



## USE CASE:

Accomplish Zero Touch IT even further by allowing end users the ability to kick off a self service request through a bot in Teams instead of having to head to a service portal

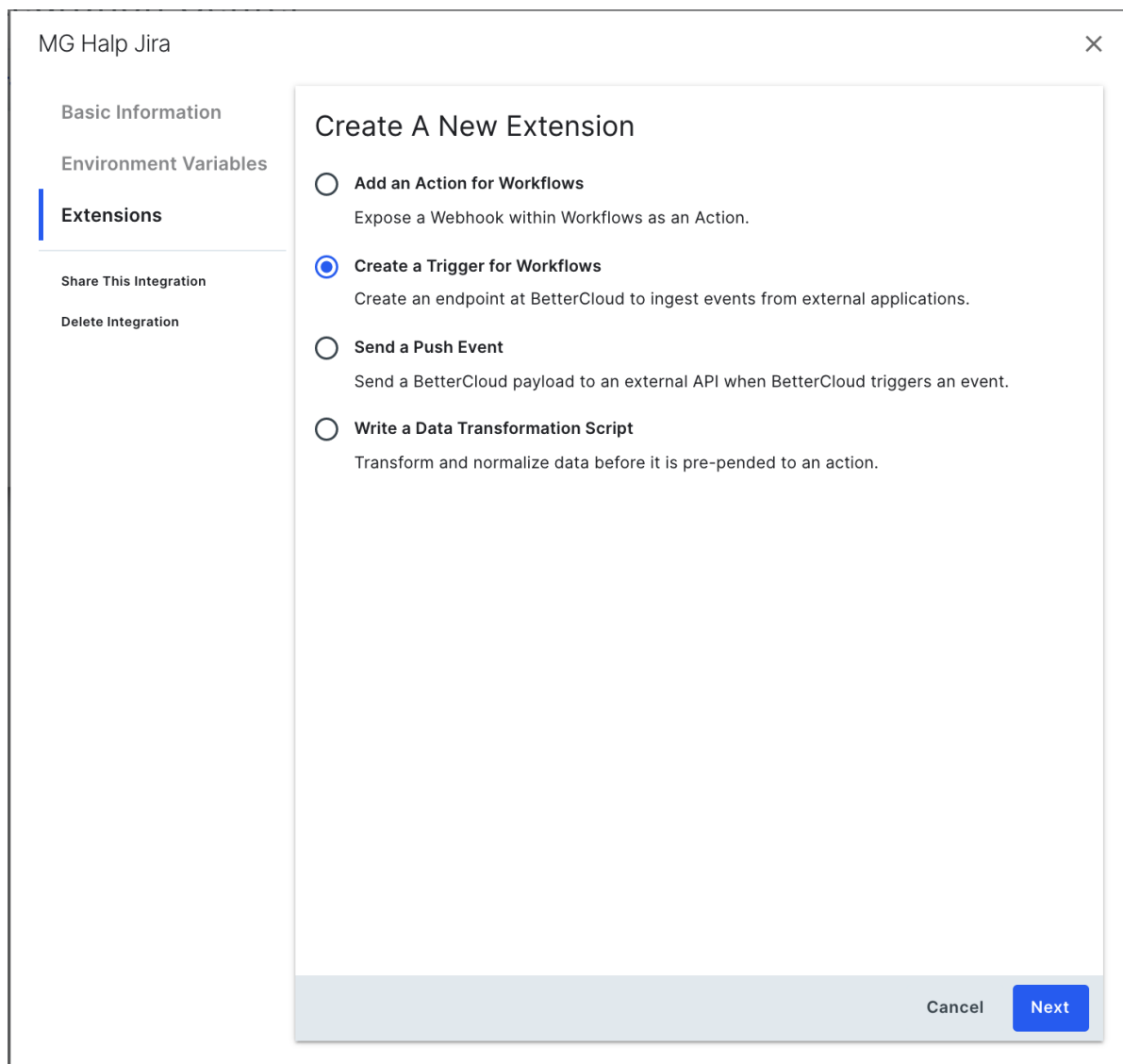
## PRE REQS:

- Jira Administrator Permission Level
- BetterCloud Platform API
- Jira Service Desk Integration by BetterCloud installed in your BetterCloud tenant with an administrative level familiarity with Jira Service Desk
- Basic Understanding of building Custom Integrations in BetterCloud
- Microsoft Teams Administrator Permission Level
- Halp Premium (If connecting to multiple Jira Projects)
- Microsoft Teams administrator email **MUST MATCH** Jira administrator email in order for sync to occur

