

SAP Advanced Event Mesh - Security

1. [What you'll learn: Overview](#)
2. [What you need: Prerequisites](#)
3. [Security and fine grained access control to topics](#)
4. [Takeaways](#)

SAP Advanced Event Mesh - Security

1. What you'll learn: Overview

- Fine-grained security access in AEM.

2. What you need: Prerequisites

- Complete all exercises to start up brokers in day 1.
You access and use the same broker you setup previously.

3. Security and fine grained access control to topics

For this exercise 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**".

The screenshot shows the Solace Broker Manager interface for the broker "MontrealBroker-10.1". At the top right, there is a red box around the "Open Broker Manager" button. Below it, the "Event Broker Service Settings" section displays three cards: "Authentication" (Enabled), "Certificate Authorities" (0 Client Certificate Authorities, 1 Domain Certificate Authority), and "Client Profiles" (1 Client Profile). At the top of this section are buttons for "Deletion Protection", "Delete Service", and "Advanced Options". The "Broker Manager Quick Settings" section below contains five cards: "Message VPN" (with a cloud icon), "Clients" (with a user icon), "Queues" (with a queue icon), "Access Control" (with a lock icon), and "Bridges" (with a gear icon). At the bottom left, there is a section titled "Other Management Tools".

You will then see the **Broker Manager** Screen and on the left you will see a more advanced "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 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: `wss://montrealbroker.messaging.solace.cloud:443`
Message VPN: `montrealbroker-10-1`
Client UserName: `email-profile`
Client Password: `*****` <- provided during the course

The screenshot shows the SAP Solace Broker interface. On the left is a sidebar with a dark background and white text, listing various broker configurations and system components. The main area is titled "Send and Receive" and is divided into sections for "Publisher" and "Subscriber".

Publisher Section:

- Establish Connection:** Shows a status of "% Connected" and a "Broker URL" input field containing "wss://mr-connection-qhgik3f2ezp.messaging.solace.cloud:443".
- Message VPN:** Set to "montrealbroker-10-1".
- Clients:** Set to "montrealbroker-10-1".
- Queues:** Set to "email-profile".
- Connectors:** Set to ".....".
- Access Control:** Set to "Topic".
- Telemetry:** Set to "Replay".
- Replay:** Set to "Bridges".
- Bridges:** Set to "JMS JNDI".
- JMS JNDI:** Set to "try-me".
- Advanced Messaging:** Set to "Delivery Mode: Direct".
- Caches:** Set to "Message Content: Hello world!".
- Transactions:** Set to "try-me".

Subscriber Section:

- Establish Connection:** Shows a status of "% Connected" and a "Broker URL" input field containing "wss://mr-connection-qhgik3f2ezp.messaging.solace.cloud:443".
- Message VPN:** Set to "montrealbroker-10-1".
- Clients:** Set to "email-profile".
- Queues:** Set to ".....".
- Connectors:** Set to ".....".
- Access Control:** Set to "Topic".
- Telemetry:** Set to "Replay".
- Replay:** Set to "Bridges".
- Bridges:** Set to "JMS JNDI".
- JMS JNDI:** Set to "try-me".
- Advanced Messaging:** Set to "Delivery Mode: Direct".
- Caches:** Set to "Message Content: Hello world!".
- Transactions:** Set to "try-me".

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.

SAP Integration Suite

Integration Cards Plus Advanced | Integration Suite | Integration Suite | Service details | SAP Integration Suite | Send and Receive | Try Me | All Bookmarks

mr-connection-[sgigk3f2exp.messaging.solace.cloud:943/#/msg-vpm/5W9udHJIVWecm9eZXhMTAeMQ==/try-me/send-receive?count=20&cursor=did...](#)

MontrealBroker-10.1

Change VPN

Messaging

Message VPN

Clients

Queues

Connectors

Access Control

Telemetry

Replay

Bridges

DMS JNDI

Try Me

Advanced Messaging

Caches

Transactions

System

Clustering

Version 10.4.1.76

Send and Receive

Publisher

Establish Connection Connected Disconnect

Publish
Select a topic or queue to publish to:
 Topic Queue
try-me Show Advanced

Delivery Mode:
 Direct Persistent

Message Content:
Hello world!

