

# Day 3 – Enhanced Integration

## Exercises for SAP Integration Suite

### Contents

Goals .....	2
Prerequisites .....	2
Prepare your Integration Suite tenant.....	3
<b>Download and import the template IFlow package</b> .....	3
<b>Importing the official SAP Advanced Event Mesh Adapter</b> .....	3
Scenario 1: “From Scratch” .....	5
<b>Queue Setup</b> .....	5
<b>IFlow Creation</b> .....	10
Scenario 2: “Dead Message Queue” .....	14
Automated AEM Broker Setup via APIs and CI/CD (optional) .....	19
Scenario 3: “Sales Order E-Mail Notification” .....	21
<b>Setup/Configure Dependency Services</b> .....	21
<b>Queue Setup</b> .....	21
<b>IFlow Setup</b> .....	22
Scenario 4: “Business Partner Address Check” (optional) .....	24
<b>Setup/Configure Dependency Services</b> .....	24
<b>Queue Setup</b> .....	25
<b>IFlow Setup</b> .....	26
Scenario 5: “SFTP as Legacy Integration” (optional).....	29
<b>Setup/Configure Dependency Services</b> .....	29
<b>Queue Setup</b> .....	29
<b>IFlow Setup</b> .....	30
Security and fine-grained access control to topics (optional) .....	33
<b>Experimenting publishing and subscribing to protected topics</b> .....	33
<b>Broker Topic ACLs</b> .....	34


## Goals

- Configuring an AEM brokers' queues and topic subscriptions
- Event enabling integration flows and connecting them to AEM brokers to create event-driven integration flows
- How to use the broker's config APIs to automate configuration and enable CI/CD pipelines
- Fine-grained security access in AEM

## Prerequisites


- Complete all activities in day 1 & 2  
You access and use the same broker you setup previously as well as the simulator to push events for testing
- Have access to an active Integration Suite/Cloud Integration tenant
- Have an SFTP server and account credentials if you want to test successful integration of events to a file based interface of a legacy system (optional)

- [Discover \(Integrations\) /](#)



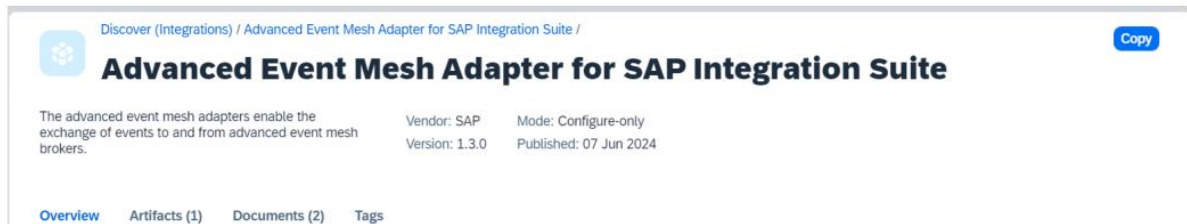
# Discover (605)

## 22 package(s) found

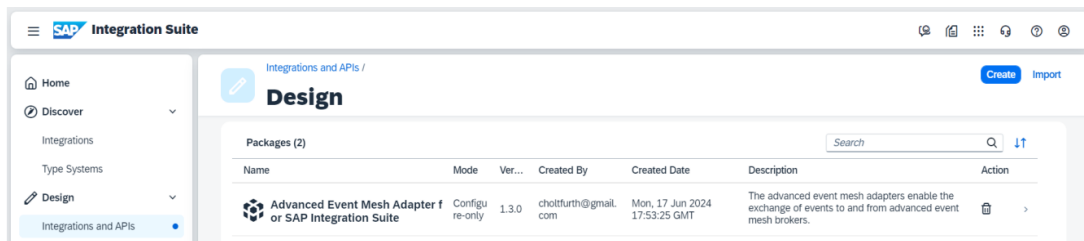


Advanced Event Mesh Adapter for SAP Integration Suite

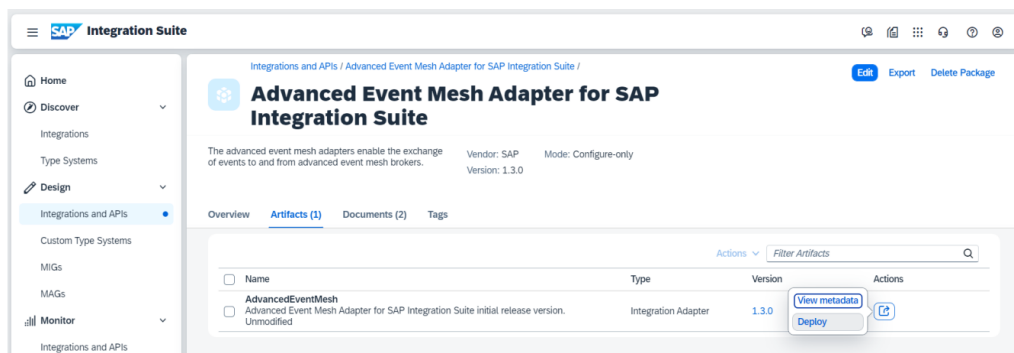
The **advanced** event mesh adapters enable the exchange of events to and from **advanced** event mesh brokers



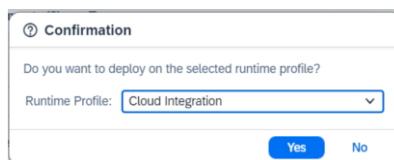
- Go to “Design” -> “Integrations and APIs” in the left hand menu and click on the newly created “Advanced Event Mesh Adapter for SAP Integration Suite” package



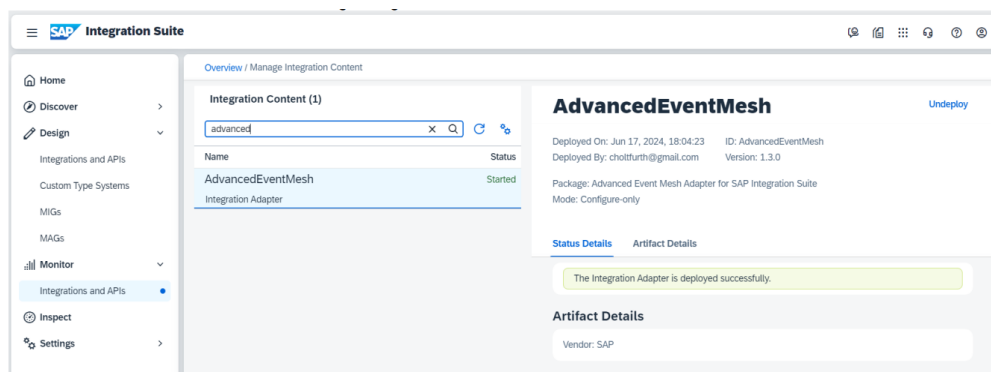
- Navigate to “Artifacts” to see the Integration Adapter and click on and select “Deploy” from the “Actions” menu



- Select the “Cloud Integration” Runtime Profile



- You should now be able to see the AdvancedEventMesh Integration Adapter if you navigate to “Monitor” -> “Integrations and APIs” and click on the tile “All” under “Manage Integration Content”



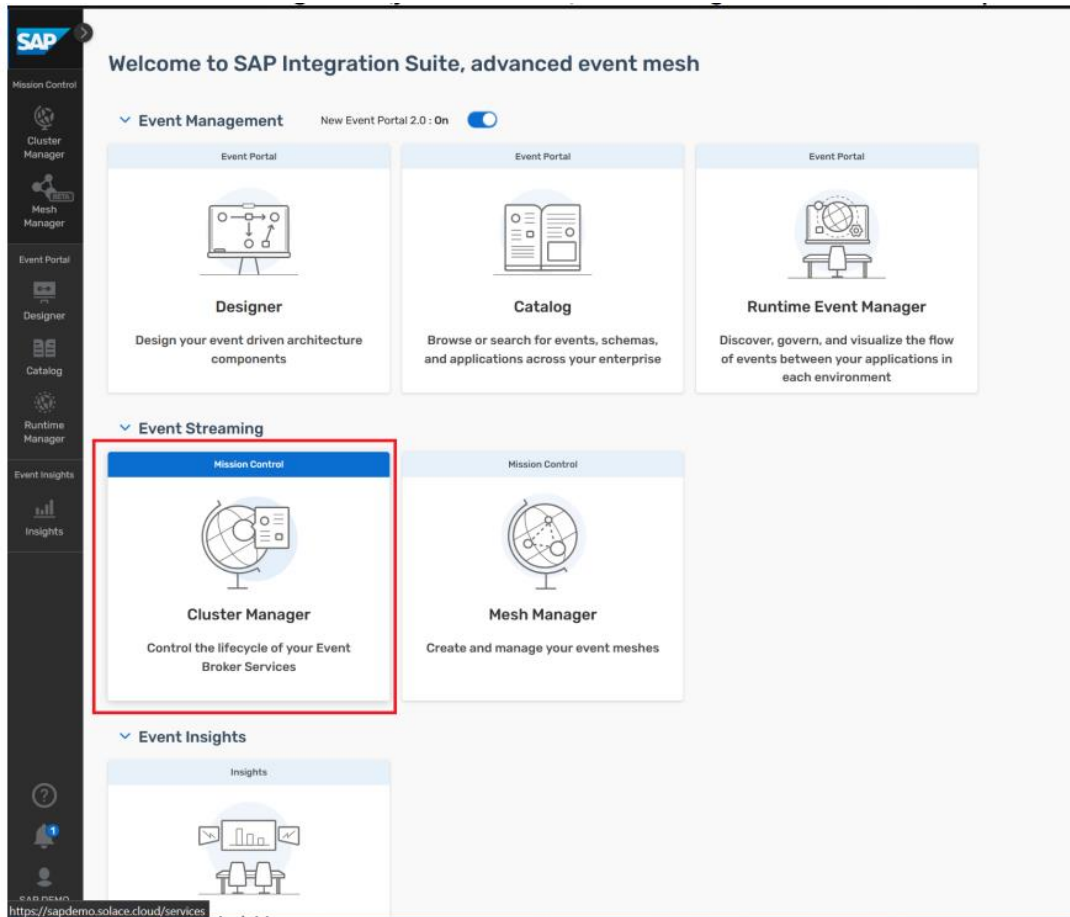
## Scenario 1: “From Scratch”

In this section we will perform the necessary steps to create an event driven IFlow from scratch

### Queue Setup

First, we will create the required input queue for the integration flow.

- Go to Cluster Manager -> {your service} -> Manage -> Queues - to open the Broker UI



# Services

Create Service

☐ Only show my services

All Services (14)

**eu1**  
SAP DEMO (eu-central-1)  
Enterprise 250 Class  
Christian Holtfurth  
Running

**us1**  
GKE - US Central (Iowa)  
Enterprise 250 Class  
Christian Holtfurth  
Running

**ap1**  
EKS - Asia Pacific (Singapore)  
Enterprise 250 Class  
Christian Holtfurth  
Running

**MontrealBroker-10.1**  
EKS - Canada Central(Montreal)  
Developer Class  
Scott Dillon  
Running

**US-Central**  
AKS - Central US (Iowa)  
Developer Class  
Karl Ossolnig  
Running

**sa1**  
EKS - Africa (Cape Town)  
Enterprise 250 Class  
Christian Holtfurth  
Running

**cn1**  
AKS - East Asia (Hong Kong)  
Enterprise 250 Class  
Christian Holtfurth  
Running

**BTP Hackathon September**  
EKS - Canada Central(Montreal)  
Developer Class  
brad.caldwell@solace.com  
Running

**MyMesh-Svc1**  
AKS - East US 2 (Virginia)  
Developer Class  
Karl Ossolnig  
Running

**MyMesh-Svc2**  
EKS - US West (Oregon)  
Developer Class  
Karl Ossolnig  
Running

**My-First-Service**  
EKS - Canada Central(Montreal)  
Enterprise 250 Class  
Andrea Kelso  
Running

**test**  
GKE - Asia South (Mumbai)  
Enterprise 1K Class  
Christian Holtfurth  
Running

**testbrokersap1**  
GKE - EU(London)  
Enterprise 1K Class  
Christian Holtfurth  
Running

**Karls-special-broker-just-for-...**  
AKS - Canada Central (Toronto)  
Enterprise 5K Class  
Karl Ossolnig  
Running

+

MontrealBroker-10.1

Open Broker Manager

Status

Connect

Manage

Monitoring

Configuration

Try Me!

