

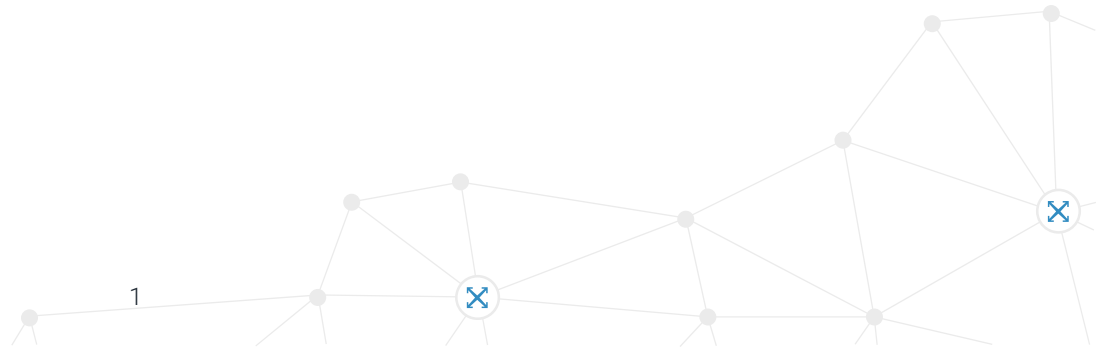


# SESSION 02

Edge Analytics Online Training

## CREATE YOUR FIRST FLOW

Work with modules and flows





# Session 2

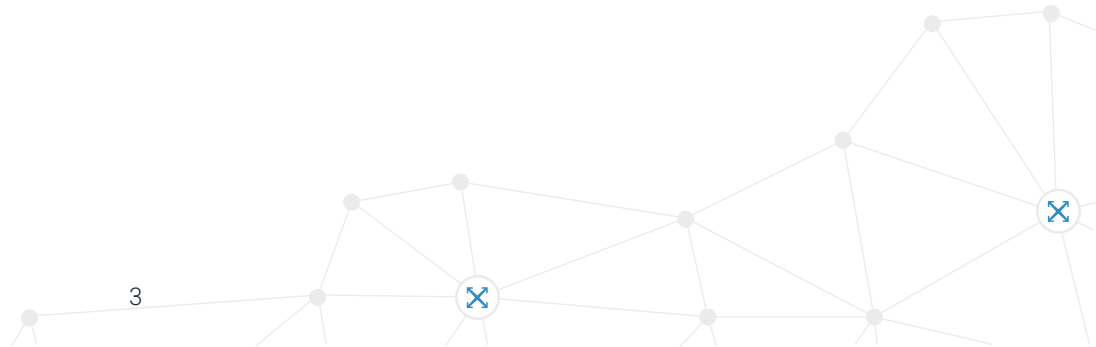
## Agenda

- UI introduction
  - Flows
  - The Flow Studio
  - Module settings
  - Testing and Deploying Flows
- Other functions
  - Message filtering
  - Managing Flows
- Modules covered:
  - Data Generator module
  - Aggregation module
  - Text Template module
  - MQTT Pub Client
- Exercise 1: Build your first flow – step by step



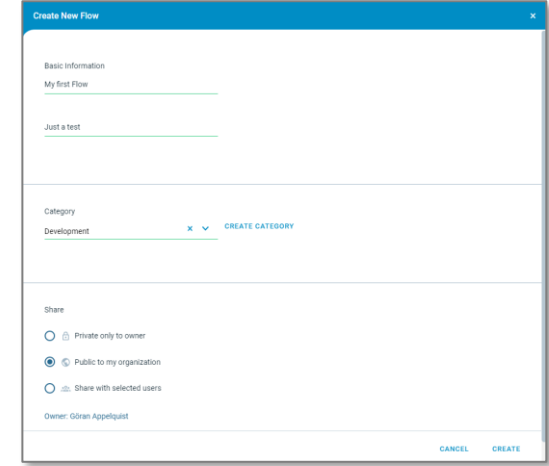
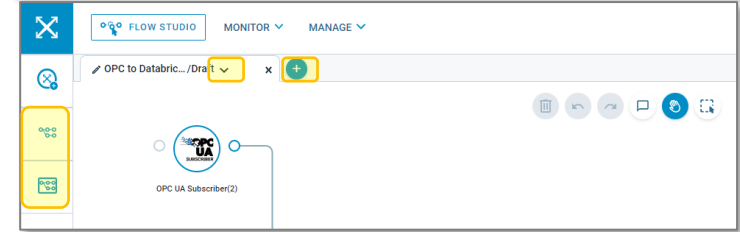
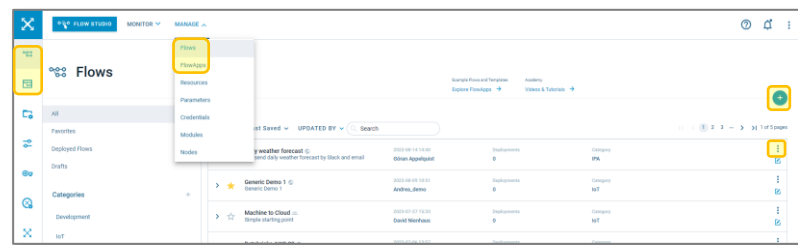
# FLOWS

This is what You create with Crosser



# Flows

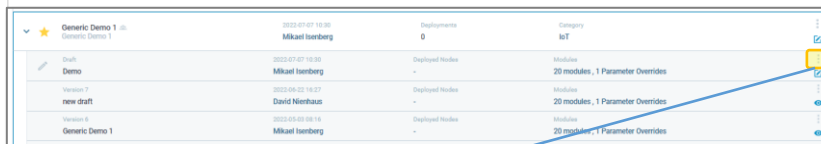
- Created on the [Flows page](#) or from within the [Flow Studio](#):
  - Using the “+” button (empty flow)
  - From a FlowApp
  - From an existing flow (menu on version/tab)
- Flows must have a unique name and a description (non-unique)
- From the Flows page:
  - Organize flows by adding them to a Category
  - Control who can access your flow through the sharing settings



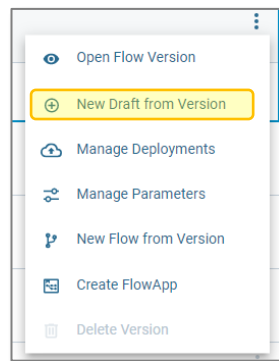
# Flow Versions

- Each flow can have any number of versions
- Versions are deployed to Nodes, not Flows
- Only the latest non-deployed version can be modified
- A version becomes read-only the first time it is deployed on a node
- Create new versions using the “New Draft from Version” action (only available on read-only flows)
- You can also create a new draft from a version inside the Flow Studio (tab menu)

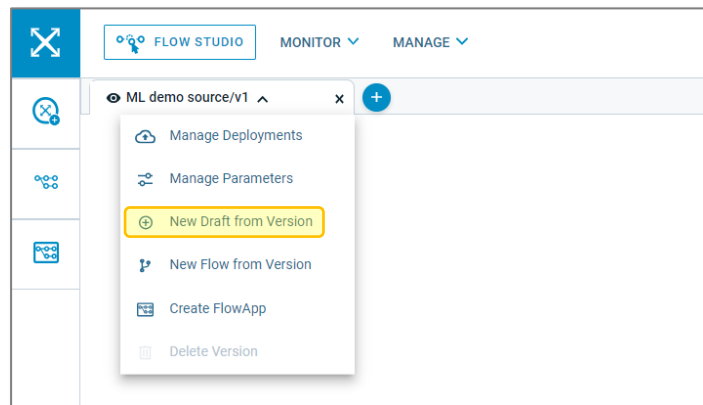
*Note: You will not be able to create new versions of your flows until you can deploy them on your own node in session 5*



Generic Demo 1	2023-07-07 10:30	Deployments	Category
Draft	2023-07-07 10:30	0	IoT
Demo	Mikael Isenberg	Deployed Nodes	Module
new draft	2023-06-20 16:27	+	20 modules, 1 Parameter Overrides
Version 7	2023-06-20 16:27	Deployed Nodes	Module
Version 6	2023-05-05 08:18	+	20 modules, 1 Parameter Overrides
Generic Demo 1	Mikael Isenberg	Deployed Nodes	Module
		+	20 modules, 1 Parameter Overrides



- Open Flow Version
- New Draft from Version**
- Manage Deployments
- Manage Parameters
- New Flow from Version
- Create FlowApp
- Delete Version



