' section with a 'Learn more about best practices' link. At the bottom is a 'Topic Structure' section with the text 'noun / verb / property / property / property'.

On the left side menu, you will see the TRY ME option select it to reveal an embedded “Publisher” and “Subscriber” directly on the broker so you can try various scenario.

The screenshot shows the SAP AEM Broker interface. On the left, there's a navigation sidebar with sections like AEM\_Broker, Messaging, System, and Try Me!. The main area has two tabs: "Send and Receive". The "Publisher" tab is selected, showing fields for "Establish Connection" (Not Connected), "Publish" (Topic selected), "Learn about topic taxonomy", "Think Events!", "Create a topic using the topic builder" (with fields for Noun, Verb, Property, and a "Publish" button), "Delivery Mode" (Direct selected), and "Message Content" (a JSON payload example). The "Subscriber" tab shows fields for "Establish Connection" (Not Connected), "Subscribe" (How do you want to receive messages, Subscribe to a topic to receive direct messages), "Tune into Events!" (Wildcards: noun / verb / property / property / property), "Suggestions" (flight / >), "Topic Subscriber" (flight/), and "Bind to an endpoint to receive guaranteed messages" (Messages (Most Recent 20)).

Connect to your broker as per the Day 1 exercise and now we will use the “publisher” tab to send in the Event that the iFlow is listening for and we can validate that it is generating the correct topic.

The payload you can use would look like this:

```
{
  "partnerId": "P123456789",
  "validFrom": "2025-09-28T08:00:00Z",
  "validTo": "2026-09-28T08:00:00Z",
  "addressNumber": "A987654321",
  "businessPartnerType": "I",
  "firstName": "John",
  "lastName": "Doe",
  "city": "New York",
  "postalCode": "10001",
  "street": "5th Avenue",
```

```

"houseNumber": "101",
"country": "USA"
}

```

Of course, this is easy to validate by referring to the Event Portal Design and checking our model.

So, if everything is working properly at this point, you can publish the event as shown below with the topic “sap/businesspartner/creationApproved/V1/1/CANADA/Ottawa” and you should see the payload appear on the right with the topic shown “sap/businesspartner/created/V1/I/USA/New York”.

2 Things:

- 1) I just noticed that Approved is spelt wrong 😊
- 2) Why does the topic have a different country/city in the transformed topic?

Publisher

Establish Connection  
% Connected

Topic  Queue

Learn about topic taxonomy

Think Events!

Build your smart topic to describe an event matching your use cases! Learn more about best practices

Topic Structure  
noun / verb / property / property / property

Create a topic using the topic builder

Noun	Verb	Property	Property	Property
flight	/ boarding	/ #1234	/ yow	/ ewr

Publish to topic  
**sap/businesspartner/creationApproved/V1/1/CANADA/Ottawa**

Delivery Mode  
 Direct  Persistent

Message Content

```
{
  "partnerId": "P123456789",
  "validFrom": "2025-09-28T08:00:00Z",
  "validTo": "2026-09-28T08:00:00Z",
  "addressNumber": "A987654321",
  "businessPartnerType": "I"
}
```

**Subscriber**

Subscribed Topics

Bind to an endpoint to receive guaranteed messages

Messages (Most Recent 20)

3 Direct 0 Persistent 0 Non-Persistent

2025-09-28 23.23.50.044 [Topic sap/businesspartner/created/V1/I/USA/New York]

Delivery Mode: Direct  
DMO Eligible: Yes  
Priority: 4

```
{
  "partnerId": "P123456789",
  "validFrom": "2025-09-28T08:00:00Z",
  "validTo": "2026-09-28T08:00:00Z",
  "addressNumber": "A987654321",
  "businessPartnerType": "I",
  "firstName": "John",
  "lastName": "Doe",
  "city": "New York",
  "postalCode": "10001"
}
```

2025-09-28 23.23.49.605 [Topic sap/businesspartner/creationApproved/V1/1/CANADA/Ottawa]

Delivery Mode: Direct  
Sender id: Try-Me-Pub/solclient[chrome-140.0.0-Windows-0.0.0/0792899412/0001  
Sender Timestamp: 2025-09-28 23.23.49.586

```
{
  "partnerId": "P123456789",
  "validFrom": "2025-09-28T08:00:00Z",
  "validTo": "2026-09-28T08:00:00Z",
  "addressNumber": "A987654321",
  "businessPartnerType": "I",
  "firstName": "John",
  "lastName": "Doe",
  "city": "New York",
  "postalCode": "10001"
}
```

If you made it this far...you rock 😊