Event Broker Service Settings

Deletion Protection

Delete Service

Advanced Options

Authentication  
Enabled

Certificate Authorities  
0 Client Certificate Authorities  
1 Domain Certificate Authority

Client Profiles  
1 Client Profile

Broker Manager Quick Settings

Message VPN

Clients

Queues

Access Control

Bridges

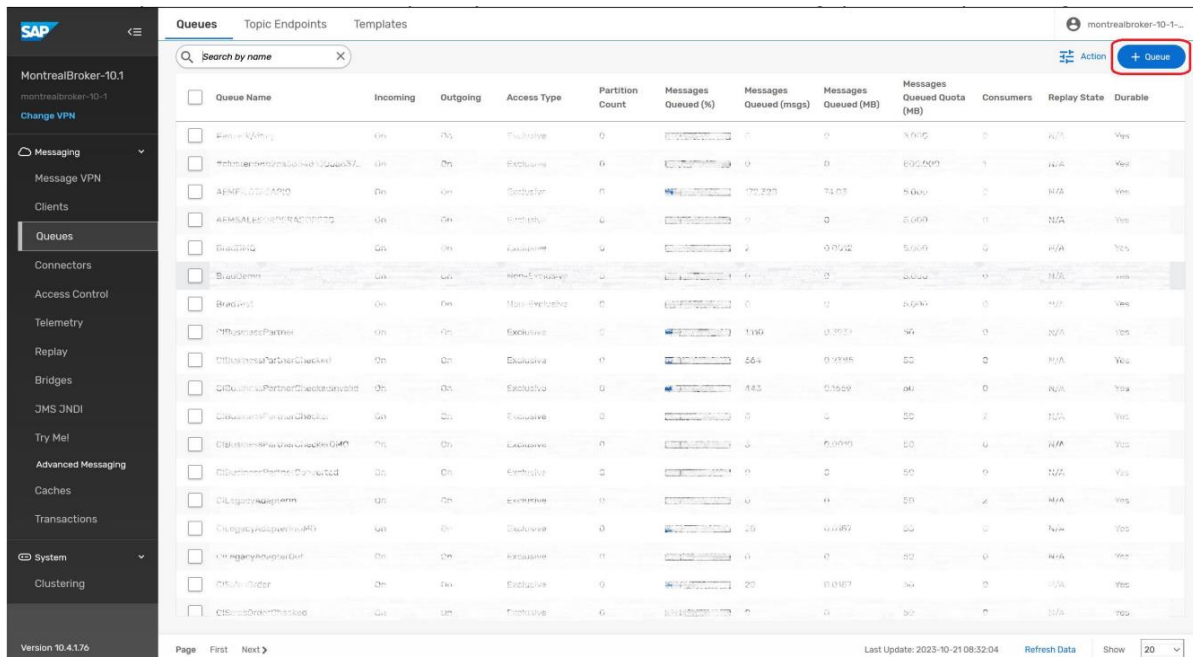
Other Management Tools

> SEMP - REST API  
The Solace Element Management Protocol (SEMP) is a REST API that you can use to manage the Event Broker Service.

> Broker Manager - Web Application  
The Broker Manager is a browser-based administration console that you can use to manage the Event Broker Service.

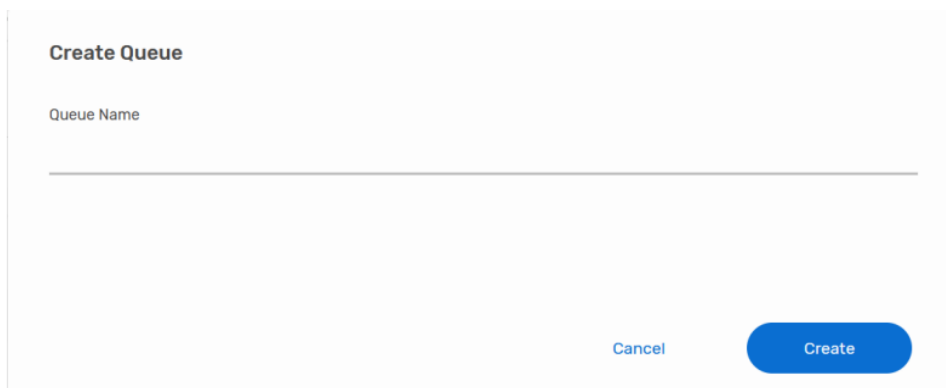
> SolAdmin - Desktop Application  
SolAdmin is a legacy desktop application that you can use to manage the Event Broker Service.

- To create a queue click the “+ Queue” button to bring up the create queue dialog



Queue Name	Incoming	Outgoing	Access Type	Partition Count	Messages Queued (%)	Messages Queued (msgs)	Messages Queued (MB)	Messages Queued Quota (MB)	Consumers	Replay State	Durable
QueueName1	On	On	Exclusive	0	0	0	0	5000	0	N/A	Yes
QueueName2	On	On	Exclusive	0	0	0	0	5000	0	N/A	Yes
QueueName3	On	On	Exclusive	0	0	0	0	5000	0	N/A	Yes
QueueName4	On	On	Exclusive	0	0	0	0	5000	0	N/A	Yes
QueueName5	On	On	Exclusive	0	0	0	0	5000	0	N/A	Yes
QueueName6	On	On	Exclusive	0	0	0	0	5000	0	N/A	Yes
QueueName7	On	On	Exclusive	0	0	0	0	5000	0	N/A	Yes
QueueName8	On	On	Exclusive	0	0	0	0	5000	0	N/A	Yes
QueueName9	On	On	Exclusive	0	0	0	0	5000	0	N/A	Yes
QueueName10	On	On	Exclusive	0	0	0	0	5000	0	N/A	Yes
QueueName11	On	On	Exclusive	0	0	0	0	5000	0	N/A	Yes
QueueName12	On	On	Exclusive	0	0	0	0	5000	0	N/A	Yes
QueueName13	On	On	Exclusive	0	0	0	0	5000	0	N/A	Yes
QueueName14	On	On	Exclusive	0	0	0	0	5000	0	N/A	Yes
QueueName15	On	On	Exclusive	0	0	0	0	5000	0	N/A	Yes
QueueName16	On	On	Exclusive	0	0	0	0	5000	0	N/A	Yes
QueueName17	On	On	Exclusive	0	0	0	0	5000	0	N/A	Yes
QueueName18	On	On	Exclusive	0	0	0	0	5000	0	N/A	Yes
QueueName19	On	On	Exclusive	0	0	0	0	5000	0	N/A	Yes
QueueName20	On	On	Exclusive	0	0	0	0	5000	0	N/A	Yes

- Provide a name for this queue or decide on your own  
! – For the sake of compatibility with the Sales Order Dashboard from Day 1 the name should be chosen as *CILegacyAdapterIn*

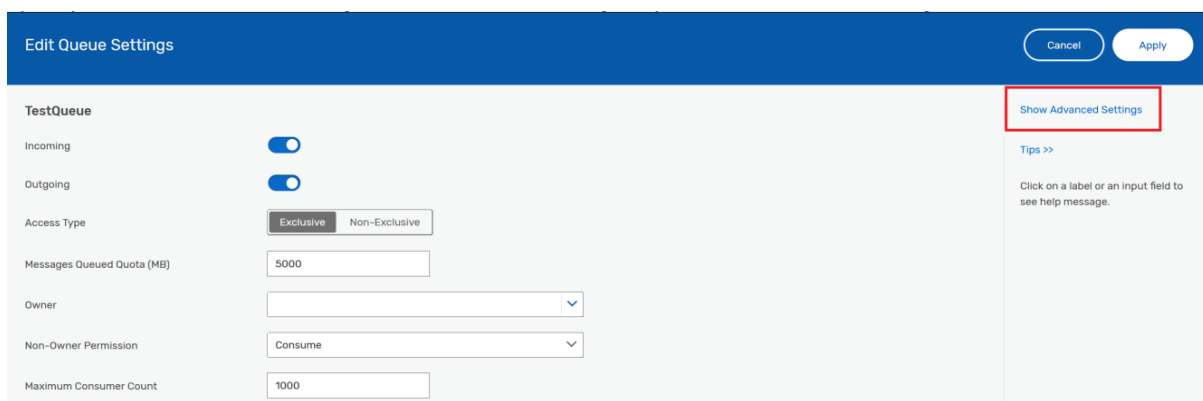


**Create Queue**

Queue Name

Cancel Create

- Open up the "Advanced Queue Settings" section, then follow along and provide the details as showing in the screenshots below



**Edit Queue Settings**

TestQueue

Incoming: ☒

Outgoing: ☒

Access Type: ☒ Exclusive ☐ Non-Exclusive

Messages Queued Quota (MB):

Owner:

Non-Owner Permission:

Maximum Consumer Count:

Cancel Apply

Show Advanced Settings

Tips >>

Click on a label or an input field to see help message.

**SAP** < Queues | CILegacyAdapterIn

Summary **Settings** Subscriptions Consumers Messages Queued Stats

MontrealBroker-10.1  
montrealbroker-10-1  
[Change VPN](#)

Messaging

- Message VPN
- Clients
- Queues**
- Connectors
- Access Control
- Telemetry
- Replay
- Bridges
- JMS JNDI
- Try Me!
- Advanced Messaging
- Caches
- Transactions

System

- Clustering

Incoming ☒

Outgoing ☒

Access Type ☒ Exclusive ☐ Non-Exclusive

Messages Queued Quota (MB)

Alert Thresholds  Clear  Raise

Owner

Non-Owner Permission

Maximum Consumer Count

Alert Thresholds  Clear  Raise

Maximum Message Size (B)

Maximum Delivered Unacknowledged Messages per Flow

DMQ Name

Enable Client Delivery Count ☐

Delivery Delay (sec)

Message Priority

Respect Message Priority ☐

**SAP** < Queues | CILegacyAdapterIn

Summary **Settings** Subscriptions Consumers Messages Queued Stats

MontrealBroker-10.1  
montrealbroker-10-1  
[Change VPN](#)

Messaging

- Message VPN
- Clients
- Queues**
- Connectors
- Access Control
- Telemetry
- Replay
- Bridges
- JMS JNDI
- Try Me!
- Advanced Messaging
- Caches
- Transactions

System

- Clustering

Message Expiry

Respect TTL ☐

Maximum TTL (sec)

Redelivery ☒

Try Forever ☐

Maximum Redelivery Count

Delayed Redelivery ☐

Multiplier

Initial Delay

Maximum Delay

Congestion Control

Reject Messages to Sender on Discard

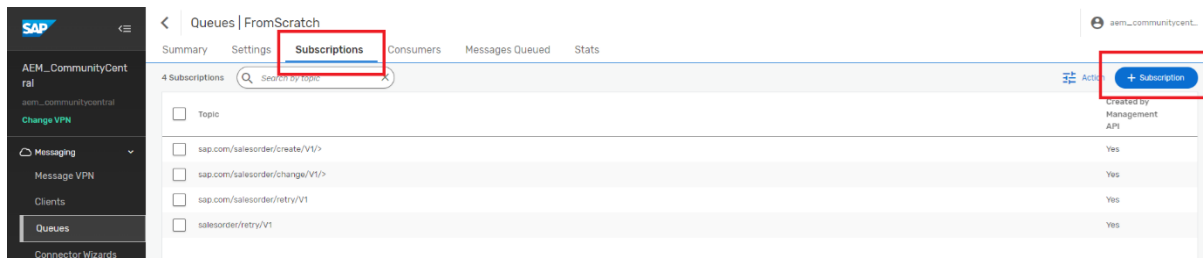
Reject Low Priority Messages ☐

Reject Low Priority Messages Limit

Alert Thresholds  Clear  Raise

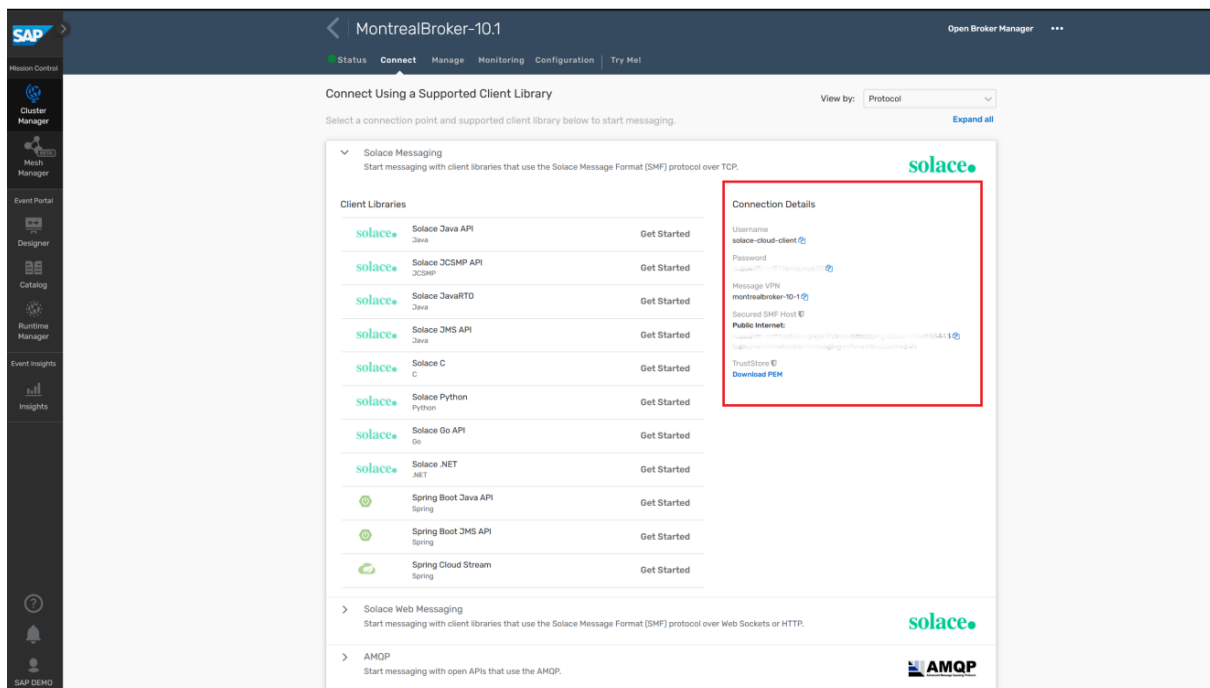
- Once the queue is created, click on the queue name in the list, navigate to the Subscriptions tab and open the subscriptions dialog.