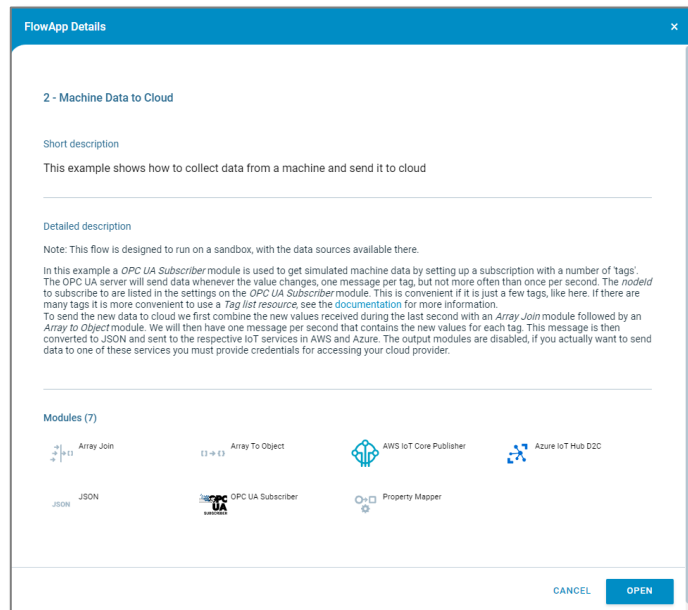
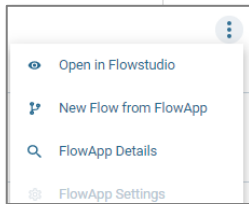
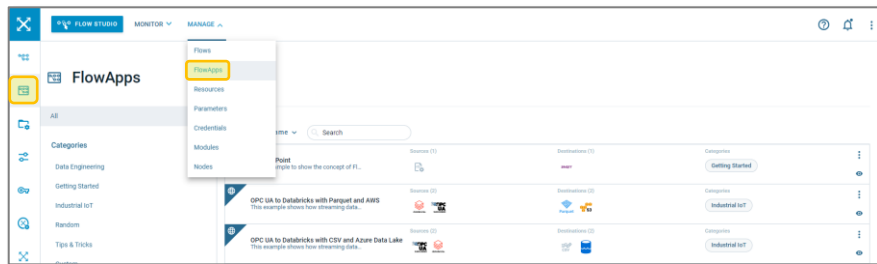
FLOW STUDIO MONITOR MANAGE

ML demo source/v1

- Manage Deployments
- Manage Parameters
- New Draft from Version**
- New Flow from Version
- Create FlowApp
- Delete Version

# FlowApps

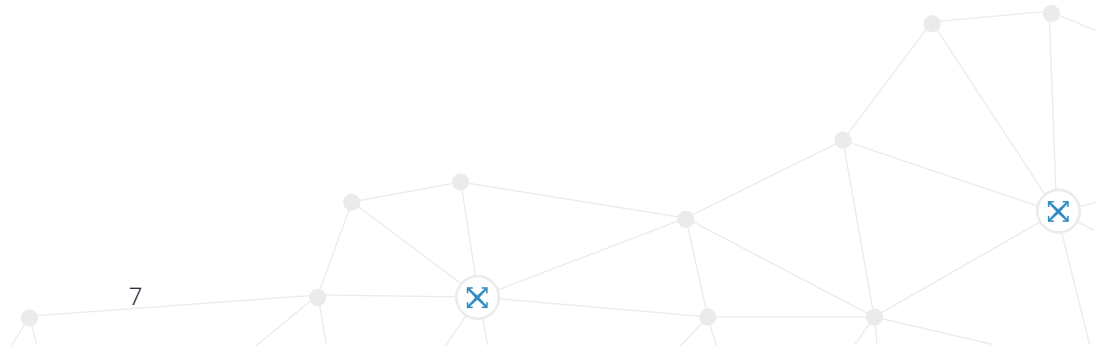
- Pre-built templates to use as starting points for new flows
  - Created by Crosser
  - Created by your organisation
- Browse to find a FlowApp that is close to what you need
  - Read description
  - Open the flow in Flow Studio
- Create a flow from a FlowApp by:
  - Using the “New Flow from FlowApp” action in the menu
- FlowApps can also be opened from inside the Flow Studio





# THE FLOW STUDIO

This is where you design and test your flows



# The Flow Studio

## Design and Test Your Flows

The screenshot displays the Flow Studio interface. At the top, there's a navigation bar with 'FLOW STUDIO', 'MONITOR', and 'MANAGE' options. Below this, a breadcrumb shows 'Zoho tickets to ... /Draft'. The main workspace is divided into three sections:

- Module Library (Left):** A sidebar with a search bar and categories like 'Most Used', 'Triggers', 'Sources', 'Cloud', and 'Databases'. It lists various database connectors such as Couchbase, Firebird, Influx, MariaDB, MongoDB, and MySQL.
- Flow Diagram (Center):** A complex workflow graph with nodes and connecting arrows. Nodes include triggers like 'Every hour, 15 past' and 'Search Contacts', and actions like 'Time Now', 'Time one year back', 'Set Date Range', 'Zoho Desk Search Tickets', 'LOOP OVER TICKETS', 'Test Templates()', 'Check if Zoho case number is set in SP', 'Create search criteria for contacts in SP', 'Remove extra data', 'Check if email missing the first part of the month', 'Test Templates', 'Array join', 'Array Priority Sort', 'JSON', 'Send message using in SP to Message', 'REMOVE ARRAY', 'Outputs', 'Create Case', 'Set SPDC tickets in Zoho', and 'Zoho Desk Update Ticket'.
- Test & Debug (Right):** A panel for testing the flow. It shows a 'Debug' tab with 'SHOW ALL', 'FREEZE', and 'CLEAR' buttons. Below, a 'Zoho Desk Search Tickets - Out' window displays a JSON response:

```
{
  "crosser": {
    "success": true
  },
  "fromDate": "2023-07-06T12:41:22.963Z",
  "request": {
    "headers": {
      "orgId": "20070280589",
      "Authorization": "VALUE EXCLUDED"
    },
    "url": "https://desk.zoho.eu/api/v1/tickets/search?createdTimeRange=2023-07-06T12:41:22.963Z,2023-07-13T12:41:22.951Z&limit=100",
    "verb": "GET"
  },
  "toDate": "2023-07-13T12:41:22.951Z",
  "zh_search": {
    "body": {
      "count": 7,
      "data": [
        {
          "accountId": null,
          "assignee": null,
          "assignee": null,
          "assigneeId": null,
          "category": null,
          "cf": {
            "inf_minimal_ticket_conf": null
          }
        }
      ]
    }
  }
}
```



# The Flow Studio

## Side Panels

Module library  
Flows  
FlowApps

The screenshot displays the Flow Studio interface. On the left is the **Module Library** panel, which includes a search bar and a list of categories: Most Used, Triggers, Sources, Cloud, and Databases. Under Databases, various database connectors are listed, such as Couchbase Executor, Firebird Select, Influx Select, MariaDB Executor, MariaDB Select, MongoDB Select, MySQL Executor, MySQL Select, MySQL Stored Procedure Read, MySQL Executor, MySQL Select, and Oracle Database Select. The central canvas shows a complex workflow with nodes like 'Every Hour: 10 last', 'Time Now', 'Time one week back', 'Get Data Range', 'Zoho Desk Search Tickets', 'LOOP OVER TICKETS', 'Test Template()', 'Check if Zoho Desk Number is valid in SP', 'CHECK search criteria for tickets in SP', 'Remove invalid data', 'Search Contacts', 'Check if email missing the full part of the phone', 'Test Template', 'Array join', 'Array Property Out', 'JSON', 'Base missing user in SP to Vlookup', 'REMOVE ARRAY', 'Counts', 'Create Date', 'Get BFCO tickets in Zoho', and 'Zoho Desk Update Ticket'. On the right is the **Test & Debug** panel, which shows a JSON response for 'Zoho Desk Search Tickets - Out'. The response includes fields like 'success', 'fromDate', 'request', 'headers', 'url', 'createdTimeRange', 'toDate', 'zoh\_search', 'body', and 'data'. The 'data' field contains an array of ticket objects with fields like 'accountid', 'assignee', 'assigneeid', 'category', 'cf', and 'cf\_email\_task\_url'. Below the JSON is a 'Test & Debug' section with buttons for 'SHOW ALL', 'FREEZE', and 'CLEAR', and a 'Test & Debug' section with buttons for 'DEBUG', 'CONNECT', 'SHOW ALL', 'FREEZE', and 'CLEAR'. At the bottom right of the panel are 'SAVE' and 'EXIT' buttons.