## SETUP:

### BetterCloud

- Navigate to your Installed instance of Jira ServiceDesk in the BetterCloud Integration Center. From there, you can select “Modify” and navigate to “Extensions” to set up your API endpoint in BetterCloud for Jira to send its event data. Select “+ Add an Extension” to get started.
- From there, you will select, “Create a Trigger for Workflows”



- Once you have given the Trigger a name and optional description, you will be able to move onto creating the API endpoint.
- You can now copy the endpoint to your clipboard directly from the BetterCloud UI

MG Halp Jira

Basic Information

Environment Variables

**Extensions**

Share This Integration

Delete Integration

### Create a Trigger for Workflows

#### Initiate an inbound call to BetterCloud

You will need to send a request to the unique URL provided below.

Configure your request using the endpoint below.

`https://api.bettercloud.com/triggers/53a1adec-83e7-11ea-98d` [Copy to Clipboard](#)

[More about configuring inbound calls](#)

Waiting for your request

If you have sent a request and we have not seen it within a reasonable amount of time:

- Verify that you are using the correct unique URL.
- Make sure the request JSON is properly formatted.

You may leave this page if you wish, and we will keep listening in the background. You may return to the Extensions page (by clicking "Modify" on the integration) at any time to check the status.

If you need additional assistance, please contact our Support team.

[Back](#) [Next: Configure Workflow Options](#)

- The next part of your setup will take place in Jira ServiceDesk


## Jira Service Desk (Setting up request type)

- Go to Project > Project Settings > Request Type
- Create Request Type, Enter a name for and select "Submit a request or incident" type


## New request type ✕




Name \*

Icon





 [Change this icon](#)


Use workflow and fields from this issue type ⓘ


 Submit a request or incident ▼

- Task
-  Ask a question
-  Submit a request or incident
-  Emailed request

- Within your created request type head to the second tab of “Request Form”


Jira Service Management Your work Projects Filters Dashboards People Apps Create     

 Back to request types

 **Marisa App Request** View workflow

Agent view **Request form** Workflow statuses

Configure the request form customers will see. To make more fields available in this form, add them to the [Create Issue Screen](#) for issue type Preview form in portal

 **Submit a request or incident.**

**Request form description** (Optional) ⓘ

- Add the fields you want in your ticket (this will need to align with what you set up in Halp).

[← Back to request types](#)

## Marisa App Request

[View workflow](#)

[Agent view](#) [Request form](#) [Workflow statuses](#)

**Request form description** (Optional) ⓘ

**Request form help and instructions** (Optional) ⓘ

Links [[link name|http://example.com]]

### Visible fields

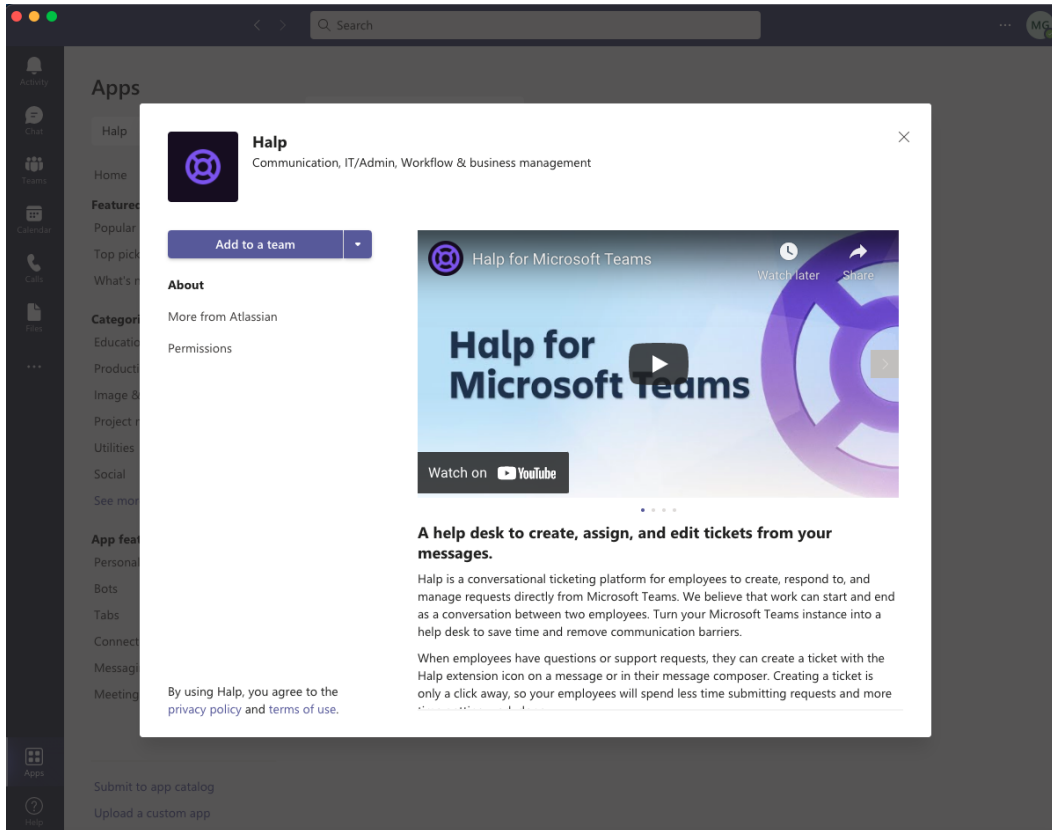
[Add a field](#)