- Add the following subscriptions to the queue
  1. sap.com/salesorder/create/V1/>
  2. sap.com/salesorder/change/V1/>
  3. sap.com/salesorder/retry/V1
  4. salesorder/retry/V1

Now, before we jump into Integration Suite: Let's head to our Advanced Event Mesh Console and go to Cluster Manager, select the service that you want to connect your Integration Suite flows to and go to the "Connect" tab. Take a note of the connectivity details underneath "Solace Messaging" (click on the section to open it up):

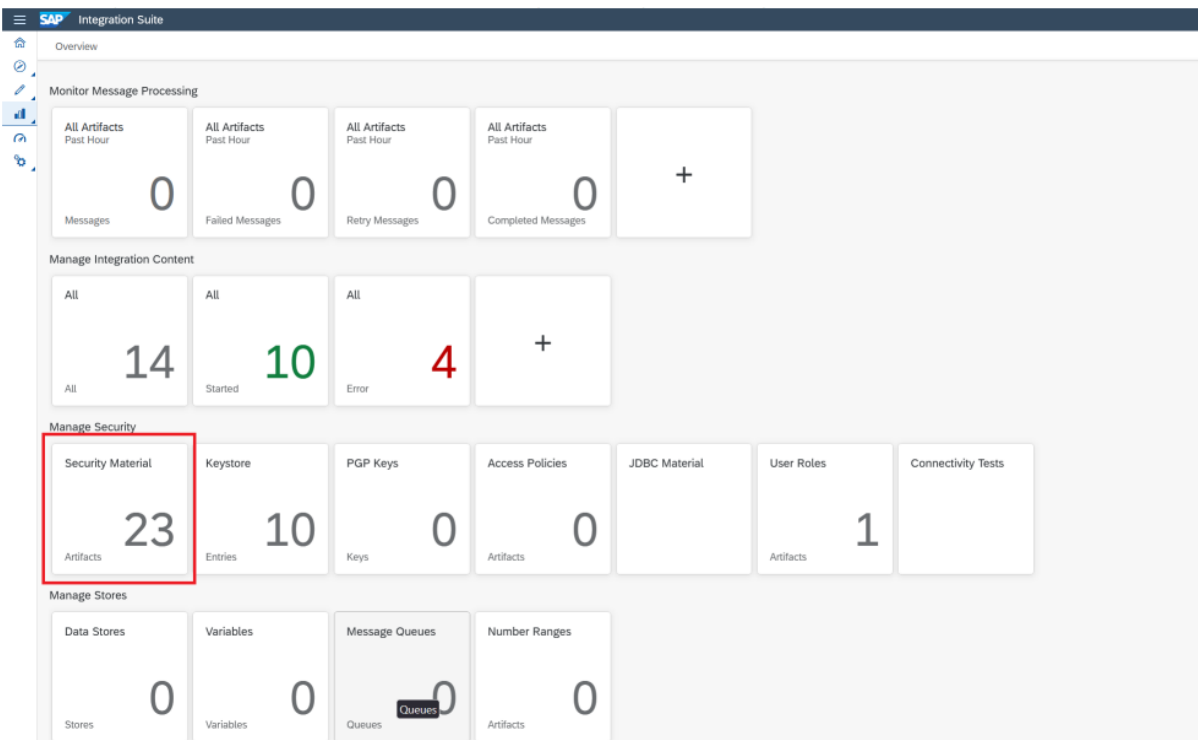


The connect tab lists all the various connectivity details for the various supported protocols. The AEM adapter uses the Solace Messaging protocol, which is AEMs very own protocol with a broad feature support. Each AEM service also comes with a default client user called *solace-cloud-client* that is configured for convenience reasons and is allowed to publish and subscribe to all topics. We will be using this user for all our IFlows. In a real production environment where security is important, you or your administrator will likely have this user disabled and will be creating separate users for each of the applications that connect to the AEM broker. Or this may even be deferred to an external authentication service over LDAP or OAuth with no client users stored on the broker itself and managed by your IAM service instead.

## IFlow Creation

The Integration Suite stored security details such as username/password pairs in so called Security Material. First, we configure security details for Basic Authentication against the Advanced Event Mesh.

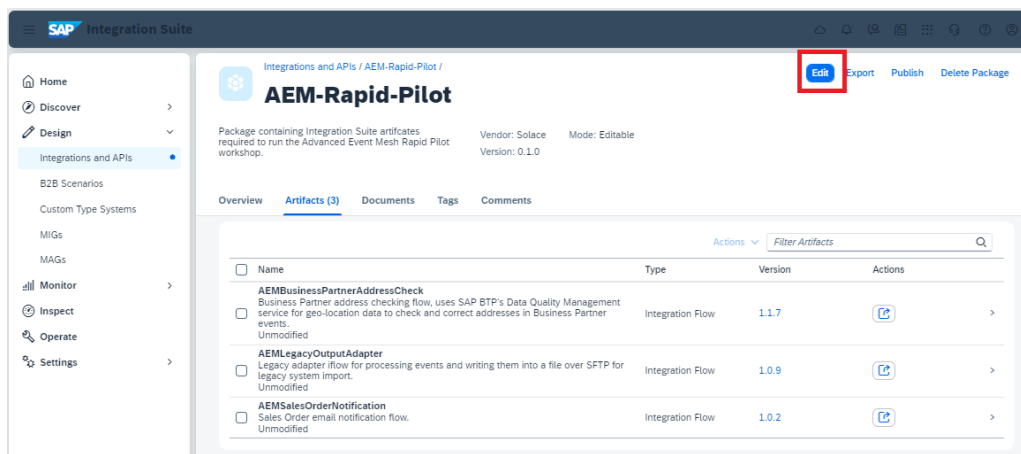
- Go to Integration Suite Monitor Artifacts -> Manage Security -> Security Material



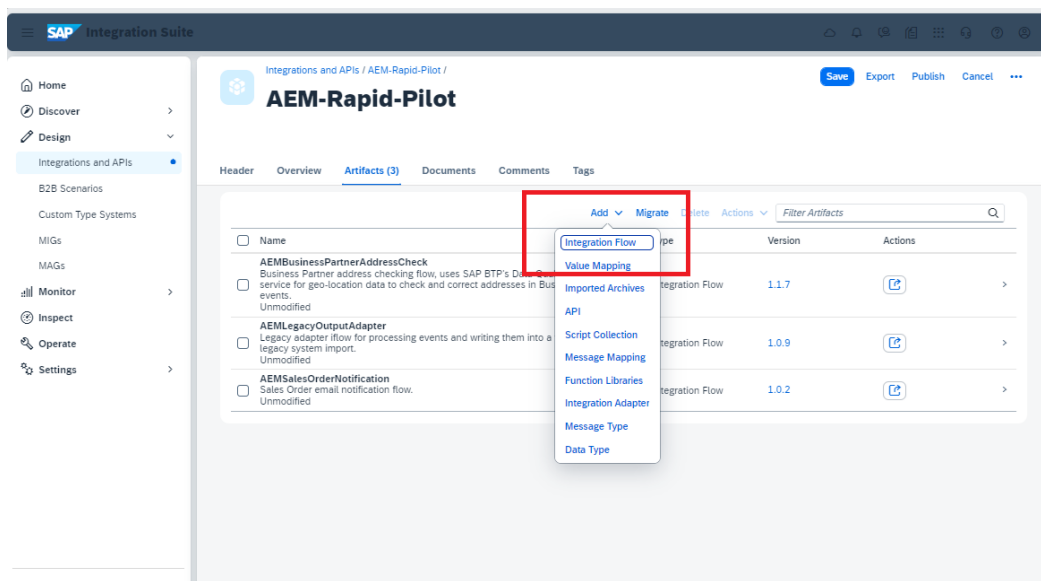
- Create Secure Parameter CABrokerUserPass and store the password for your solace-cloud-client application user credentials. We use this name since it will be the default used by the other provided IFlow content, but feel free to adhere to other naming conventions for this scenario.

Next, we will create a new IFlow and configure it with the minimum configuration to start receiving events.

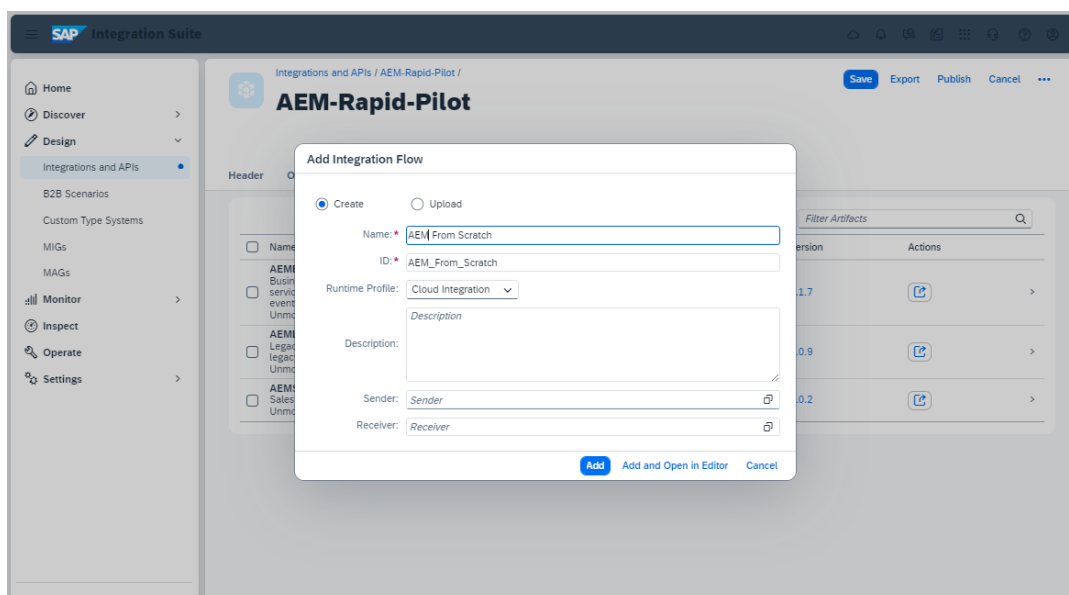
- Navigate to the imported package in the Design section of the Integration Suite



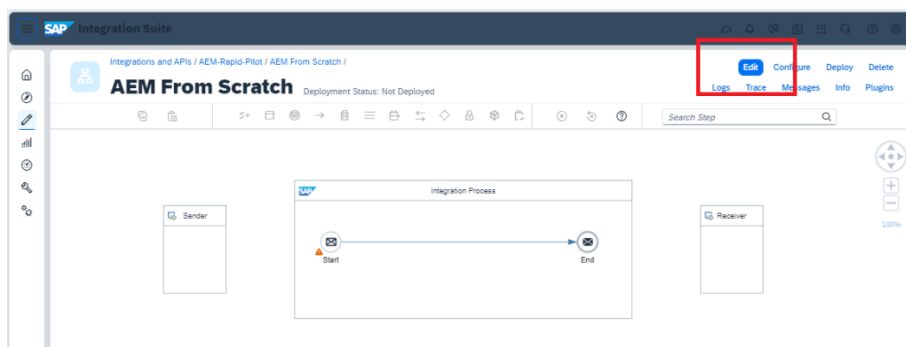
- To create a new IFlow first edit the Integration Package



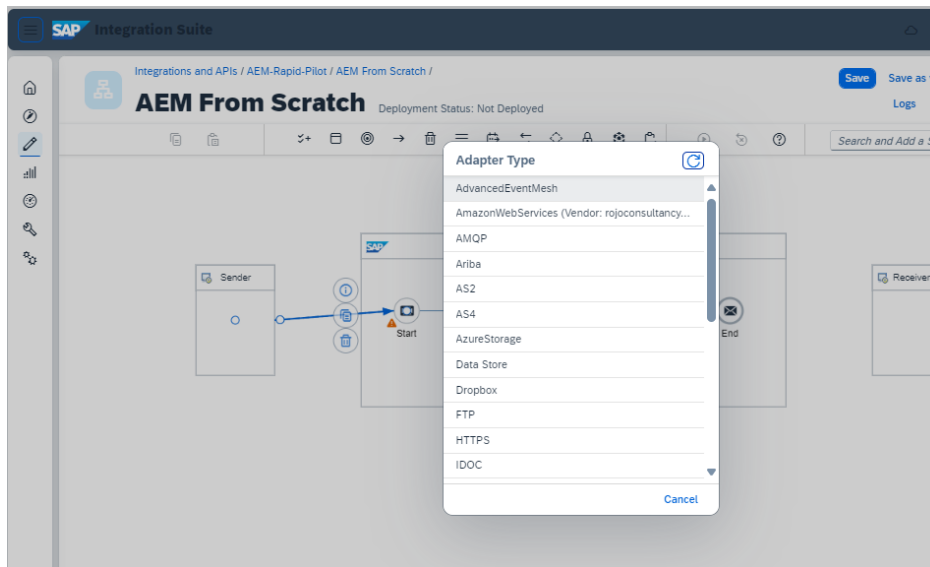
- The name and further details of this IFlow can be chosen however you like. In case you already have established Integration Suite guidelines within your landscape feel free to adhere to them



- Click "Add and Open in Editor", you will be shown a completely empty IFlow template which we want to edit



- Start by clicking the “Sender” and drag a connection to the “Start” of you Integration Process. Choose the AdvancedEventMesh Adapter



- The bare minimum that needs to be configured to enable the IFlow to be triggered by events is the following (replace the Host and Message VPN with the information you found in your broker at the beginning of these exercises and the name of the queue with yours).

**AdvancedEventMesh**

General **Connection** Processing

SENDER CONNECTION DETAILS

Host: \*

Message VPN: \*

Username: \*

Authentication Type:

Password Secure Alias: \*

JCSMP Properties:

☐ Key

- Take care when configuring the Processing details. The next exercise will go into more detail on these settings.