Test & Debug  
Notifications  
Flow version settings  
Resources  
Annotations

# The Flow Studio

## Module Library

The screenshot displays the Flow Studio interface with a workflow diagram in the center, a Module Library on the left, and a Test & Debug panel on the right.

**Module Library:** A sidebar on the left contains a search bar and a list of categories. The 'Databases' category is expanded, showing various database connectors. A text overlay reads: "Find the modules you need".

- Categories: Most Used, Triggers, Sources, Cloud, Databases
- Databases: Influx Select, MariaDB Executor, MariaDB Select, MongoDB Select, MsSQL Executor, MsSQL Select, MsSQL Stored Procedure Read, MySQL Executor, MySQL Select, Oracle Database Select

**Workflow Diagram:** A central workspace shows a complex flowchart with nodes and connections. The nodes include:

- Every hour: 10 past
- Time Now
- Time one week back
- Set Date Range
- Zoho Desk Search Tickets
- LOOP OVER TICKETS
- Text Template(2)
- Check Zoho case number is set in SP
- Check search string for contacts in SP
- Remove extra data
- Search Contacts
- Check if email missing the 1st day of the month
- Text Template
- Array Join
- Array Property Set
- JSON
- Send missing case in SP to Manager
- REMOVE ARRAY
- Casearb
- Create Case
- Set SPID tickets in Zoho
- Zoho Desk Update Ticket

**Test & Debug Panel:** On the right, a 'Test & Debug' window is open, showing a JSON response for 'Zoho Desk Search Tickets - Out'. The JSON includes fields like 'crosser', 'success', 'fromDate', 'request', 'headers', 'url', 'toDate', and 'zh\_search'.

```
{
  "crosser": {
    "success": true
  },
  "fromDate": "2023-07-06T12:41:22.963Z",
  "request": {
    "headers": {
      "origin": "20070280589",
      "Authorization": "VALUE EXCLUDED"
    },
    "url": "https://desk.zoho.eu/api/v1/tickets/search?createdTimeRange=2023-07-06T12:41:22.963Z,2023-07-13T12:41:22.951Z&limit=100",
    "verb": "GET"
  },
  "toDate": "2023-07-13T12:41:22.951Z",
  "zh_search": {
    "count": 7,
    "data": [
      {
        "accountId": null,
        "assignee": null,
        "assigneeId": null,
        "category": null,
        "cf": {

```

# The Flow Studio

## Drawing Canvas

The screenshot displays the Flow Studio interface. On the left is the **Module Library** with a search bar and categories: Most Used, Triggers, Sources, Cloud, and Databases. The Databases category is expanded, listing various database executors like Couchbase, Firebird, Influx, MariaDB, MongoDB, and MySQL. The central **Drawing Canvas** is highlighted with a yellow border and contains a workflow diagram with nodes such as 'Every hour: 10 past', 'Time Now', 'Time one week back', 'Get Date Range', 'Zoho Desk Search Tickets', 'LOOP OVER TICKETS', 'Text Templates', 'Check if Zoho case number is set in SP', 'Check search criteria for context in SP', 'Remove extra data', 'Search Contexts', 'Check if email missing the top bar of the message', 'Array Join', 'Array Property Get', 'JSON', 'Send missing cases in SP to Service', 'REMOVE ARRAY', 'Deserialize', 'Create Class', 'Set SPOC tickets in Zoho', and 'Zoho Desk update Ticket'. A text overlay in the center of the canvas reads: "Add modules, connect and configure them". On the right is the **Test & Debug** panel, showing a 'Debug' tab with a 'Test & Debug' button and a 'Connect' button. Below this is a 'SHOW ALL' dropdown, 'FREEZE', and 'CLEAR' buttons. A 'Zoho Desk Search Tickets - Out' window displays a JSON response:

```
{
  "crosser": {
    "success": true
  },
  "fromDate": "2023-07-06T12:41:22.963Z",
  "request": {
    "headers": {
      "origin": "20070280589",
      "Authorization": "VALUE EXCLUDED"
    },
    "url": "https://desk.zoho.eu/api/v1/tickets/search?createdTimeRange=2023-07-06T12:41:22.963Z,2023-07-13T12:41:22.951Z&limit=100",
    "verb": "GET"
  },
  "toDate": "2023-07-13T12:41:22.951Z",
  "zh_search": {
    "body": {
      "count": 7,
      "data": [
        {
          "accountId": null,
          "assignee": null,
          "assigneeId": null,
          "category": null,
          "id": 1
        }
      ]
    }
  }
}
```

# The Flow Studio

## Test & Debug

The screenshot displays the Flow Studio interface. On the left is a 'Module Library' with categories like 'Most Used', 'Triggers', 'Sources', 'Cloud', and 'Databases'. The main workspace shows a flow diagram with nodes such as 'Every hour: 10 past', 'Time Now', 'Time one week back', 'Get Date Range', 'Zoho Desk Search Tickets', 'LOOP OVER TICKETS', 'Text Template', 'Check if Zoho Desk member is set in DP', 'Create Search criteria for contacts in DP', 'Removing extra data', 'Search Contacts', 'Check if email matching the 1st day of the month', 'Text Template', 'Array Join', 'Array Priority Set', 'JSON', 'Send missing cases in DP to Maria', 'REMOVE DUPLICATES', 'CaseId', 'Create Case', 'Get EPIC tickets in Zoho', and 'Zoho Desk Update Ticket'. On the right, a 'Test & Debug' window is open, showing a JSON output for 'Zoho Desk Search Tickets - Out'. The output includes fields like 'crosser', 'success', 'headers', 'url', 'statusCode', 'body', and 'data'.

```
{
  "crosser": {
    "success": true
  },
  "headers": {
    "Content-Type": "application/json",
    "Authorization": "VALUE EXCLUDED"
  },
  "url": "https://desk.zoho.eu/api/v1/tickets/search?createdTimeRange=2023-07-06T12:41:22.963Z,2023-07-13T12:41:22.951Z&limit=100",
  "verb": "GET",
  "statusCode": 200,
  "body": {
    "count": 7,
    "data": [
      {
        "accountId": null,
        "assignee": null,
        "assigneeId": null,
        "category": null,
        "cf": {
          "cf_email_address": null
        }
      }
    ]
  }
}
```

# The Flow Studio

## Manage Your Flows

The screenshot displays the Flow Studio interface. On the left, a sidebar lists sources under 'Sources', 'Cloud', and 'Databases'. A yellow overlay titled 'Manage your Flow' is positioned over the sidebar, containing options: 'Manage Deployments', 'New Flow from Draft', and 'Delete Draft'. The main workspace shows a flow diagram with nodes such as 'Every Hour: 10 past', 'Time Now', 'Time one week back', 'Get Date Range', 'Zoho Desk Search Tickets', 'LOOP OVER TICKETS', 'Test Template()', 'Check if Zoho case number is set in SF', 'Check search criteria for contacts in SF', 'Remove node data', 'Search Contacts', 'Check if email meeting the 1st day of the month', 'REMOVE ARRAY', 'CreateIt()', 'Create Case', 'Get SFDC tickets in Zoho', 'Zoho Desk Update Ticket', 'Array Join', 'Array Property Get', 'JOIN', and 'Send meeting cases in SF to Hubspot'. On the right, the 'Test & Debug' panel is active, showing a 'Debug' tab and a 'Connect' button. Below, a 'Zoho Desk Search Tickets - Out' node displays a JSON response:

```
{
  "crosser": {
    "success": true
  },
  "fromDate": "2023-07-06T12:41:22.963Z",
  "request": {
    "headers": {
      "orgId": "20070280589",
      "Authorization": "VALUE EXCLUDED"
    },
    "url": "https://desk.zoho.eu/api/v1/tickets/search",
    "toDate": "2023-07-13T12:41:22.951Z",
    "pk_search": {
      "body": {
        "count": 7,
        "data": [
          {
            "accountId": null,
            "assignee": null,
            "assigneeId": null,
            "category": null,
            "cf": {
```

# The Flow Studio

## Create New Flows or Open Existing

The screenshot displays the Flow Studio interface with the following components:

- Top Bar:** Includes a search bar for "FLOW STUDIO", "MONITOR" and "MANAGE" dropdowns, and a tab for "External2" with a 90% completion indicator.
- Left Panel (Flows):** A list of flows under the "Marketing" category. A search bar is at the top. A note at the bottom states: "The list is limited but you can still search for all available Flows".
- Center Panel (Flow Diagram):** Shows three flow diagrams:
  - Flow 1:** "one week back" -> "Set Date Range" -> "Zoho Desk Search Tickets".
  - Flow 2:** "Text Template" -> "Check if Zoho case number is set in SF" -> "Create search criteria for Contacts in SF" -> "Remove meta data".
  - Flow 3:** "if email missing day of the month" -> "Text Template(2)" -> "Array Join" -> "Array Property Get" -> "JSON" -> "JSON" -> "Send missing users in HF to Marcus".
- Right Panel (Test & Debug):** Shows a "Debug" tab with "SHOW ALL", "FREEZE", and "CLEAR" buttons. A message states "The flow is currently not running". Below is a "Search Contacts - Out" JSON response:

```
{
  "channel": "Email",
  "classification": "Spam-advertising",
  "crosser": {
    "success": true
  },
  "email": "maria.gordan@leadinfosolution.tech",
  "id": "44473000008318067",
  "request": {
    "headers": {
      "Authorization": "VALUE EXCLUDED"
    },
    "url": "https://eu19.salesforce.com/services/data/v53.0/parameterizedSearch?q=maria.gordan@leadinfosolution.tech&object=Contacts&contact.fields=id",
    "verb": "GET"
  },
  "search": {
    "body": {
      "searchRecords": [
      ]
    }
  },
  "contentType": "application/json",
  "statusCode": 200
}
```

# The Flow Studio

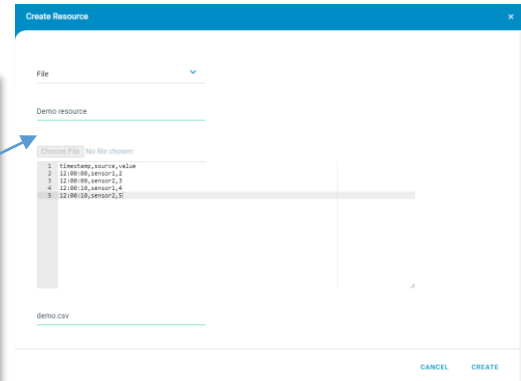
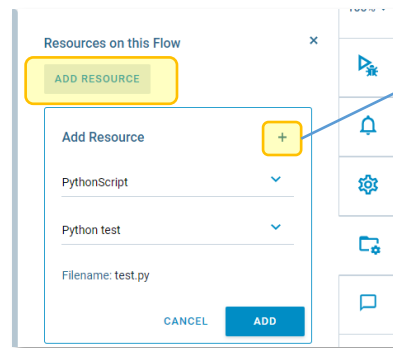
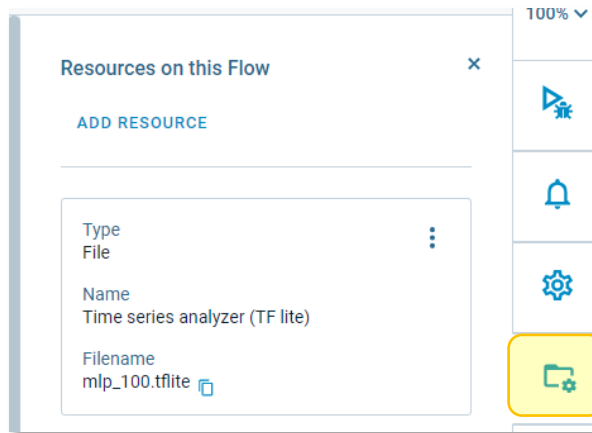
## Visual Appearance

The screenshot displays the Flow Studio interface with the following components:

- Top Bar:** Includes a search bar for "FLOW STUDIO", "MONITOR" and "MANAGE" dropdowns, and a "Zoho tickets to ... /Draft" tab.
- Module Library (Left):** A sidebar with a search bar and categories: "Most Used", "Triggers", "Sources", "Cloud", and "Databases". Under "Databases", various database connectors are listed, such as Couchbase, Firebird, Influx, MariaDB, MongoDB, and Oracle.
- Workflow Diagram (Center):** A visual flowchart with nodes and connecting lines. A yellow box highlights a set of control icons (undo, redo, multi-select, annotations) at the top. Another yellow box highlights a "Zoho Desk Lookup Ticket" node at the bottom right. The diagram includes nodes like "Every hour: 10 past", "Time Now", "Time one week back", "Get Date Range", "Zoho Desk Search Tickets", "LOOP OVER TICKETS", "Text Template", "Check if Zoho case number is set in EP", "Check if search criteria for contacts in EP", "Remove extra data", "Search Contacts", "Check if email missing the last day of the month", "Text Template", "Array Join", "Array Priority Out", "JSON", "Send missing cases in EP to MariaDB", "REMOVE action", "Queue", "Create Case", and "Get EPID created in Data".
- Test & Debug (Right):** A panel with "Debug" and "Connect" tabs. It shows a "Zoho Desk Search Tickets - Out" output with a JSON response. The JSON includes fields like "success", "fromDate", "request", "headers", "url", "createdTimeRange", "toDate", "zh\_search", "body", "count", "data", and "cf".

# Resources

- Add resources from the library that your flow needs:
  - Files
  - ML models
- Create new resources by uploading files or enter data in the editor
- Resources added to a flow will be downloaded by nodes when the flow is deployed
- *Note: PLC resources are added from the respective module settings*





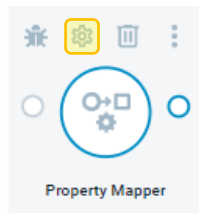


# USING MODULES IN THE FLOW STUDIO

# Modules

## Settings tab

- The *Settings* tab shows module specific settings
  - Any setting called something with *Property* references messages. Which data to use from the incoming message (typically called *Source* or *Value* properties) and where to add the result on the output message (*Target* property)
  - I/O modules have settings to specify how to access external systems
  - Analytics modules have settings to control the processing of the selected data
- Modules have versions:
  - When dragging a module from the library the latest version is used
  - Multiple instances of the same module in a flow must have the same version
  - Different flows can use different versions of a module
  - The module and versions available depends on the Node version you are targeting (advanced feature)

A screenshot of the 'Property Mapper' settings dialog. The dialog has a title bar with a close button (X) and a subtitle 'Property Mapper'. Below the title bar are three tabs: 'Settings' (highlighted with a yellow border), 'Common', and 'Documentation >'. The main content area is divided into sections:

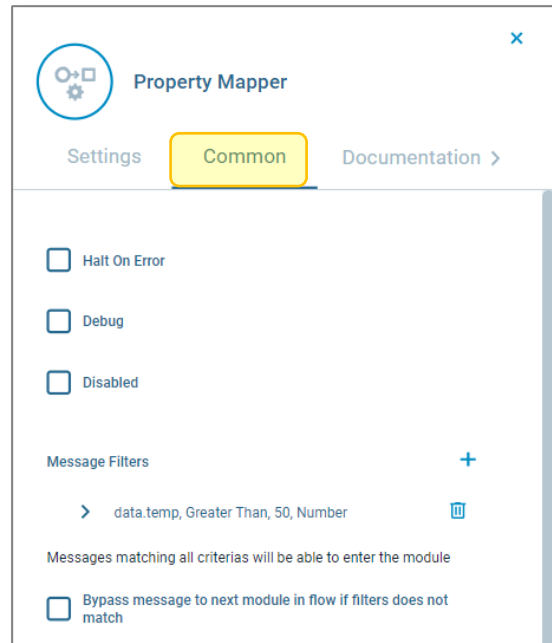
- Name:** Property Mapper
- Version:** 3.0.1 (with a dropdown arrow)
- Keep Properties:** A checked checkbox. Below it, text reads: 'If true all properties not renamed or removed will be included in the output. False will only include properties renamed or added'.
- Move Properties:** A section with a '+' icon on the right. It contains a table with two columns: 'data.id' and 'data.name'. Below the table, text reads: 'Left is current name, right is new name'.
- Remove Properties:** A section with a '+' icon on the right. Text below reads: 'The name of the property to remove.'
- Add Properties:** A section with a '+' icon on the right. It contains a table with two columns: 'source' and 'mqtt'.