Publish

Message is rejected with error X
Publish ACL Denied

Messages Published Clear Stats
1 Direct 0 Persistent

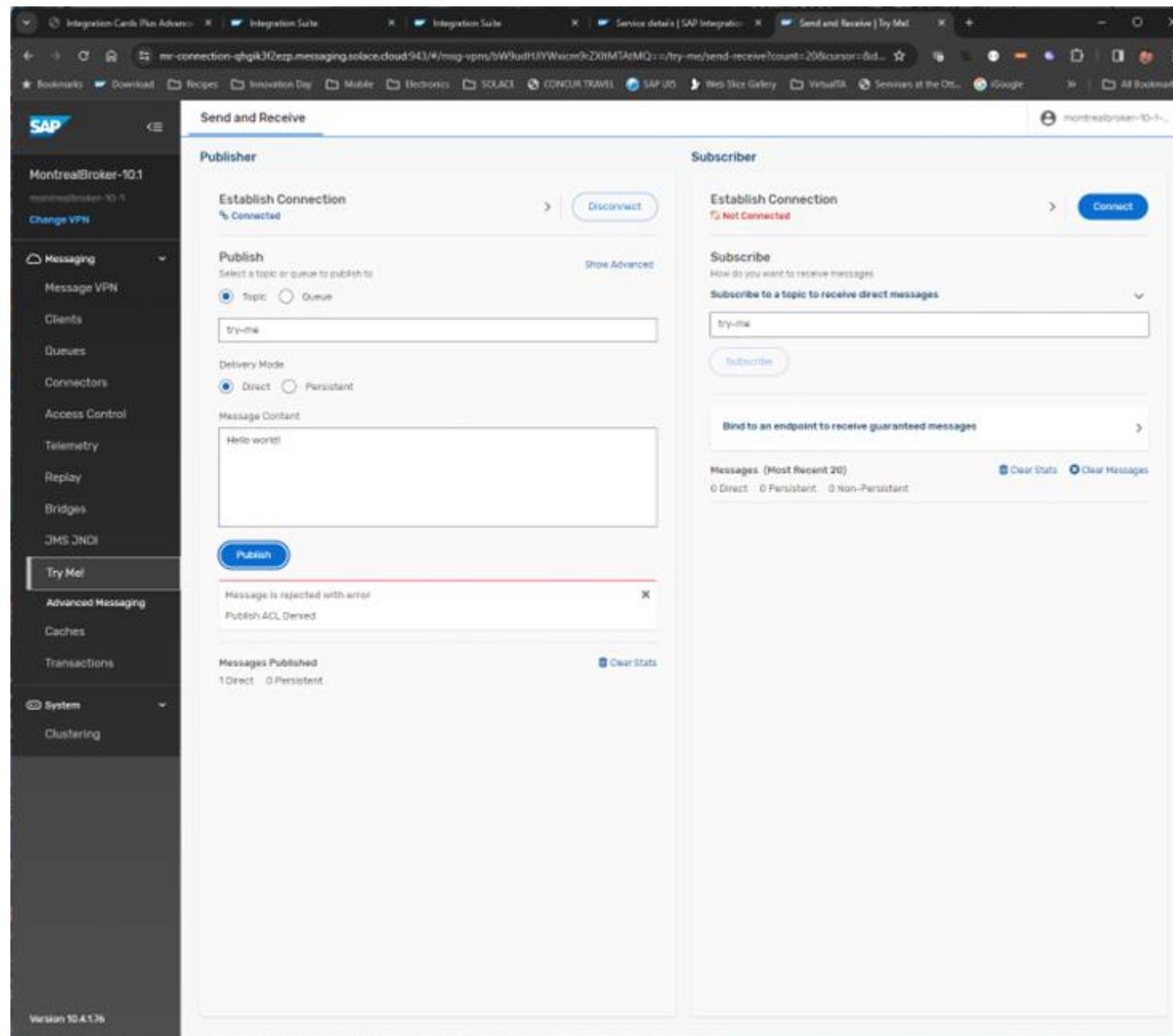
Subscriber

Establish Connection Not Connected Connect

Subscribe
How do you want to receive messages:
Subscribe to a topic to receive direct messages try-me

Bind to an endpoint to receive guaranteed messages >

Messages (Most Recent 20) Clear Stats Clear Messages
0 Direct 0 Persistent 0 Non-Persistent



Now, let's try the exact same thing with the subscription. Hit the "**Connect**" button, and you should see the connection properties already populated so accept this and hit connect.

Subscriber

Establish Connection

Not Connected

Broker URL Same as Publisher
wss://mr-connection-qhgik3f2ezp.messaging.solace.cloud:44

Message VPN Same as Publisher
montrealbroker-10-1

Client Username Same as Publisher
email-profile

Client Password
.....