**AdvancedEventMesh**

General Connection **Processing**

SENDER PROCESSING DETAILS

Consumer Mode: \* Guaranteed ▼

Parallel Consumers: \* 1

Queue Name: \* CILegacyAdapterIn

Selector:

Acknowledgment Mode: \* Automatic On Exchange Complete ▼

Settlement Outcome After Maximum Attempts: \* Failed ▼

Maximum Message Processing Attempts: \* 2

Retry Interval (in ms): \* 500

Maximum Retry Interval (in ms): \* 500

Exponential Backoff Multiplier: \* 1.0

- Save your configured IFlow and hit deploy at the top right
- Go to Monitor Integrations and APIs -> Manage Integration Content -> All  
If you did everything correctly you should find your IFlow with Status “Started”

**SAP Integration Suite**

Overview / Manage Integration Content

Runtime: Cloud Integration ▼

Integration Content (621)

Filter by Name or ID

Name	Status
AEM From Scratch Integration Flow	Started

**AEM From Scratch**

Deployed On: Aug 16, 2024, 13:21:01 ID: AEM\_From\_Scratch Package: AEM-Rapid-Pilot  
Deployed By: erik.heinrich@sap.com Version: 1.0.0

Endpoints Status Details Artifact Details Log Configuration

There are no endpoints configured.

- Navigate back to the queues in your AEM Broker, check that the queue you used for this exercise now has at least one consumer connected to it

SAP

<=>

MontrealBroker-10.1

montrealbroker-10-1

Change VPN

Messaging

Message VPN

Clients

Queues

Connectors

Access Control

Telemetry

Replay

Bridges

JMS JNDI

Try Mail

Advanced Messaging

Caches

Transactions

System

Clustering

Queues

Topic Endpoints

Templates

Q ci

X

montrealbroker-10-1

Action

+ Queue

<input type="checkbox"/>	Queue Name	Incoming	Outgoing	Access Type	Partition Count	Messages Queued (%)	Messages Queued (msgs)	Messages Queued (MB)	Messages Queued Quota (MB)	Consumers	Replay State	Durable
<input type="checkbox"/>	CiBusinessPartner	On	On	Exclusive	0	<div><div></div></div>	917	0.3088	50	0	N/A	Yes
<input type="checkbox"/>	CiBusinessPartnerChecked	On	On	Exclusive	0	<div><div></div></div>	560	0.1963	50	0	N/A	Yes
<input type="checkbox"/>	CiBusinessPartnerCheckedInvalid	On	On	Exclusive	0	<div><div></div></div>	369	0.1266	50	0	N/A	Yes
<input type="checkbox"/>	CiBusinessPartnerChecker	On	On	Exclusive	0	<div><div></div></div>	0	0	50	2	N/A	Yes
<input type="checkbox"/>	CiBusinessPartnerCheckerDMQ	On	On	Exclusive	0	<div><div></div></div>	6	0.0019	50	0	N/A	Yes
<input type="checkbox"/>	CiBusinessPartnerConverted	On	On	Exclusive	0	<div><div></div></div>	0	0	50	0	N/A	Yes
<input type="checkbox"/>	CiLegacyAdapterIn	On	On	Exclusive	0	<div><div></div></div>	0	0	50	2	N/A	Yes
<input type="checkbox"/>	CiLegacyAdapterInDMQ	On	On	Exclusive	0	<div><div></div></div>	880	0.5214	50	1	N/A	Yes
<input type="checkbox"/>	CiLegacyAdapterOut	On	On	Exclusive	0	<div><div></div></div>	3	0.0001	50	0	N/A	Yes
<input type="checkbox"/>	CiSalesOrder	On	On	Exclusive	0	<div><div></div></div>	1,371	0.8406	50	0	N/A	Yes
<input type="checkbox"/>	CiSalesOrderChecked	On	On	Exclusive	0	<div><div></div></div>	0	0	50	0	N/A	Yes
<input type="checkbox"/>	CiSalesOrderNotification	On	On	Exclusive	0	<div><div></div></div>	0	0	50	2	N/A	Yes
<input type="checkbox"/>	CiSalesOrderNotificationProcesse...	On	On	Exclusive	0	<div><div></div></div>	766	0.4715	50	0	N/A	Yes

Congratulations, if you can see the consumer connected to your queue, then your IFlow has successfully connected to the AEM and is up and running waiting for event messages to arrive. If you

have the simulator tool set up and your queue configured properly you should be able to trigger the IFlow. To confirm go to Monitor Integrations and APIs -> Monitor Message Processing -> All

The screenshot shows the SAP Integration Suite interface. On the left is a navigation menu with options: Home, Discover, Design, Integrations and APIs, B2B Scenarios, Custom Type Systems, MIGs, MAGs, and Monitor. The 'Monitor' section is expanded, showing 'Integrations and APIs' and 'B2B Scenarios'. The main area is titled 'Overview / Monitor Message Processing'. It has a 'Runtime' dropdown set to 'Cloud Integration'. Below are filters for 'Time' (set to 'Past Hour'), 'Status' (set to 'All'), and 'Artifact' (set to 'AEM From Scratch'). The time range is 'Aug 16, 2024, 12:36:18 - Aug 16, 2024, 13:36:18'. A table shows one message: 'AEM From Scratch' with status 'Completed' and a processing time of '6 ms'. On the right, a detailed view for 'AEM From Scratch' shows it was last updated at 'Aug 16, 2024, 13:22:27'. Below this, a green box states 'Message processing completed successfully.' and 'Processing Time: 6 ms'.

Of course, since the created IFlow is just an empty message consumer you will see nothing happening except for the entry in the Monitoring of the Integration Suite. The following exercises will introduce a more elaborate event handling.

## Scenario 2: “Dead Message Queue”

In this scenario we will modify the IFlow we just created to intentionally result in an error which in turn will cause AEM to move the message to a Dead Message Queue (DMQ).

- In the previous scenario we already configured the DMQ in our queue in AEM:

< Queues | CILegacyAdapterIn

Summary **Settings** Subscriptions Consumers Messages Queued Stats

Incoming ☒

Outgoing ☒

Access Type **Exclusive** Non-Exclusive

Messages Queued Quota (MB) 5000

Alert Thresholds ☐ % ☐ # Clear 18 Raise 25

Owner solace-cloud-client

Non-Owner Permission Consume

Maximum Consumer Count 1000

Alert Thresholds ☐ % ☐ # Clear 60 Raise 80

Maximum Message Size (B) 10000000

Maximum Delivered Unacknowledged Messages per Flow 10000

DMQ Name CILegacyAdapterInDMQ

Enable Client Delivery Count ☐

- Create a new queue with the name you have configured as DMQ Name in your initial queue. This queue will not need any subscriptions since the messages will be automatically moved here after Retry handling.

< Queues | CILegacyAdapterInDMQ

Summary **Settings** Subscriptions Consumers Messages Queued Stats

Incoming ☒

Outgoing ☒

Access Type **Exclusive** Non-Exclusive

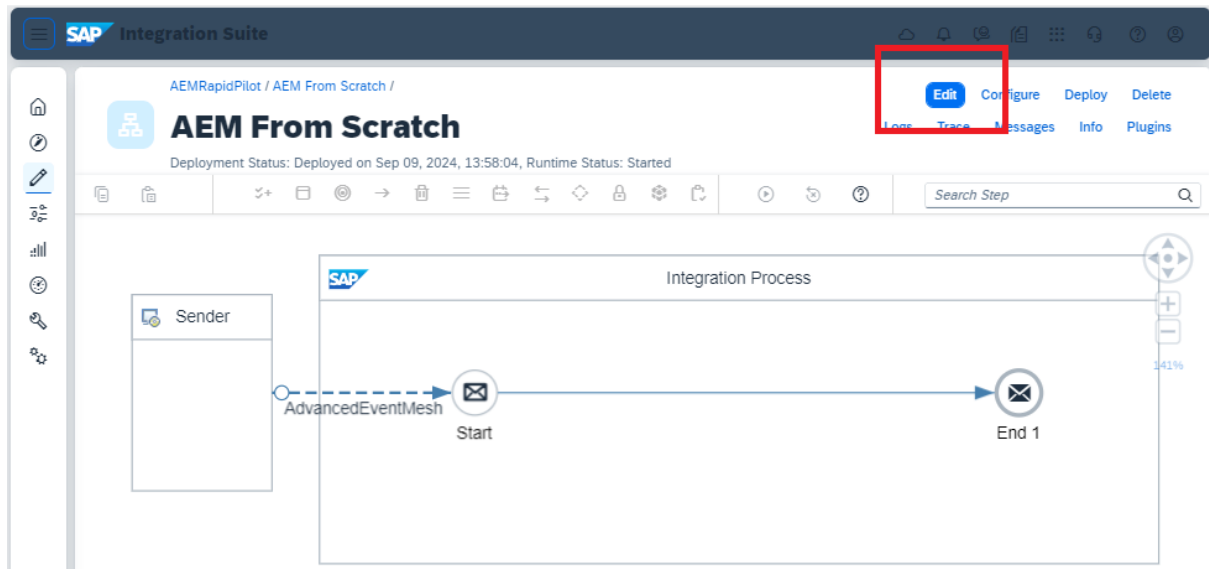
Messages Queued Quota (MB) 5000

Owner solace-cloud-client

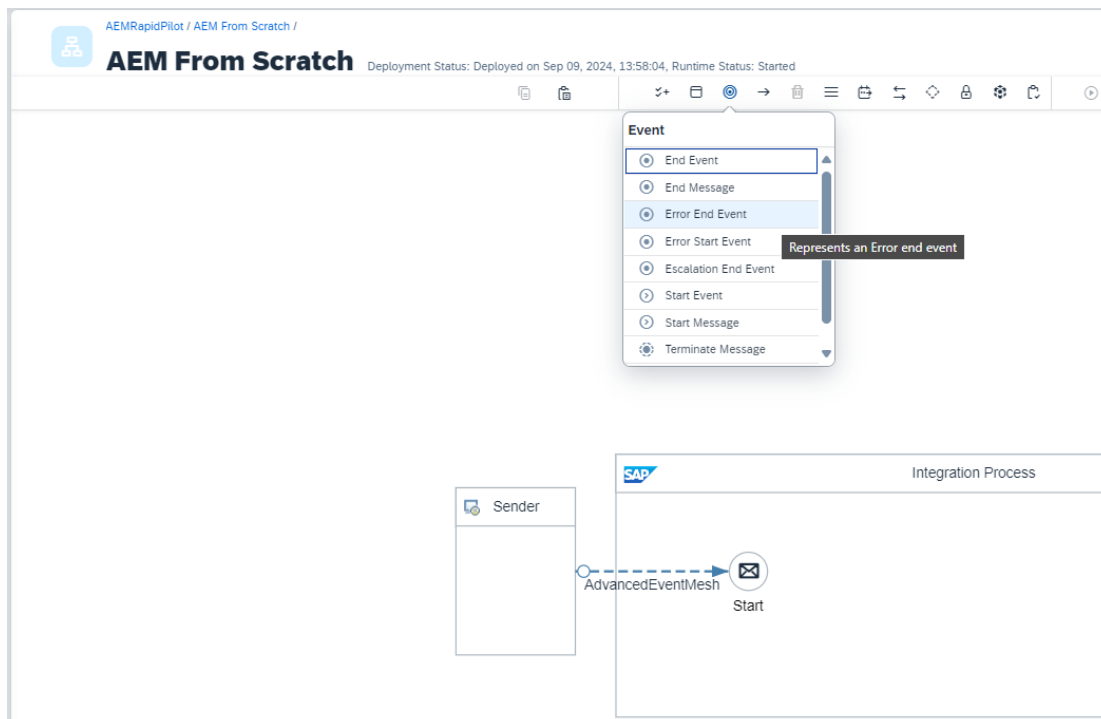
Non-Owner Permission Consume

Maximum Consumer Count 1000

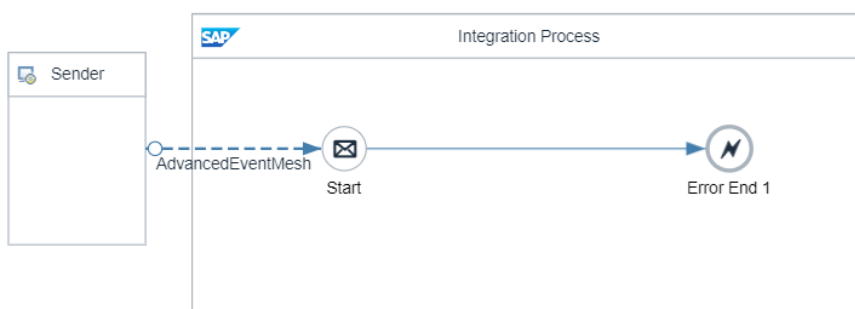
- Now we will make a tiny change to our IFlow. Go back into the Integration Suite and edit your IFlow.



- Delete the Message “End” and replace it with an Error End Event



- Click on the Message “Start” and draw the connector to the new Error End



- Save your changes and deploy the IFlow again



Let's now look at these Processing settings in the AEM adapter one by one:

- **Acknowledgement Mode: "Automatic on Exchange Complete"**  
The most important setting when it comes to not accidentally acknowledging and therefore removing a message from the broker's queue. This setting tells the flow/AEM adapter to only acknowledge (ack) the message after the flow has successfully completed processing the message. If any in the processing occurs, the AEM adapter will instead send a negative acknowledgment back (nack) to tell the broker to keep the message and retry it, because it couldn't be successfully processed by the flow. The alternative is to immediately ack the message when it's received, which will always result in the message being removed from the queue even if the flow fails to successfully process the message. (!!)
- **Settlement Outcome After Maximum Attempts: "Failed"**  
This setting controls the nack type and behaviour, we have two options here:
  - a) Failed, which will nack the message back to the broker and let's the broker check the retry count of the message to trigger retries based on the queue settings and only sending messages to DMQ when the retry count on the message has exceeded the max retry settings on the queue.
  - b) Rejected, which will nack the message telling the broker to immediately move the message to DMQ when the AEM adapter settings (Maximum Message Processing Attempts) are exceeded irrespective of queue settings.
- **Max. Message Processing Attempts: 2**  
Controls how often we want to retry a message inside the IFlow before we "give up" and pass it back to the broker.
- **Retry interval, Max Retry Interval and Exponential Backoff Multiplier**  
These are all settings that control how quickly we want to retry and whether we want to incrementally increase our retry delay with each failure. A good retry delay value prevents the IFlow from repeatedly retrying a message within a few milliseconds and gives some time for transient error situations to clear before we retry.