# Modules

## Common tab

- The same on all modules
- Flow behavior settings
  - Halt On Error
    - If enabled the flow will stop on module errors
    - If disabled, the behavior is controlled by the flow setting
  - Debug
    - Enable debug output from this module when testing (same as debug action on the module icon)
  - Disabled
    - “Turn off” a module, e.g. to prevent data to be sent externally during testing
- Message filters
  - Without filters all messages will be processed
  - With filters only selected messages will be processed
  - Messages not matching the filter can either be ignored or bypassed to the next module (“Bypass...” setting)

*Note: Queue/Retry settings will not be covered in the Fundamentals course*



# Modules

## Documentation tab

- Module documentation
  - General description
  - Settings
  - Message requirements
  - Examples
- Release Notes
  - Describes changes in module versions

The screenshot shows a web interface for the 'Property Mapper' module. At the top, there are two tabs: 'Documentation' (selected) and 'Release Notes'. Below the tabs, the 'Documentation' section is active, showing the following content:

**Property Mapper**

The module is used to manipulate properties on an existing object. The operations are executed in the order: Move, Remove, Add.

This module is designed to be located in the middle of a flow.

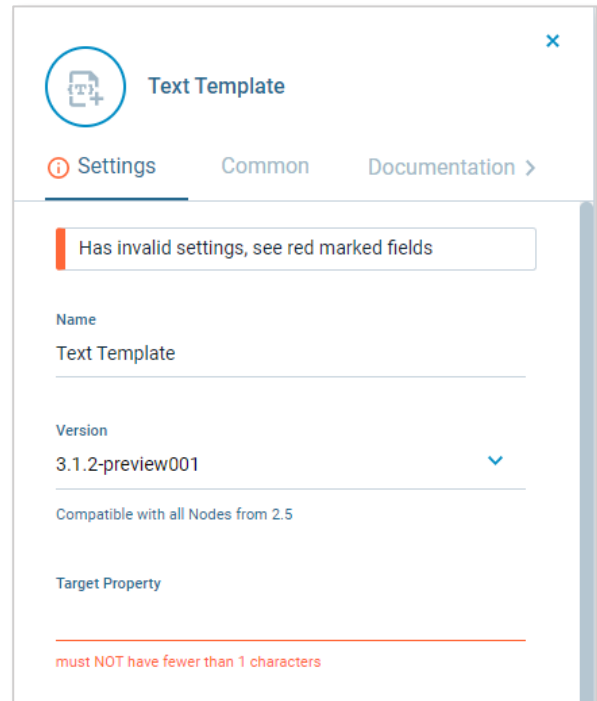
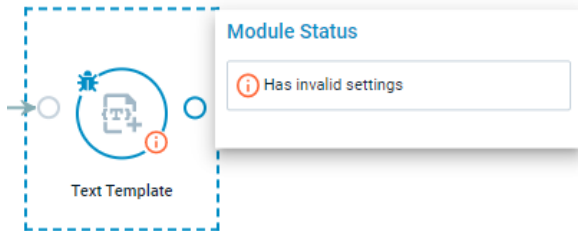
**Settings**

Name	Requirements	Purpose	Default
Keep Properties	true/false	If true all properties not renamed or removed will be included in the output. False will only include properties renamed or added	
Move	Dictionary string,string	Left is current name, right is new name	
Remove	string	The name of the property to remove.	
Add	Dictionary string,string	Left is name, right is value	

# Modules

## Settings validations

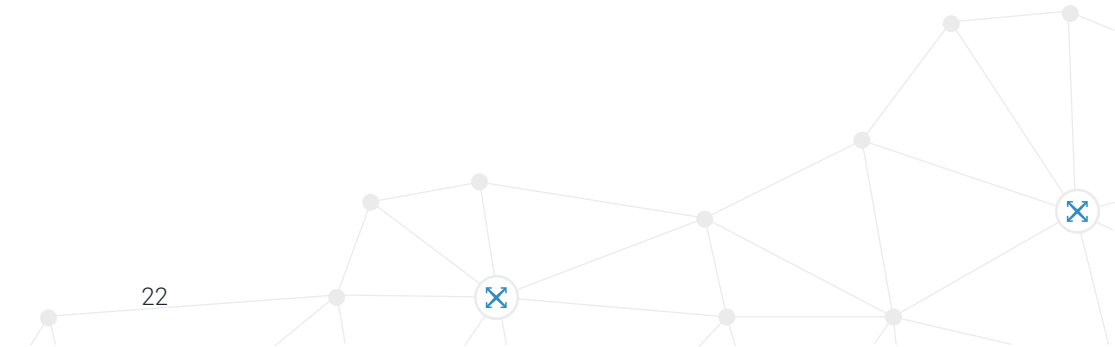
- Settings entered in modules are validated
- Invalid or incomplete settings:
  - Are indicated on the module in the canvas
  - In the Settings panel for the module





# TESTING YOUR FLOW

In the Flow Studio  
Deploy to a Node



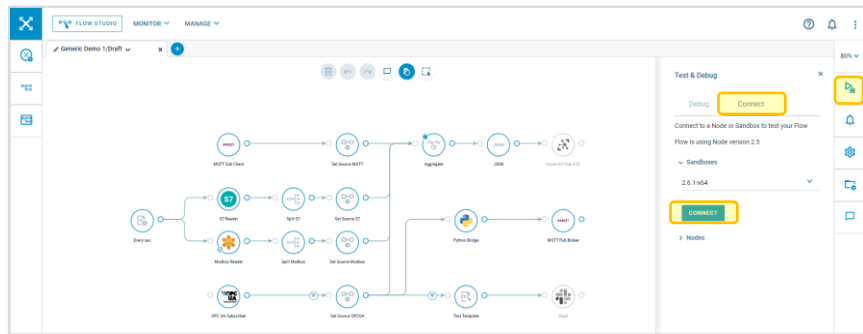
# Test Flow

## Connect to a Node

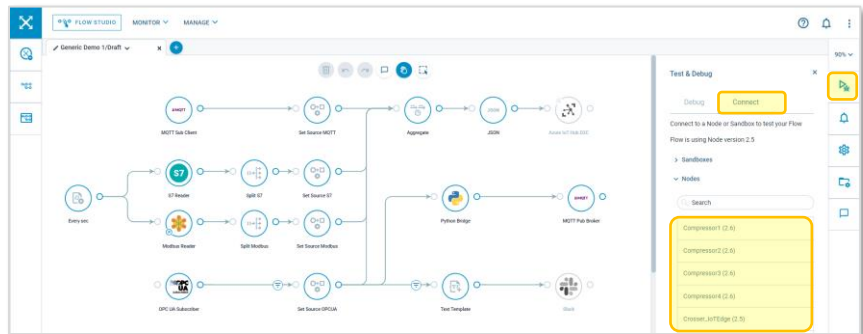
- Connect to a node to use for testing
  - **Sandbox:** Node hosted by Crosser. Can be used with flows that don't need access to local data sources. Only available in the Flow Studio
  - **Local nodes:** Nodes running on your infrastructure

*Note: Exercises can be tested on a Sandbox, if not explicitly referring to your local node*

## Sandbox



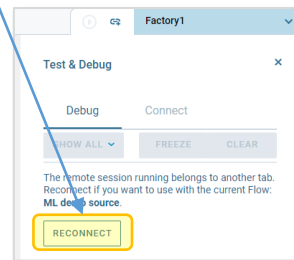
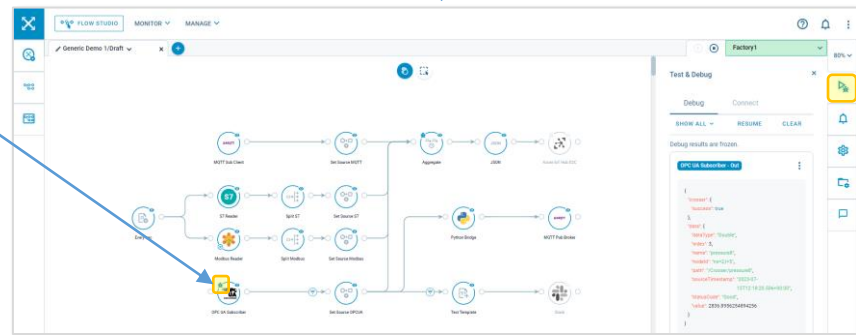
## Local Nodes



# Test Flow

## Run Your Flow and Check Results

- Start (and Stop) the flow
  - The flow must be saved before a remote session can be started. This can be done automatically, if you chose to enable that in the Save dialog
- Enable debugging on modules to see output in the Debug window (hover and use the debug action)
- A remote session is connected to a specific Flow version. You need to *Reconnect* to test another Flow version.
- The Flow is removed from the Node when you leave the Flow Studio