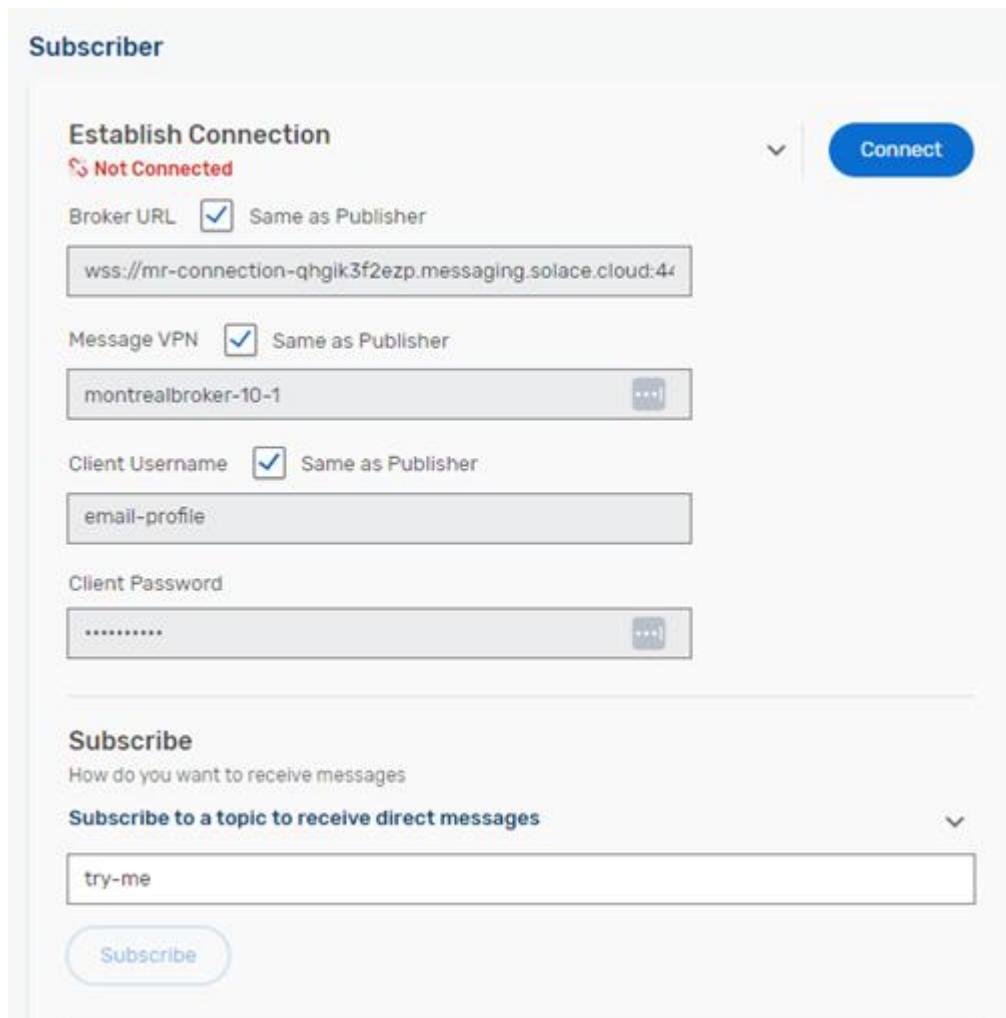
Subscribe

How do you want to receive messages

Subscribe to a topic to receive direct messages

try-me

Subscribe

The screenshot shows the Solace Cloud Subscriber interface. At the top, it says "Subscriber". Below that is a section titled "Establish Connection" with a status of "Not Connected". It includes fields for "Broker URL" (set to "Same as Publisher" with value "wss://mr-connection-qhgik3f2ezp.messaging.solace.cloud:44"), "Message VPN" (set to "Same as Publisher" with value "montrealbroker-10-1"), "Client Username" (set to "Same as Publisher" with value "email-profile"), and "Client Password" (redacted). Below this is a "Subscribe" section with a dropdown menu set to "Subscribe to a topic to receive direct messages". A single topic "try-me" is listed in the dropdown. At the bottom is a "Subscribe" button.