Note that the error handling and retry settings go hand-in-hand with the DMQ and retry settings on the input queue for this flow (queue retry settings multiply with the internal retry settings in the IFlow, e.g. if the IFlow tries 2 times internally every time we pass it a message and the broker is configured to retry the same message 3 times to the broker, then we might get 8 executions before the message is actually stopped being processed and moved to the DMQ [(1 initial attempt + 3 times retry) \* 2 times retry inside the IFlow = 8 processing attempts]). We will observe this behaviour in the next exercise/scenario.

Now if you use any tool you prefer to publish a Sales Order Event to our queue (the ERP simulator or the Try-Me console in the AEM broker itself) you will find the IFlow has been triggered and resulted in failure exactly 8 times:

Overview / Monitor Message Processing

Runtime: Cloud Integration

Time: Past Hour

Status: All

Artifact: AEM From Scratch

Sep 09, 2024, 13:27:46 - Sep 09, 2024, 14:27:46

Messages (16)

Artifact Name

Status

AEM From Scratch	Failed
Sep 09, 2024, 14:07:32	4 ms
AEM From Scratch	Failed
Sep 09, 2024, 14:07:31	5 ms
AEM From Scratch	Failed
Sep 09, 2024, 14:07:31	5 ms
AEM From Scratch	Failed
Sep 09, 2024, 14:07:31	5 ms
AEM From Scratch	Failed
Sep 09, 2024, 14:07:31	5 ms
AEM From Scratch	Failed
Sep 09, 2024, 14:07:30	5 ms
AEM From Scratch	Failed
Sep 09, 2024, 14:07:30	5 ms
AEM From Scratch	Failed
Sep 09, 2024, 14:07:30	5 ms
AEM From Scratch	Failed
Sep 09, 2024, 14:07:30	8 ms

AEM From Scratch

Last Updated at: Sep 09, 2024, 14:07:32

Status

Properties

Logs

Artifact Details

Message processing failed.

Processing Time: 4 ms

Error Details

com.sap.esb.camel.error.handler.ErrorEventException: Error Event Exception

Properties

Message ID: AGbe5QQovkAQDICKztahd29EH40W

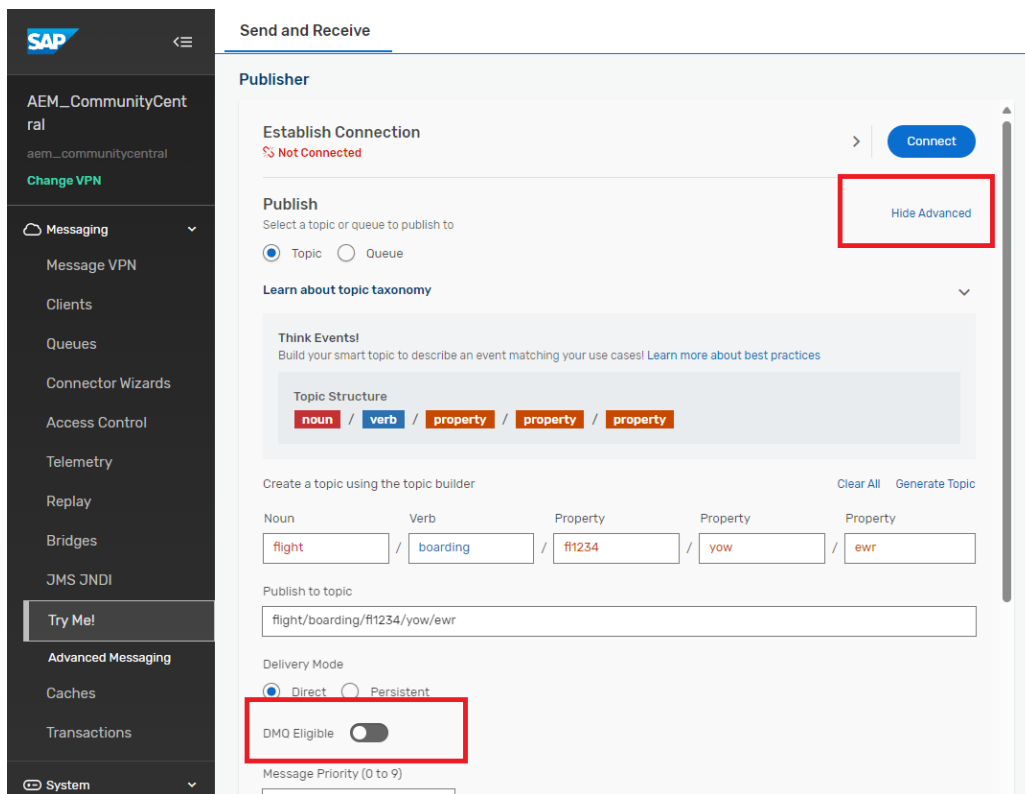
Correlation ID: AGbe5QT2nC9wzerifg-HxKUo0NqD

Logs

And your DMQ in your message Broker will now have a new message in it:

Queue Name	Incomi...	Outgoi...	Access Type	Partition Count	Messages Queued (%)	Messages Queued (msgs)	Mess Quet (MB)
CILegacyAdapterIn	On	On	Exclusive	0		0	0
CILegacyAdapterInDMQ	On	On	Exclusive	0		1	0.00

Be aware that the Publisher decides if an event is eligible to be stored in a DMQ, so when testing with the Try-Me feature be sure to check "DMQ eligible" under "Show Advanced"



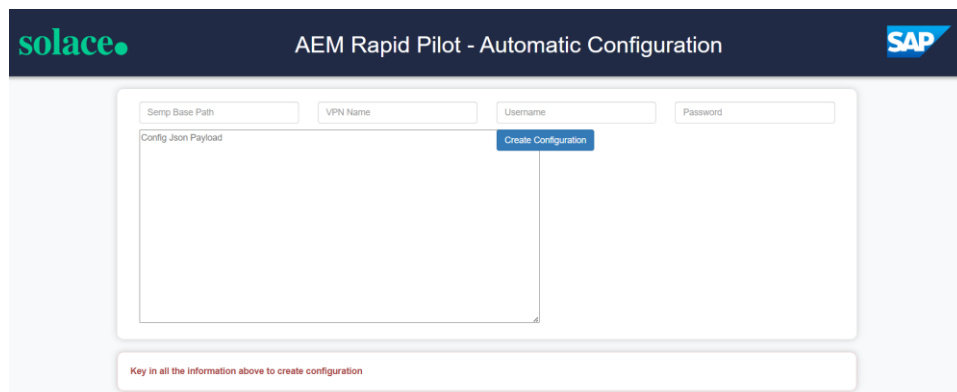
## Automated AEM Broker Setup via APIs and CI/CD (optional)

All configuration within AEM can be automated using a combination of two APIs:

- AEM Cloud API (for any configuration on the AEM Cloud Console)
- The broker's SEMP API (for any configuration on the broker service directly)

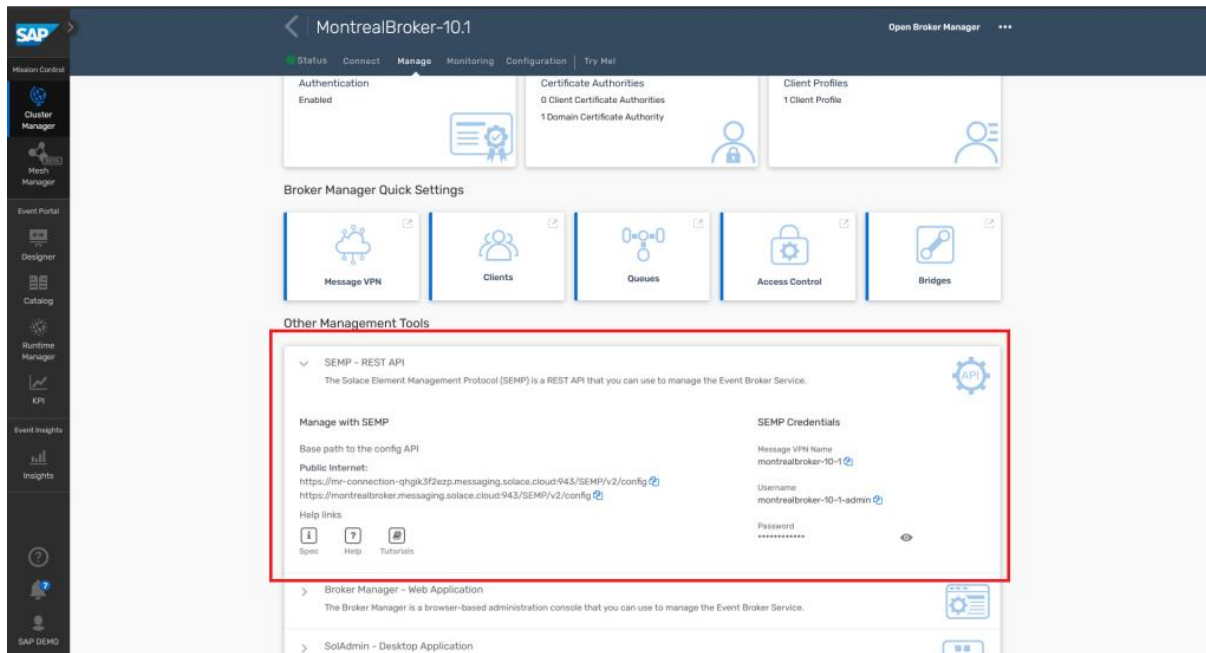
The documentation of these APIs can be found in the AEM docs [here](#)

Both these APIs are RESTful and can be used in numerous ways to pull/push configuration like queues and client configuration through CI/CD pipelines or configure/promote Event Portal content in step with code promotion from environment to environment. There are many options/tools that can be used to automate these tasks, like Jenkins, Ansible, scripts, Terraform etc. For the purpose of this exercise, we are going to use a sample application written in CAP that has the nice advantage of running in our browser that we can point at our broker's API and feed some configuration files.

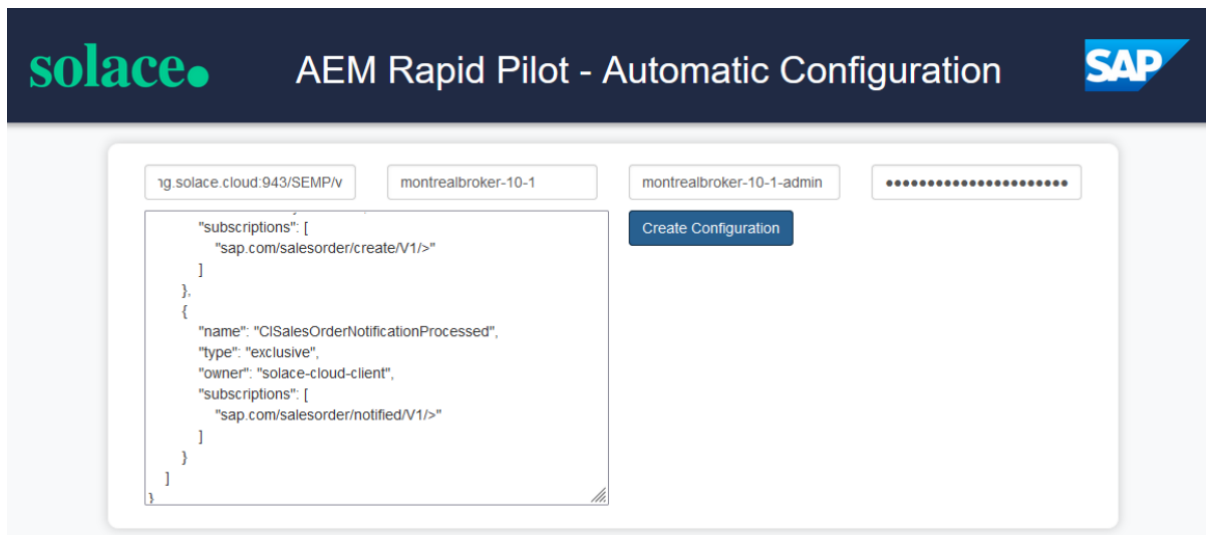


The AEM Rapid Pilot - Automatic Configuration can be accessed [here](#)

We will need some details from your AEM service again to connect the configuration tool with your AEM service. Let's head to our Advanced Event Mesh Console and go to Cluster Manager, select the service that you want to connect your Integration Suite flows to and go to the "Manage" tab. Take a note of the connectivity details underneath "SEMP - REST API" (click on the section to open it up)



Copy & paste the URL, vpn name, admin username and password into the config tool:



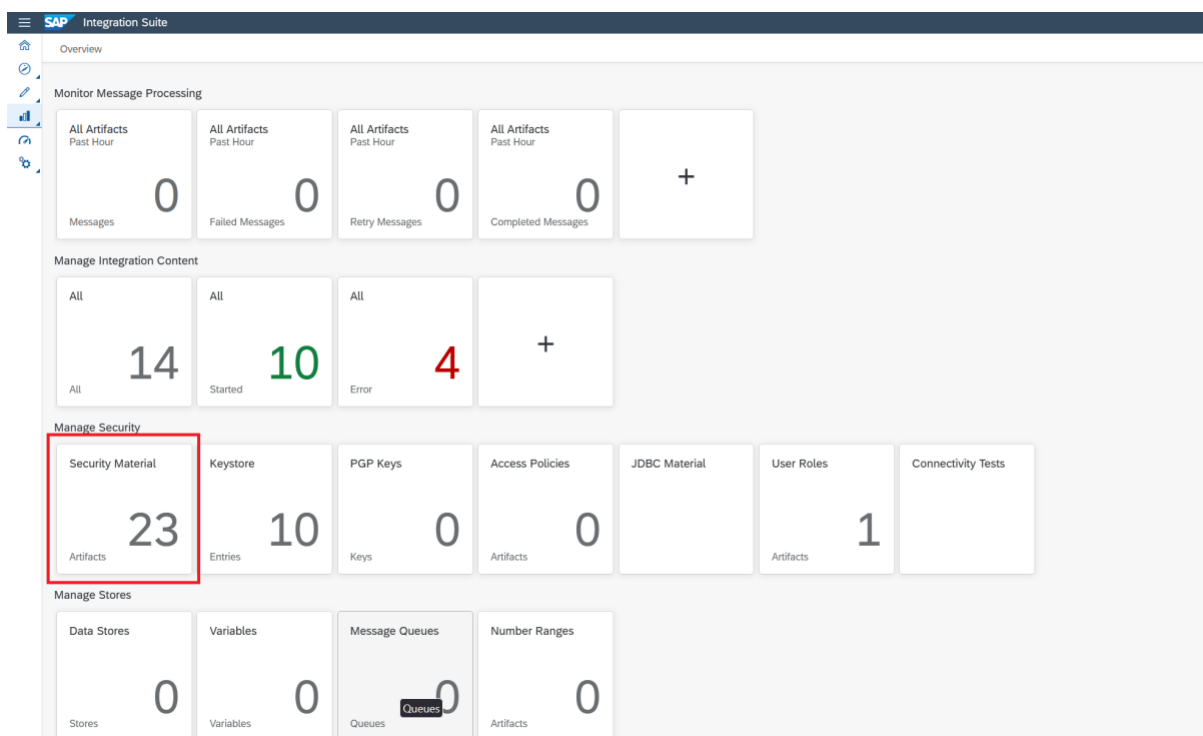
Please download the configuration file **IS\_SEMP.json** provided to you together with the content packages and copy & paste the content into the "Config JSON Payload" input field. Hit "Create Configuration" to apply this config to your broker.