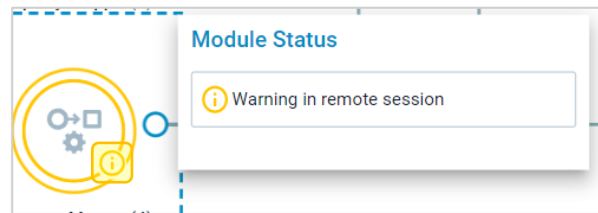
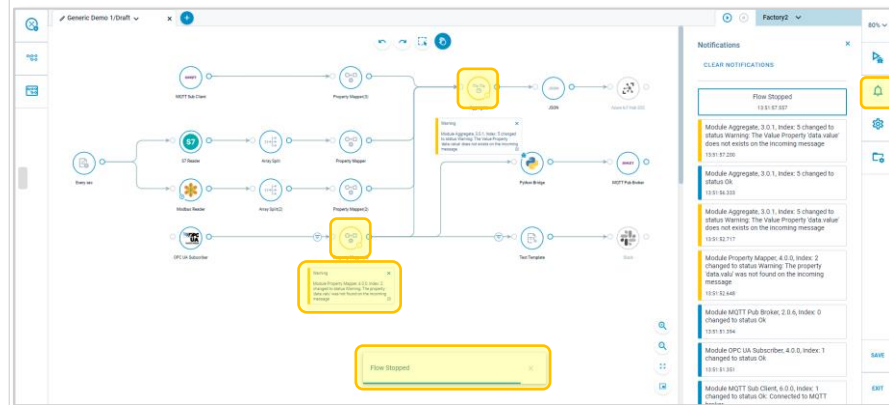




# Test Flows

## Notifications

- Notifications from the Flow will show up as *toasters* (bottom of screen):
  - Blue: Information and status
  - Yellow: Warnings
  - Red: Errors (flow will stop)
- Errors and Warnings from modules will be added on the module causing the event, as a colored ring, with the message below (large messages can be opened in a separate window)
- Notifications are also added to the **Notifications** tab and to the notification list on modules
- When the flow is deployed notifications will be sent to the Event log in the Control Center

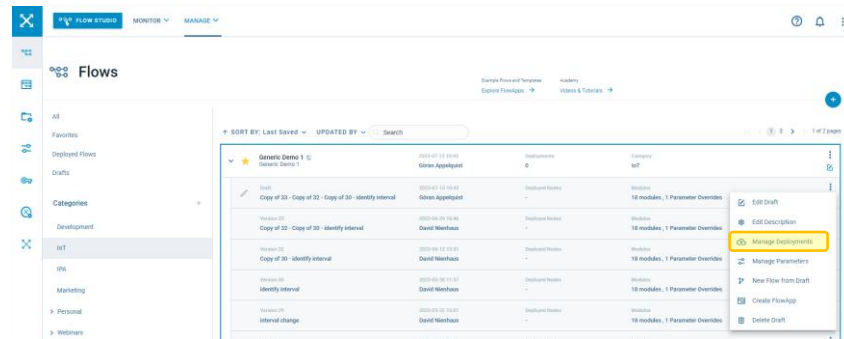
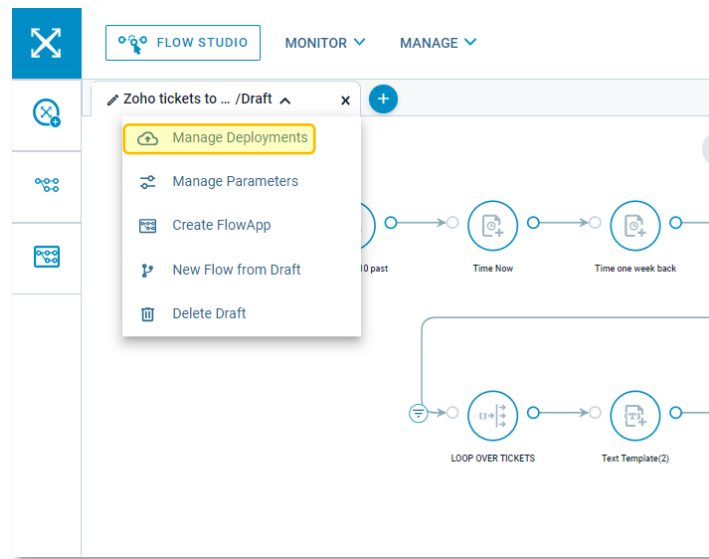


# Deploy Flows

## Install permanently

Open the [Deployment tool](#)

- In the Flow Studio:
  - Select the [Manage Deployments](#) action in the tab menu
- On the Flows page:
  - Expand a flow to see the versions
  - Select the [Manage Deployments](#) action in the menu, or click in the [Deployments](#) column



# Deploy Flows

## Select Node(s)

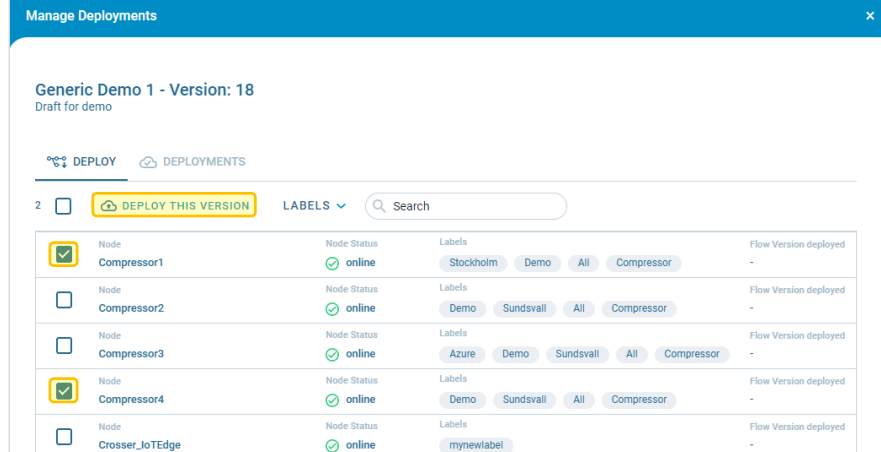
In the Deployment tool

- In the list of available Nodes, selects the ones where you want to install the Flow
- If the Node is currently offline, it will install the Flow the next time it comes online
- Click on Deploy This Version

*Note 1: After a Flow has been deployed to Nodes, processing will run locally on those Nodes without any dependencies on the Crosser Control Center service*

*Note 2: Deploying a Flow will make it read-only. To update, create a new version*

*Note 3: You will not be able to deploy a flow until session 5, where you will install your own node*



The screenshot shows the 'Manage Deployments' interface for 'Generic Demo 1 - Version: 18'. The interface includes a 'DEPLOY' button (highlighted in yellow) and a 'DEPLOYMENTS' tab. Below the tabs is a search bar and a 'LABELS' dropdown menu. The main content is a table listing nodes and their deployment status.

Node	Node Status	Labels	Flow Version deployed
Compressor1	online	Stockholm Demo All Compressor	-
Compressor2	online	Demo Sundsvall All Compressor	-
Compressor3	online	Azure Demo Sundsvall All Compressor	-
Compressor4	online	Demo Sundsvall All Compressor	-
Crosser_IoTEdge	online	mynewlabel	-



# MODULES USED IN EXERCISES

Data Generator  
Aggregate  
Text Template  
SMTP Send

# MODULE

## Data generator

- Create test data (from JSON template)
- Useful for testing a flow without using external data sources
- Multiple “Samples” can be generated, e.g. to simulate multiple sensors or sources
- Enter a JSON template with the message structure you want to generate → Click “Update”
- In the “Data Rules” section, specify how you want to generate data for each property in the template

Integers/GUIDs can be:

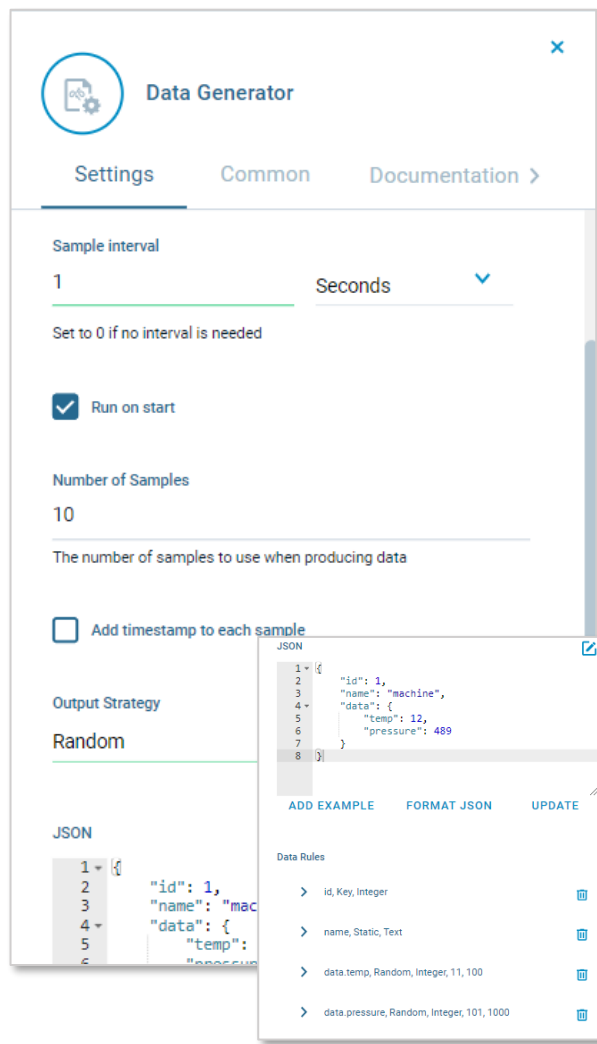
- Static → Value from template
- Key → Value unique per sample
- Random

Text can be:

- Static → Value from template
- Identifier → The template value plus a key added per sample

Double/Booleans can be Static or Random

- Change “Output Strategy” to generate an ordered sequence of IDs, or an array of samples

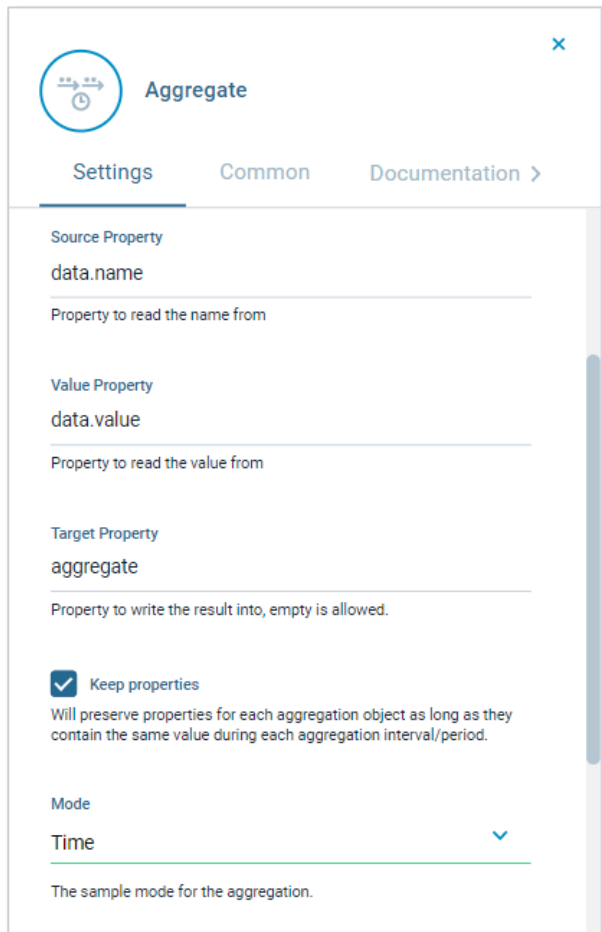


# Module

## Aggregate

- The Aggregate module is used to aggregate values:
  - Over a time window
  - Over a certain number of messages
  - An output is generated when the end of the aggregation window is reached
- Calculates Average, Min and Max values
- Can group messages by source, i.e. aggregate data from multiple sources

*Note: Use this module with streaming data. If you want to operate on data in an array, use the Array Statistics module.*



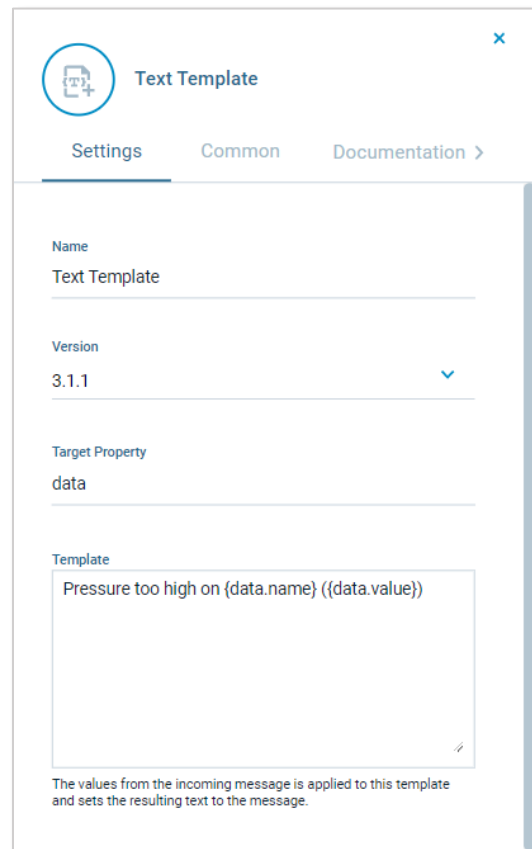
The screenshot displays the configuration interface for the 'Aggregate' module in the Apache Kafka UI. At the top, there is a circular icon with a refresh symbol and the title 'Aggregate'. Below the title are three tabs: 'Settings' (which is selected), 'Common', and 'Documentation >'. The main configuration area is divided into several sections:

- Source Property:** The property is set to 'data.name'. Below it, a label reads 'Property to read the name from'.
- Value Property:** The property is set to 'data.value'. Below it, a label reads 'Property to read the value from'.
- Target Property:** The property is set to 'aggregate'. Below it, a label reads 'Property to write the result into, empty is allowed.'
- Keep properties:** This option is checked with a blue checkmark. Below it, a description reads: 'Will preserve properties for each aggregation object as long as they contain the same value during each aggregation interval/period.'
- Mode:** The mode is set to 'Time', indicated by a dropdown arrow on the right. Below it, a label reads 'The sample mode for the aggregation.'

# Module

## Text Template

- Create text strings based on a template where data from the flow message can be inserted
- Anything inside {} is replaced with the value from the corresponding property on the incoming message
- Use it to:
  - Add/append text to messages
  - Convert numeric values to strings



The screenshot shows the 'Text Template' configuration window. It has a title bar with a close button (X) and a circular icon containing a document with a plus sign. Below the title bar are three tabs: 'Settings' (selected), 'Common', and 'Documentation >'. The 'Settings' tab contains the following fields:

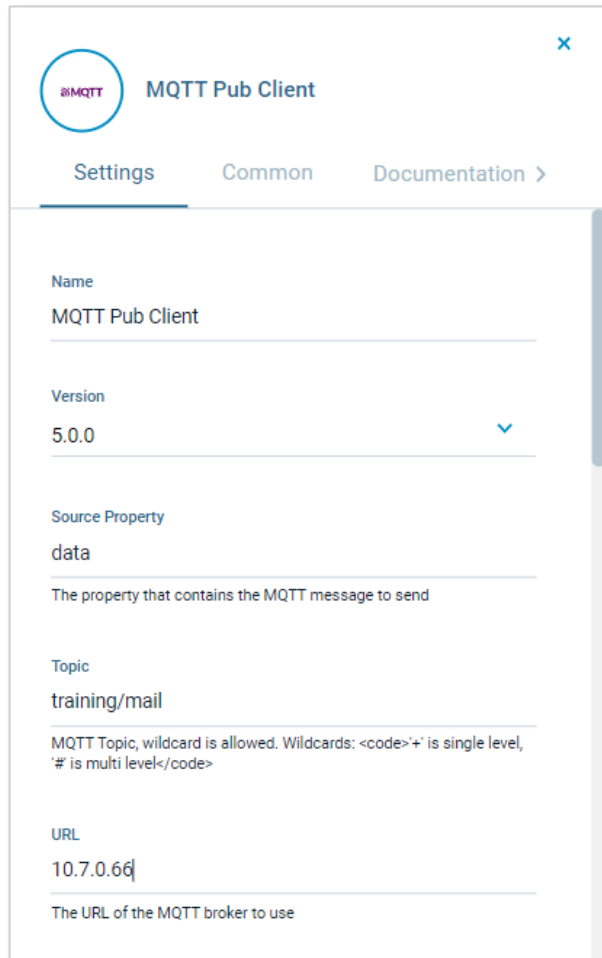
- Name:** Text Template
- Version:** 3.1.1 (with a dropdown arrow)
- Target Property:** data
- Template:** Pressure too high on {data.name} {(data.value)}

At the bottom of the window, there is a note: "The values from the incoming message is applied to this template and sets the resulting text to the message."