Display name	Required	Field help (Optional)	Actions
Summary	Yes		<a href="#">Hide</a> <a href="#">Remove</a>
Your Email Address	No		<a href="#">Hide</a> <a href="#">Remove</a>
Application Name MG	No		<a href="#">Hide</a> <a href="#">Remove</a>

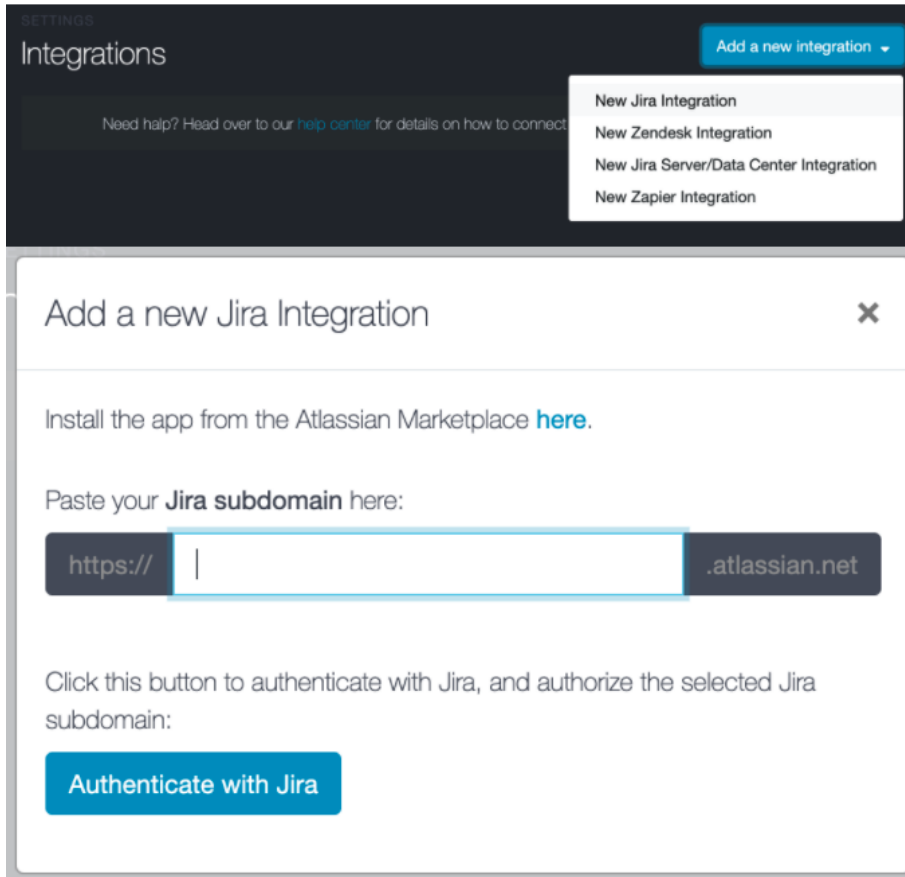
- Your Jira Request type should now be ready, next is to set up Halp