Once connected, you will see this:

The screenshot shows the MQTT.js Subscriber interface. At the top, it says "Subscriber". Below that, there's a section for "Establish Connection" with a progress bar labeled "% Connected" and a "Disconnect" button. The main area is titled "Subscribe" and asks "How do you want to receive messages". A dropdown menu is open, showing "Subscribe to a topic to receive direct messages" and a text input field containing "try-me". Below this is a "Subscribe" button. Another section below is titled "Bind to an endpoint to receive guaranteed messages". At the bottom, there's a summary of "Messages (Most Recent 20)" with counts for Direct, Persistent, and Non-Persistent messages, and buttons for "Clear Stats" and "Clear Messages".

From here, just hit the **Subscribe** button and you should see the following screen:

Subscriber

Establish Connection

% Connected



Disconnect

Subscribe

How do you want to receive messages

Subscribe to a topic to receive direct messages



Subscribe

Subscribed Topics

try-me

Subscription request failed with error



Subscription ACL Denied

Bind to an endpoint to receive guaranteed messages



Messages (Most Recent 20)

Clear Stats Clear Messages

0 Direct 0 Persistent 0 Non-Persistent

Again, you have been "**Denied**" - That was expected. 😊

Now, let's head back to the "Publisher" and change the "Topic" to `sap.com/emailnotification/created/v1` Use the following structure for your Message Content....be sure to copy the entire structure below including all of the curly braces. This is the structure that is passed to the Solace Event Mesh for processing. If successful, you should receive an email shortly after publishing with the information contained in the message. In the structure below, please replace "**YOUREMAILADDRESS**" with your actual email address prior to hitting the publish button.

```
{"orderHeader": [{"salesOrderNumber": "SO2958", "creator": "John Doe", "date": "2023-08-11", "salesType": "Online", "ordertype": "Expedited", "salesOrg": "SA03", "distributionChannel": "DC01", "division": "DV02", "netvalue": 423.76, "currency": "CAD", "customer": [{"customerId": "CUST008", "customerName": "scott", "zipCode": "13579", "street": "Seventh Avenue", "phone": "555-888-9999"}, {"country": "USA", "city": "Houston", "emailAddress": [{"email": "YOUREMAILADDRESS"}]}], "orderItem": [{"item": "ITEM013", "material": "MAT013", "materialType": "Product", "itemType": "Standard", "itemDescription": "Volt Electric bike", "orderSchedule": [{"scheduleNumber": "SCH013", "quantity": 40, "uom": "EA"}]}]}]
```

Send and Receive

Publisher

Establish Connection > Connected Disconnect

Publish Select a topic or queue to publish to Show Advanced

Topic Queue

sap.com/emailnotification/created/V1

Delivery Mode

Direct Persistent

Message Content

```
{"orderHeader": [{"salesOrderNumber": "SO2958", "creator": "John Doe", "date": "2023-08-11", "salesType": "Online", "orderType": "Expedited", "salesOrg": "SA03", "distributionChannel": "D001", "division": "DV02", "netValue": 423.76, "currency": "CAD"}, {"customer": [{"customerId": "CUST008", "customerName": "scott", "zipCode": "13579", "street": "Seventh Avenue", "phone": "555-888-9999", "country": "USA", "city": "Houston", "emailAddress": [{"email": "scott.dillon@solace.com"}]}]}, {"orderItem": [{"item": 1}]}]
```

Publish

Messages Published: 1 Direct 0 Persistent

Clear Stats

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' interface for an 'email-profile'. The 'Publish Topic' tab is selected. Under 'Publish Default Action', a dropdown menu is set to 'Disallow'. Below this, there is a search bar labeled 'Exceptions' with a placeholder 'Search by topic' and a clear button. Two checkboxes are present: 'Publish Exception Topic' and 'sap.com/emailnotification/created/V1', with the second one being checked and highlighted with a yellow border.

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.



ACL Profiles | email-profile

[Client Connect](#)[Publish Topic](#)[Subscribe Topic](#)[Subscribe Share Name](#)[Profile Users](#)

Subscribe Default Action

Disallow



Exceptions



Search by topic



Subscribe Exception Topic

Now that you have this understanding, you will see that this ACL profile could be used for a client that is only allowed to send EmailnotificationCreated events on the specified topic. Everything else is prohibited by the email-profile user ACL, so it can only be used for this single purpose.

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".

11. Takeaways

- How fine-grained security access in AEM works.

Thanks for participating in this codelab! Let us know what you thought in the [Solace Community Forum](#)! If you found any issues along the way we'd appreciate it if you'd raise them by clicking the Report a mistake button at the bottom left of this codelab.