# Module

## MQTT Pub Client

- Publish message data to an external MQTT broker
- Default settings will convert message to JSON
  - Allows complete message structures to be transferred
  - XML and Raw formats also available
- Required settings:
  - Source Property: Which part of the message to send
  - Topic: Which topic to use when sending the data
  - URL: The IP address or hostname of the external broker



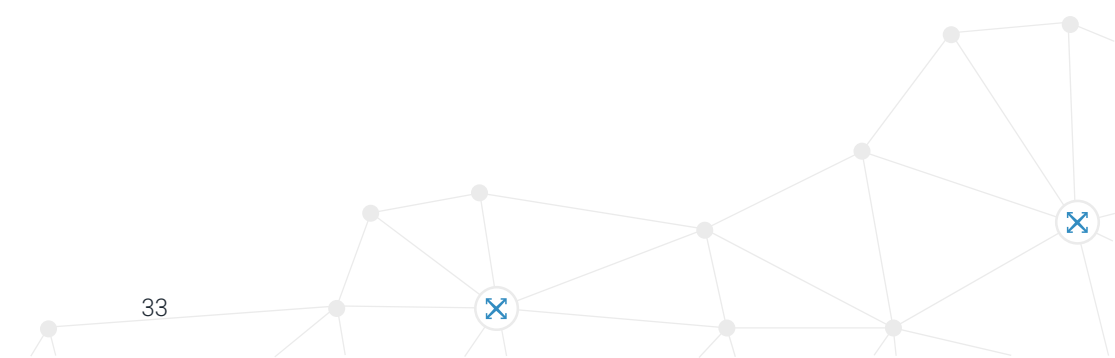
The screenshot shows the 'MQTT Pub Client' settings window. At the top, there is a logo with '@MQTT' and the title 'MQTT Pub Client'. Below the title are three tabs: 'Settings' (selected), 'Common', and 'Documentation >'. The main content area is divided into several sections, each with a label and a text input field:

- Name:** MQTT Pub Client
- Version:** 5.0.0 (with a dropdown arrow)
- Source Property:** data (with a description: 'The property that contains the MQTT message to send')
- Topic:** training/mail (with a description: 'MQTT Topic, wildcard is allowed. Wildcards: <code>+</code> is single level, # is multi level</code>')
- URL:** 10.7.0.66 (with a description: 'The URL of the MQTT broker to use')



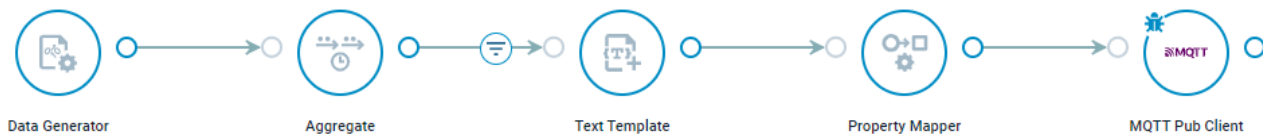


# EXERCISES



# Exercise 1

## Overview



- You will build a flow step by step, starting with an internal data source
- You will see how we can process this data (aggregate)
- Use message filters
- Finally, you should get an email from your flow

# Exercise 1A

## Your very first flow!



```
{
  "name": "machine",
  "data": {
    "temp": 12,
    "pressure": 489
  }
}
```

1. Login to Crosser Cloud (<https://cloud.crosser.io>)
2. On the Flows page, create a new flow called **Exercise 1**
3. Add a **Data Generator** module with default settings
  - Click on 'ADD EXAMPLE' to add the default template
4. Start an interactive debug session on the sandbox node
  - Click on 'Connect Node' and go to the 'Sandboxes' tab and click on 'Connect'
5. Start the flow, turn on debugging on the module and watch the output in the debug window
6. Change settings on the 'Data Generator':
  - Change *Number of Samples* to **3**
  - In the 'Data Rules' section at the bottom, set the behaviour on the *name* property to **Identifier**
7. Run the flow again and check the output

# Exercise 1B

## Aggregation



1. Add an [Aggregate](#) module
  - Value Property: `data.temp`
  - Target Property: `aggregate`
  - Interval: `10 seconds`
2. Run the flow and check the output from the 'Aggregate' module

# Exercise 1C

## Grouping



- 
1. In the 'Aggregate' module, add `name` as the *Source Property*
  2. Run the flow and check the output from the 'Aggregate' module
  3. What is the difference?

# Exercise 1D

## Creating a text message



1. Add a [Text Template](#) module after the 'Aggregate' module:
  - Target Property: `data.content`
  - Template: `The max value is {aggregate.max}`
2. Run the flow on your node and check the output

# Exercise 1E

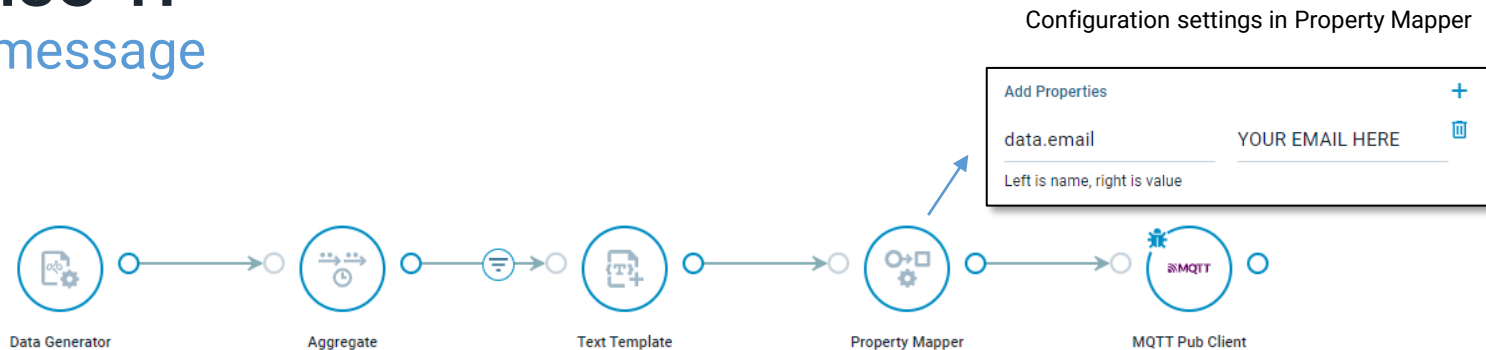
## Message filters



1. In the 'Text Template' module open the *Common* tab and specify a filter that selects messages where the *name* property is `machine-1`
2. Run the flow on your edge node and notify the difference

# Exercise 1F

## Send a message

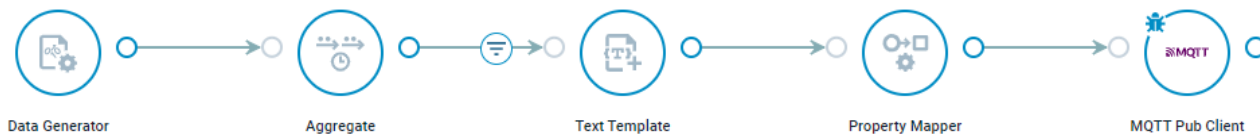


1. Add a [Property Mapper](#) module and add the settings shown above
  - The Property Mapper will be covered in the next session
2. Add a [MQTT Pub Client](#) module:
  - Topic: [training/mail](#)
  - URL: [10.0.48.117](#)
3. Run the flow and you should receive an email!
  - You cannot send more than one email per minute



# Exercise 1

## Wrap-up



### Things to test/consider:

- Try changing some settings in the Data Generator and notify the changes in the output
- Why did the number of output messages change when you added the 'Source Property' in the Aggregate module?
- Try some other filters in the Text Template module, like only letting through values in a range
- Try creating some other messages in the Text Template module, e.g. specifying the range of values (using min/max from the Aggregate module)



# SESSION – 02 END

How to build a new flow  
How to work with modules  
Testing flows in the FlowStudio