## Scenario 3: “Sales Order E-Mail Notification”

### Setup/Configure Dependency Services

We will give you connectivity details to one of our brokers where we have an IFlow deployed that is configured to send emails via an external email service to enable us to automatically send confirmation emails. You are welcome to use a SMTP mail server of your own if you have access to the necessary credentials. You will need to edit the provided IFlow slightly but if you are interested, we are happy to support.

- If you want to use the broker we provide you need to go to the Integration Suite – Monitor Artefacts – Manage Security – Security Material



- You should have created a Secure Parameter (CABrokerUserPass) in the first exercise already
- Create another Secure Parameter *email-profile-pwd* and store the password we have handed out in the workshop.

### Queue Setup

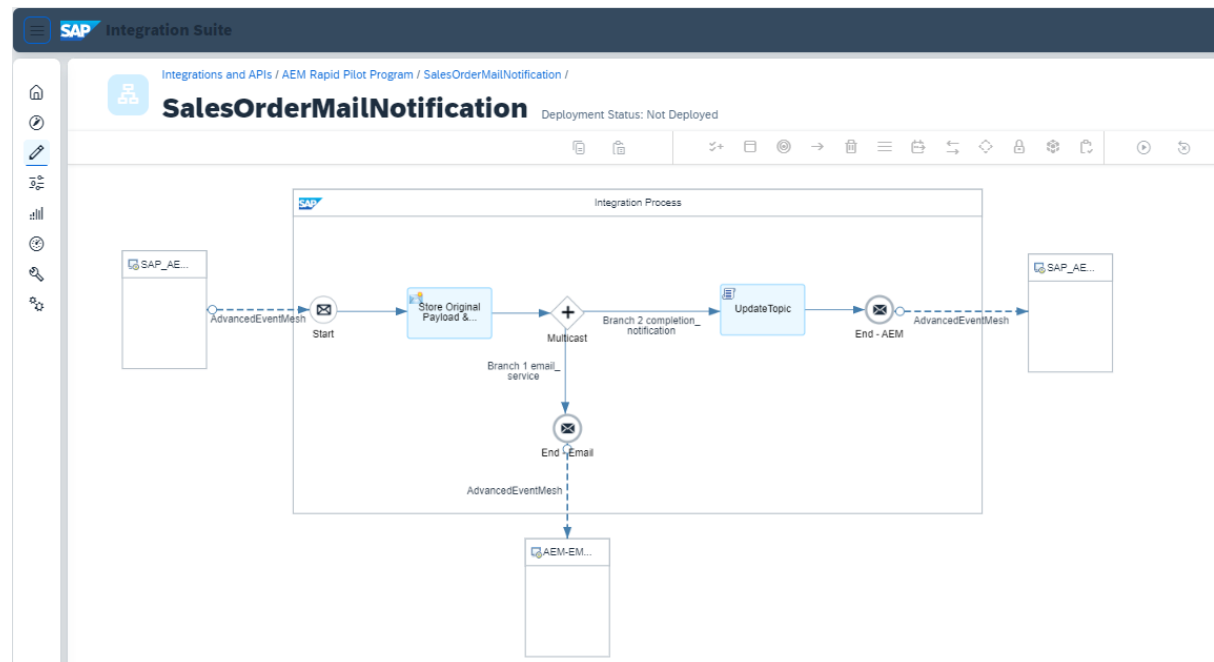
If you have used the CI/CD tooling in the previous section to automate the configuration of the AEM queues you can skip this section. If not, have a look at the IS\_SEMP.json provided to you. For this exercise you will need the following queues:

- **CISalesOrderNotification**
- **CISalesOrderNotificationProcessed**

Create the queues like you did in the previous sections using the details in the SEMP file for topic subscriptions.

## IFlow Setup

Let us first look at the SalesOrderMailNotification IFlow:



This flow gets triggered by Sales Order events and does two things:

- It creates an email request and by forwarding this event to an email service iflow on another broker (hosted by us) on topic `sap.com/emailnotification/created/V1`.
- It sends a new event to `sap.com/salesorder/notified/V1/{salesOrg}/{distributionChannel}/{division}/{customerId}` to indicate that the email request was successfully forwarded.

- Hit configure at the top right and fill in the details to connect to your AEM broker service like you did in the previous exercises

Configure "SalesOrderMailNotification"	
Sender	Receiver
Sender:	SAP_AEM_Sender
Adapter Type:	AdvancedEventMesh
Connection	
Host:	tcps://broker.messaging.solace.cloud:55443
Message VPN:	vpn_name
Username:	solace-cloud-client
Password Secure Alias:	CABrokerUserPass

- There is no need to configure the Receiver separately, the configuration will fill for both.

Configure "SalesOrderMailNotification"

Sender	Receiver
Connection	
Receiver:	SAP_AEM_Receiver
Adapter Type:	AdvancedEventMesh
Host:	tcps://broker.messaging.solace.cloud:55443
Message VPN:	vpn_name
Username:	solace-cloud-client
Password Secure Alias:	CABrokerUserPass


- No need to configure the adapter connecting to our email service, we've prepopulated this one and we have already deployed the necessary security configuration in the step above
- Hit deploy at the top right.

If you get confused about which parts of your IFlow to connect to your broker and which ones to connect to our broker, remember this simple rule: The sender and receiver (left and right of your IFlow) connect to your broker. The connector down at the bottom connects to our, from your point of view external, broker.

Like in the first exercises feel free to test if everything is setup correctly. The IFlow on our end that will send out the email notification expects an event with the same structure as the **ExampleSOEvent.json** provided by us. Simply modify the content of said SalesOrder events with your mail address before pushing it into your broker.

If all went well you will receive a mail (check your Spam/Junk folders):

Sales Order SO1001 has been processed



aemrapiddeployment@gmail.com

To: Heinrich, Erik

Retention Policy: Junk Email (30 days)

This item will expire in 30 days. To keep this item longer apply a different Retention Policy.

Expires: 09/10/2024

[You don't often get email from [aemrapiddeployment@gmail.com](mailto:aemrapiddeployment@gmail.com). Learn why this is important at <https://aka.ms/LearnAboutSenderIdentification>]

Dear ABC Corp,

Your order has been processed!

👤 Reply

👤 Reply All

➔ F

## Scenario 4: “Business Partner Address Check” (optional)

### Setup/Configure Dependency Services

The next IFlow that we are going to deploy is invoking the SAP Data Quality Management service (DQM) to check and cleanse address data in the BusinessPartner events. For the flow to work properly, you will need a working DQM service subscription so you can configure your IFlow with this. For completing this section, you have two options:

#### A) Use DQM service credentials provided by us during the workshop

We will hand out the token and connectivity details to our DQM service, which can use

#### B) Alternative: Activate your own SAP Data Quality Management service in BTP

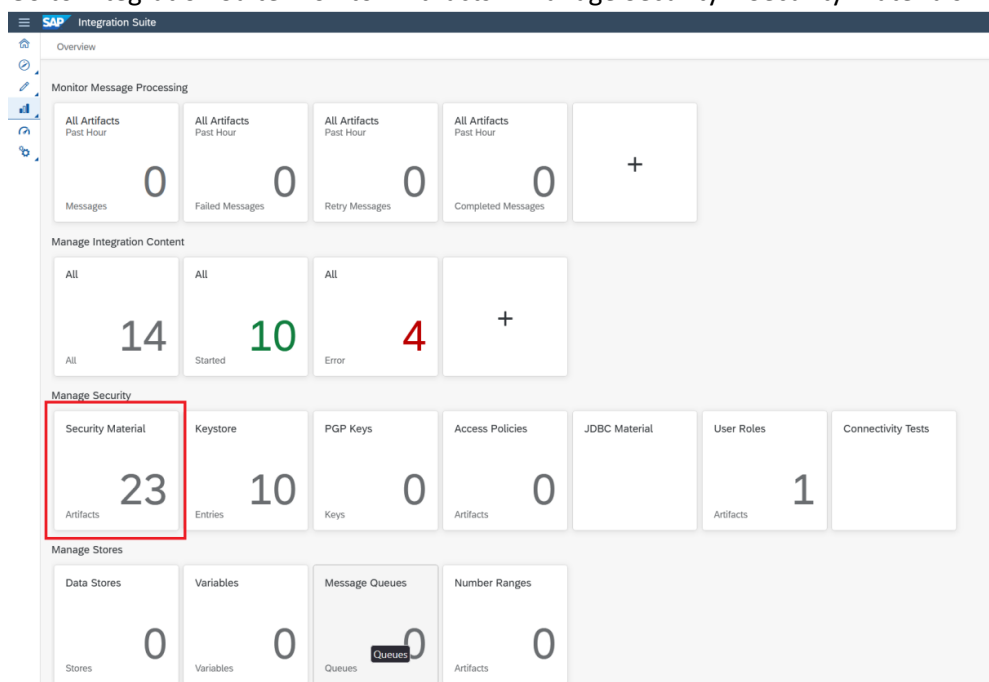
Please note that if you want to proceed down this route, it may take some time to complete, so you may want to complete this in your own time after the workshop.

The good news, if you don't have a DQM subscription already or are not using our instance, then you can use a free tier subscription for this purpose. Please follow along the steps in this [blog post](#) by Hozumi Nakano to active the service.

Additionally, you will have to create a service instance and a service key to be configured with your integration flow later. Follow [these steps](#) to create a service instance and key. Take a note of the URL and user credentials once you've activated the service.

Independent of whether you are using our provided DQM service or your own you will want to create an OAuth2 Client Credential.

- Go to Integration Suite Monitor Artifacts – Manage Security – Security Materials








- Create OAuth2 Client Credentials with the name *DMQCred*, this is important because it will be used by the provided IFlow.

Edit OAuth2 Client Credentials

---

Name: *	<input type="text" value="DQM Cred"/>
Description:	<input type="text" value="Client credential for calling DQM service"/>
Token Service URL: *	<input type="text" value="https://scpsubaccount.authentication.eu10.hana.ond ..."/>
Client ID: *	<input type="text"/>
Client Secret: *	<input type="password"/> 
Client Authentication: *	<input type="text" value="Send as Request Header"/> 
Scope:	<input type="text"/>
Content Type:	<input type="text" value="application/json"/> 
Resource:	<input type="text"/>
Audience:	<input type="text"/>

---

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

- For Token Service URL use the uaa/url from the DQM service key **appended with /oauth/token** or the one we provide
- For ClientID/Secret use uaa/clientid and uaa/clientsecret from the DQM service key or the one we provide

## Queue Setup

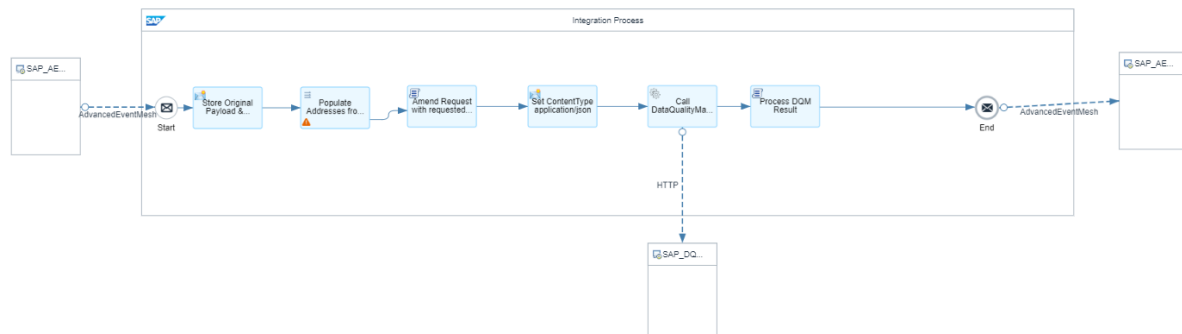
If you have used the CI/CD tooling in the previous section to automate the configuration of the AEM queues you can skip this section. If not, have a look at the IS\_SEMP.json provided to you. For this exercise you will need the following queues:

- **CIBusinessPartnerChecker**
- **CIBusinessPartnerCheckerDMQ**
- **CIBusinessPartnerChecked**
- **CIBusinessPartnerCheckedInvalid**

Create the queues like you did in the previous sections using the details in the SEMP file for topic subscriptions.

## IFlow Setup

Let's look at the BusinessPartnerAddressCheck IFlow:



This flow receives Business Partner Create and Change events and invokes the Data Quality Management Service in BTP to check and correct the addresses inside the Business Partner event payload. It does this by

1. Storing the original event payload in an environment variable.
2. Populating the DQM request payload with the addresses in the input event.
3. Invoking the DQM service over REST and
4. Parsing the response, checking whether the DQM service evaluated the input addresses to be Valid, Invalid, Blank or has Corrected them.
5. Merging any corrected addresses back into the original payload.
6. And finally publishing the result back as a new event to the AEM broker with an updated topic in the format:

*sap.com/businesspartner/addressChecked/V1/{businessPartnerType}/{partnerId}/{addressCheckStatus}*

Let's also look at what happens in order to publish a new event back to the Advanced Event Mesh broker. First, on the integration flow overall configuration settings, we are preserving the destination header field to have access to the original topic that this event was published on. This matters, because the event may contain valuable meta-data that helps us and downstream consumers filter for events relevant to them and it saves us from reparsing the payload, which can be CPU and I/O intensive.

The screenshot shows the 'Integration Flow' configuration page. The 'Runtime Configuration' tab is selected. The 'Runtime Profile' is set to 'Cloud Integration'. The 'Namespace Mapping' is empty. The 'Allowed Header(s):' field is highlighted with a red box and contains the value 'Destination'. The 'HTTP Session Reuse' is set to 'None'.

Secondly we are using a couple of lines in the script that is evaluating the DQM service result and merging the corrected addresses back into the original payload to retrieve and parse the original topic, replace one level (the verb) to create a new event and amend another extra meta-data level that contains the result of the address check (either Valid, Corrected, Invalid or Blank), which can be

used by downstream systems to filter for specific outcomes. We are storing the newly created topic in the Destination field of the message header.

```
1- /* Refer the link below to learn more about the use cases of script.
2- https://help.sap.com/viewer/368c481cd6954bdfaf5d8a35479fd4eaf/Cloud/en-US/148851bf8192412cha1f9d2c17f4bd25.html
3-
4- If you want to know more about the SCRIPT APIs, refer the link below
5- https://help.sap.com/doc/a56f52e1a3b0e4ebac77adbf5b2b26/Cloud/en-US/index.html */
6- importClass(com.sap.gateway.ip.core.customdev.util.Message);
7- importClass(java.util.HashMap);
8-
9- function processData(message) {
10- //Body
11- var DQResponse = String(message.getBody( new java.lang.String().getClass()));
12- DQResponse = JSON.parse(DQResponse);
13-
14- var origBody = message.getProperty("ORIG@PAVLOAD");
15- var origTopic = message.getProperty("ORIG@TOPIC");
16- origBody = JSON.parse(origBody);
17-
18- var i=0;
19- var addressStatus = "Unknown"
20- origBody.businessPartner.forEach(function(bp){
21- bp.addressLink.forEach(function(link){
22- link.address.forEach(function(address){
23- const currAddress=DQResponse.addressOutput[i];
24- if (currAddress.addr_asmt_info == "C") {
25- //corrected address; copy changes over from each addressoutput into customer addressoutput
26- address.street = currAddress.std_addr_prim_address + currAddress.std_addr_sec_address;
27- address.city = currAddress.std_addr_locality_full;
28- address.zipCode = currAddress.std_addr_postcode_full;
29- address.country = currAddress.std_addr_country_2char;
30- //additional fields available, not used:
31- //currAddress.std_addr_region_full
32- addressStatus = "Corrected";
33- i++;
34- } else if (currAddress.addr_asmt_info == "V") {
35- //valid address, nothing to do
36- addressStatus = "Valid";
37- } else if (currAddress.addr_asmt_info == "I") {
38- //invalid address, can't autofill, just flag on the topic
39- addressStatus = "Invalid";
40- } else if (currAddress.addr_asmt_info == "B") {
41- //invalid address, can't autofill, just flag on the topic
42- addressStatus = "Blank";
43- }
44- })
45- })
46- })
47-
48- var topicLevels = origTopic.split("/");
49- topicLevels[2] = "addressChecked";
50- var outputTopic = topicLevels.join("/") + "/" + addressStatus;
51-
52- message.setHeader("Destination", outputTopic);
53- message.setBody(JSON.stringify(origBody));
54- return message;
55- }
```

Lastly, the AEM Receiver adapter is configured to persistently (to avoid message loss) publish to a topic, taking the value from the header field that we set in the previous step/script.

AdvancedEventMesh

General Connection **Processing** Message Properties

PUBLISHER PROCESSING DETAILS

Delivery Mode: Persistent

Endpoint Type: Topic

Destination Name: \${header.Destination}

Message Type: Automatic

SYNCHRONOUS REQUEST PROCESSING DETAILS

Convert Publish Into Synchronous Requestor?: ☐

Configuring and deploying the BusinessPartnerAddressCheck IFlow:

- Hit configure at the top right and fill in the details to connect to your AEM broker service like you did in the previous exercises.

## Configure "BusinessPartnerAddressCheck"

Sender	Receiver
Sender: SAP_AEM_Sender	
Adapter Type: AdvancedEventMesh	
Host: tcps://mr-connection-h91kb3o1b6w.messaging.solace.cloud:55443	
Message VPN: aem_communitycentral	
Username: solace-cloud-client	
Password Secure Alias: CABrokerUserPass	

- No need to configure the adapter connecting to DQM, we've prepopulated this one and we have already deployed the necessary security configuration in the step above
- Hit deploy at the top right.

As before test if your scenario works using either the ERP simulator tool or the Try-Me feature. In the following example we happened to trigger a BusinessPartner event from the ERP simulator tool with an address in Vietnam that was missing necessary punctuation. You can see the actual JSON content of the event before the IFlow on the left and after on the right.



The event was then published on the topic:

<sap.com/businesspartner/addressChecked/V1/Customer/BP1010/Corrected>

Which is included in the wildcard subscription of the "Checked" queue:

<input type="checkbox"/> Queue Name	Incoming	Outgoing	Access Type	Partition Count	Messages Queued (%)	Messages Queued (msgs)	Messages Queued (MB)	Messages Queued Quota (MB)	Consumers
<input type="checkbox"/> #telemetry-Tracing	Off	On	Non-Exclusive	0		0	0	800,000	1
<input type="checkbox"/> BusinessPartner	On	On	Exclusive	0		10,768	8.05	5,000	0
<input type="checkbox"/> CiBusinessPartnerChecked	On	On	Exclusive	0		1	0.0003	5,000	0
<input type="checkbox"/> CiBusinessPartnerCheckedInvalid	On	On	Exclusive	0		0	0	5,000	0
<input type="checkbox"/> CiBusinessPartnerChecker	On	On	Exclusive	0		0	0	5,000	1
<input type="checkbox"/> CiBusinessPartnerCheckerDMQ	On	On	Exclusive	0		0	0	5,000	0

## Scenario 5: “SFTP as Legacy Integration” (optional)

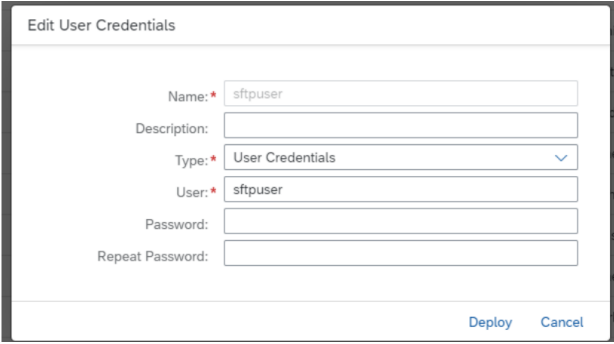
### Setup/Configure Dependency Services

With the introduction of the Advanced Event Mesh, not all customers will be ready to adopt an Event Driven approach for all their applications. They will be able to innovate in some areas but also must sustain existing applications that use for example Flat Files. In this scenario, we will use a couple of the common Cloud Integration artifacts to facilitate this scenario. The common integration path for Flat Files is SFTP so in this scenario we will want to leverage a SFTP server. Similar to previous exercises you have two options:

#### A) Use a SFTP server of you own business that you might have access to

If you want to proceed with this or if you want to complete this exercise after the workshop when you have access to an SFTP server you will have to create/modify 2 Security artifacts in the Integration Suite:

- A User Credential to Authenticate against your SFTP server. This can be as simple as Username/Password which you input and then deploy



The screenshot shows a web form titled "Edit User Credentials". It has the following fields: "Name" with the value "sftpuser", "Description" (empty), "Type" with a dropdown menu showing "User Credentials", "User" with the value "sftpuser", "Password" (empty), and "Repeat Password" (empty). At the bottom right, there are two buttons: "Deploy" and "Cancel".

- You will have to add your SFTP server to the SSH Known Hosts list of your Integration Suite. Follow the instructions in [this blog](#) for more information.

#### B) Alternative: Use the included Groovy script in the IFlow to generate the file as an attachment to the IFlow run

This option will allow you to complete this exercise without an SFTP server. We still encourage you to try to setup the SFTP integration on your time.

### Queue Setup

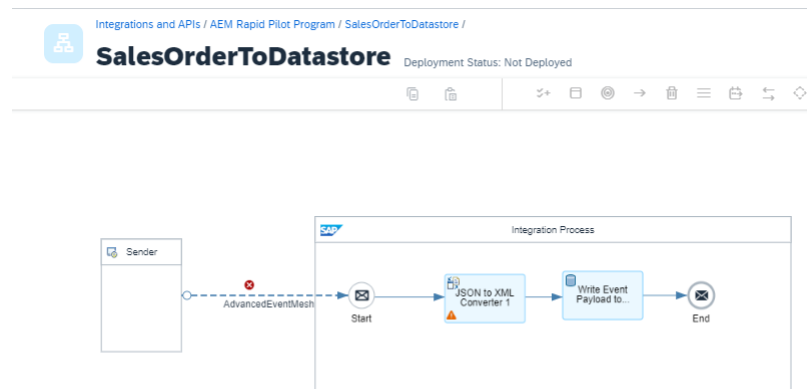
If you have used the CI/CD tooling in the previous section to automate the configuration of the AEM queues you can skip this section. If not, have a look at the IS\_SEMP.json provided to you. For this exercise you will need the following queues:

- **FFSalesOrders**

Create the queues like you did in the previous sections using the details in the SEMP file for topic subscriptions.

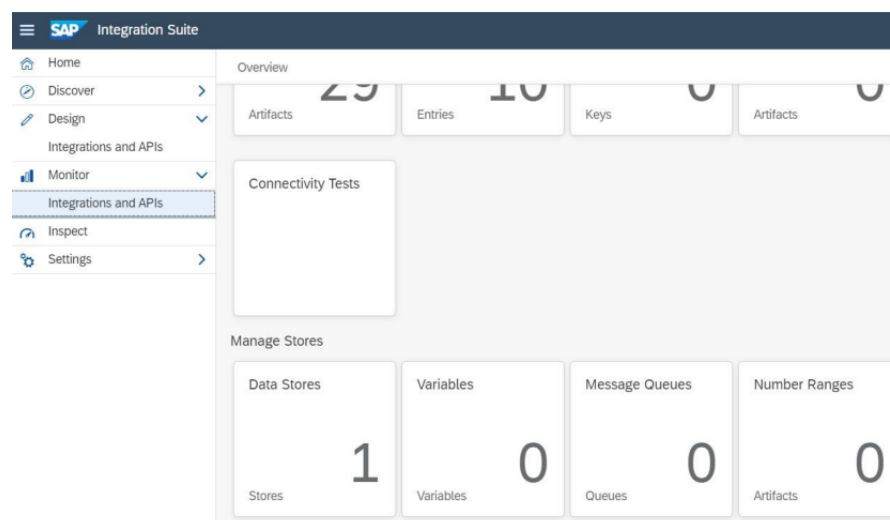
## IFlow Setup

The IFlow “SalesOrderToDatastore” will subscribe to events on the Advanced Event Mesh, convert them to XML (E.g. The DataStore object only works with XML) and then write them into the DataStore.

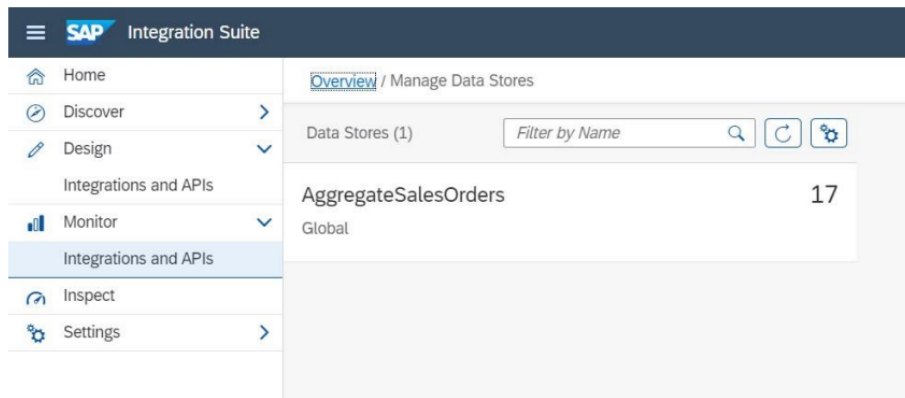


- To configure this IFlow all you need to do is enter your AEM broker information like in the previous exercises.
- Deploy the IFlow

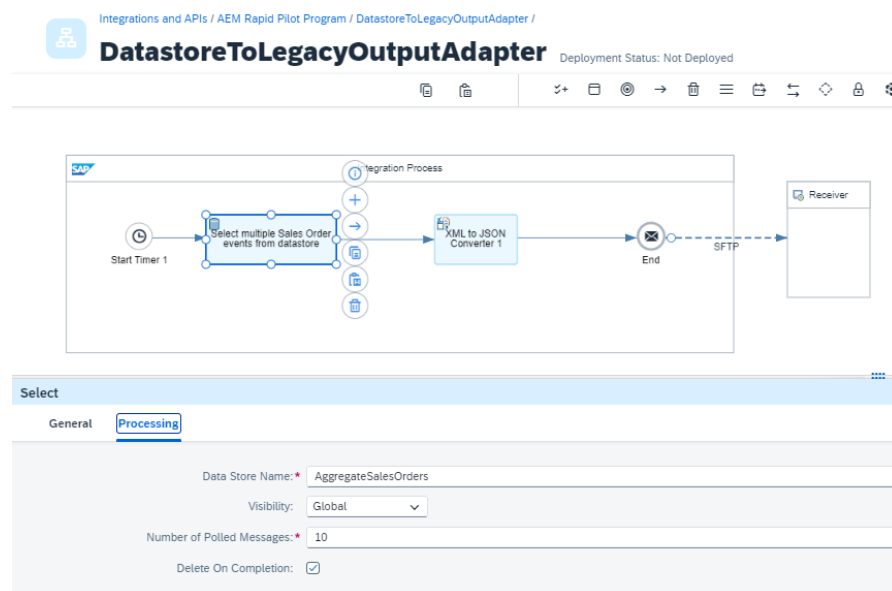
By using any testing tool provided to you (ERP Simulator, Try-Me feature,..) you can do a quick testing. From the monitor, you should now see a Data Store as shown in the screenshot. (\*\*\*) Of course, if you are already using Data Stores for your integrations, you should see the number incremented \*\*\*)



Clicking on the Data Stores Tile, you should see the name of the Data Store specified in the IFlow.



We will then use a second iFlow to pull the events from the DataStore, convert them back to JSON and then use the SFTP adapter to create a flat file with a batch of up to 10 of these events. This is based on the settings in the Data Store operation.



- For this IFlow you will need to configure your SFTP details and credentials. If you are not able to provide your own SFTP server, feel free to edit the IFlow, remove the SFTP adapter and instead add the Groovy script **AggregatedSalesOrdersToAttachment.groovy** in a Groovy script step to the end of your IFlow.

### Configure "DatastoreToLegacyOutputAdapter"

**Receiver**

Receiver:

Adapter Type:

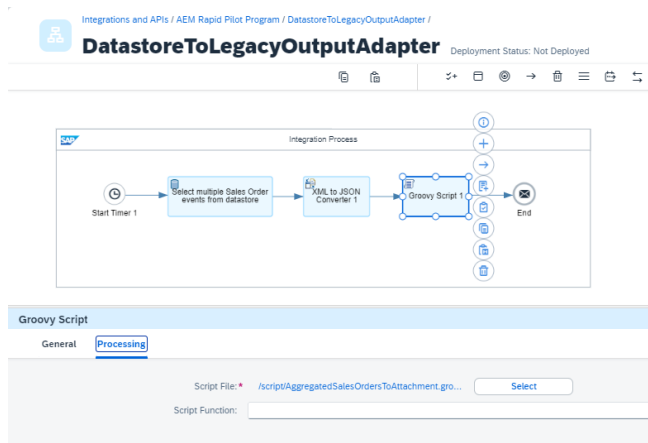
**Target**

Directory:

File Name:

Address:

Credential Name:



- The Timer of the IFlow is by default set to "execute once". Therefore, it will run only one time after it is deployed. Make sure you have sent some events into the datastore before deploying the IFlow.

You should either see a flat file appear in the SFTP directory you specified or if you decided not to use SFTP you can check the Monitoring of the Integration Suite and see the flat file attached to the IFlow run there.

Message Status Overview Hide Filter

Status: All Type: All Package: All Artifacts: All or ID: Message, Correlation or Application Message

0:01

**DatastoreToLegacyOutputAdapter**

Last updated at: Sep 12, 2024, 11:09:49

Status: Completed 415 ms

Message processing completed successfully.

Processing Time: 415 ms

**Properties**

Message ID: AG2H3BPHB2H9Y27Hkgd0X  
Correlation ID: AG2H3BPHB2H9Y27Hkgd0X

**Retention Periods**

for Completed Messages: 30 days  
for Uncompleted Messages: 30 days

**Logs**

Log Level: Info  
Instance ID: 0

**Attachments**

Entries (1)

Name	Type	Modified At	Size
salesorders	text/plain	Sep 12, 2024, 11:09:49	1 KB



## Security and fine-grained access control to topics (optional)

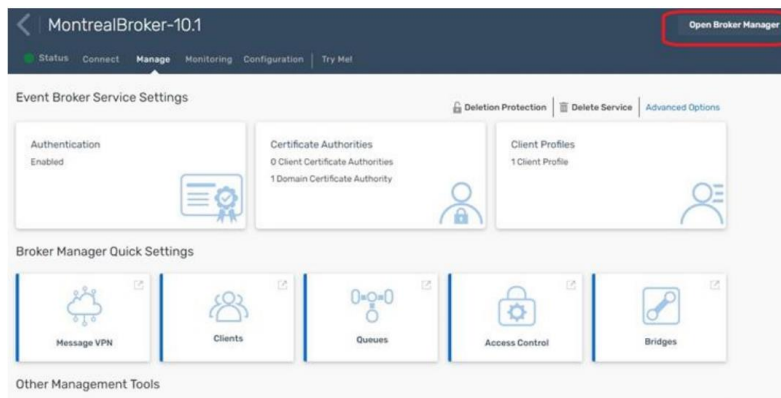
For this section and in general, we would like to take this opportunity to answer a common question. Is it possible to prevent clients from publishing to certain Topics and/or subscribing to certain topics?

**The answer is absolutely!**

We would like everyone to experience an Access Control List and how it can be used to control what is published or subscribed.

### Experimenting publishing and subscribing to protected topics

From the console, we need to navigate to the broker manager. You can get there by either clicking on the "Open Broker Manager" button or clicking any of the Tiles labelled "Clients", "Queues", "Access Control".



You will then see the Broker Manager Screen and on the left you will find the "Try-Me" test client. Click on it, to reveal the information you must provide to connect: For this screen, you will be trying to connect to our Broker (Note: We are doing this because your iFlow on Day 3 will send an Event to our broker to send an email) where we have created an ACL to limit what you can do and on what topics you can publish. The information you will use is as follows:

*Broker URL: tcps://mr-connection-h91kb3o1b6w.messaging.solace.cloud:55443*

*Message VPN: aem\_communitycentral*

*Client Username: email-profile*

*Client Password: \*\*\*\*\* - provided during the course*

**Publisher**

**Establish Connection**  
 % Connected Disconnect

Broker URL

Message VPN

Client Username

Client Password

**Publish**  
 Select a topic or queue to publish to Show Advanced

Once you have entered in the connectivity information, you should see the "Connected" message in blue. Once connected, change nothing and hit "Publish", you should immediately see the "Publish ACL Denied" on this action because the ACL will not permit you to complete this action. You can try the exact same thing with a subscription as well.

**Publisher**

**Establish Connection**  
 % Connected Disconnect

**Publish**  
 Select a topic or queue to publish to Show Advanced

☒ Topic ☐ Queue

[Learn about topic taxonomy](#)

Publish to topic

Delivery Mode  
☒ Direct ☐ Persistent

Message Content  

```
{
  "airline": "ExampleAirline",
  "region": "Ontario",
  "requestid": 44334,
  "flight": {
    "flightModel": "boeing737",

```

Publish

Message is rejected with error  
 Publish ACL Denied

**Subscriber**

**Establish Connection**  
 % Connected Disconnect

**Subscribe**  
 How do you want to receive messages

[Subscribe to a topic to receive direct messages](#)

Topic Subscriber

Subscribe

Subscribed Topics  
 flight/> X

Subscription request failed with error  
 Subscription ACL Denied

[Bind to an endpoint to receive guaranteed messages](#)

Messages (Most Recent 20)  
☐ Direct ☐ Persistent ☐ Non-Persistent

Clear Stats Clear Messages

Now, lets head back to the "Publisher" and change the "Topic" to *sap.com/emailnotification/created/V1*

You can again use the content of the **ExampleSOEvent.json** we provided for exercise 3. Make sure to replace the email address with an actual address prior to hitting the publish button.

If you have entered the topic and message body correctly, you should see that 1 message has been published. So how did we do that? The magic happens in the ACL Profile as shown next.

## Broker Topic ACLs

We have changed the Default Publish Action to be Disallow. In other words, unless we specify an exception, the user profile associated with this ACL cannot publish anything by default. In this case, as you can see, we have listed one exception.

The screenshot shows the 'ACL Profiles' configuration page for a profile named 'email-profile'. The page has a navigation bar with tabs: 'Client Connect', 'Publish Topic' (which is selected), 'Subscribe Topic', 'Subscribe Share Name', and 'Profile'. Below the navigation bar, there is a section for 'Publish Default Action' with a dropdown menu set to 'Disallow'. Below this is an 'Exceptions' section with a search bar labeled 'Search by topic'. Under the search bar, there is a checkbox labeled 'Publish Exception Topic' which is unchecked. Below the checkbox, there is a list of exceptions, with one entry visible: 'sap.com/emailnotification/created/V1'.

< | ACL Profiles | email-profile

Client Connect **Publish Topic** Subscribe Topic Subscribe Share Name Profile

Publish Default Action Disallow

Exceptions Search by topic X

☐ Publish Exception Topic

☐ sap.com/emailnotification/created/V1

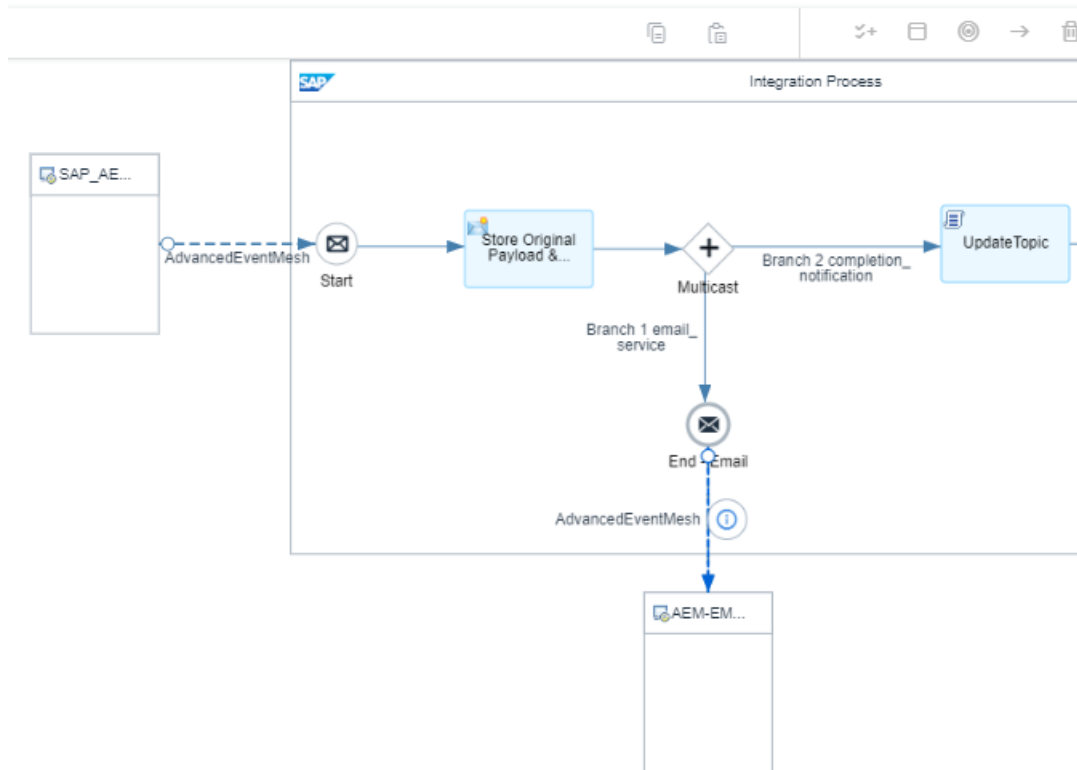
For the subscription settings, it's very simple: We specify the Default Action is "Disallow" and do not provide any exceptions. AKA, this ACL does not permit any subscriptions.

Now that you have this understanding, you will see when you configure/deploy this iFlow why we have a Username "email-profile". It has been assigned the ACL email-profile so the iFlow can publish to our broker but only on that topic. Everything else is prohibited by the email-profile user ACL, so it can only be used for this single purpose. See below screenshot for where you would have used these credentials in your IFlow.



# SalesOrderMailNotification

Deployment Status: Not Deployed



## AdvancedEventMesh

General

**Connection**

Processing

Message Properties

### RECEIVER CONNECTION DETAILS

Host:	tcps://mr-connection-h91kb3o1b6w.messaging.solace.cloud:55443
Message VPN:	aem_communitycentral
Username:	email-profile
Authentication Type:	Basic
Password Secure Alias:	email-profile-pwd

As you can see, broker ACLs are a quite powerful tool to tightly control access to the broker and its topics. You can separately control publish topics and subscribe topics and even IP address ranges that clients are allowed to connect from. In addition to topic ACLs, remember that queue access is controlled by the queue ownership model and the "other permission".