## Halp (Installing and setting up ticket form)

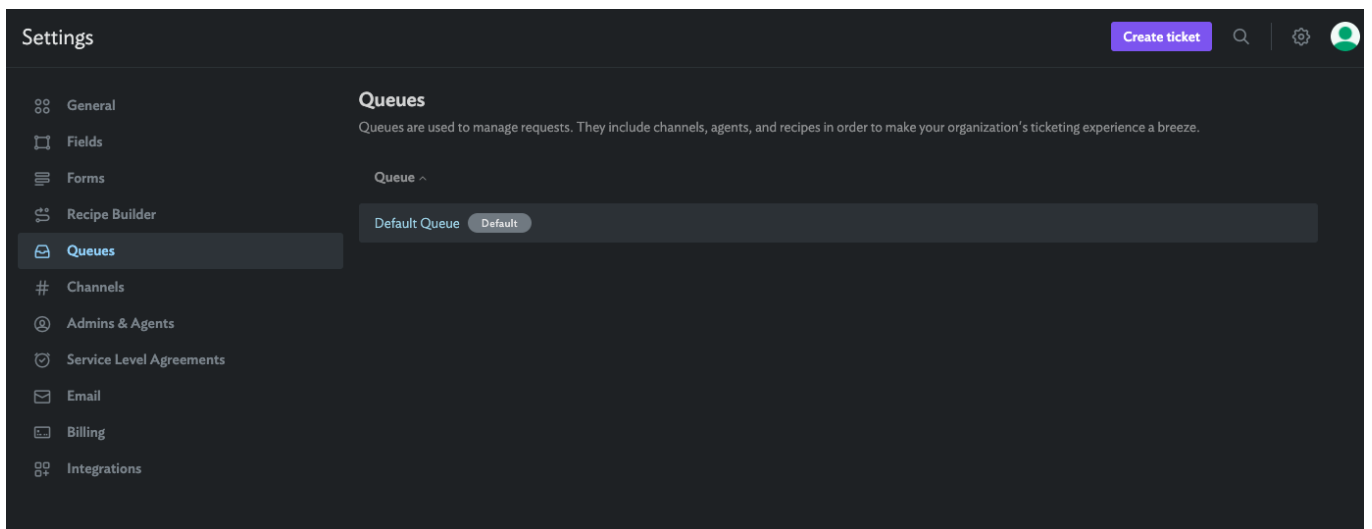
- Add Halp Bot to Teams - Go to <https://www.atlassian.com/software/halp> and click “Add to Teams”
- This will lead you to log into your Microsoft administrator account
- In bottom left hand corner of your teams desktop app click on apps> search for Halp
- Add Halp to a team (Most likely your IT team channel)



- This will create your triage team
- Head into your Halp web app ([your-workspace.halp.com](https://your-workspace.halp.com)) go to Settings > Integrations.
- Choose the option to Add a new Jira integration. Follow the instructions to OAuth with Jira. This enables us to verify your identity to tie the Atlassian app to your Halp account.



- Once your project is connected, head to “Queues” on the left hand side









- Select the queue you would like to create the ticket form under
- Click “Fields”

- Create the fields you want to have in your form (these fields need to align with the fields in your request type in Jira)

**Field Builder** Create

In Halp, fields are the groundwork for forms. Fields can also be used with recipes to move tickets between queues, channels, etc.

Field ^	Type	Options	Enabled	
App Name 1	Dropdown	A... Al... SAL... ZO...	<input checked="" type="checkbox"/>	 
Application Name	Dropdown	ZOOM ... GOOGLE ENTERPRI...	<input checked="" type="checkbox"/>	 
Requestor Email 1	Long Text	NONE	<input checked="" type="checkbox"/>	 

- Once you have your fields head to “Forms”
- Create a form (this is what is going to be asked of the user when submitting a ticket through Halp)



### Edit Form ✕

Name  
App Request 1

Queue  
Default Queue ▼

Fields  
Add field to form ▼

Form preview

Requestor Email 1

⋮   Required 🗑️

Cancel Save

- Go back to the integrations page
- Click “Configure”
- Select the Jira project and the type of issue or request you built in Jira and click “Next”

Connect Request Type for: ✕  
https://myotherurl.atlassian.net

---

Project:

Request type

- Map fields that are carried over from Halp to Jira (these will auto populate from what is available on both sides and should line up).

Configure myotherurl ✕  
Marisa App Request • https://myotherurl.atlassian.net

---

<input type="text" value="Title"/>	should sync with	<input type="text" value="Summary"/>
<input type="text" value="Requestor Email 1"/>	should sync with	<input type="text" value="Your Email Address"/> <input type="button" value="✕"/>
<input type="text" value="App Name 1"/>	should sync with	<input type="text" value="Application Name MG"/> <input type="button" value="✕"/>

[Learn more about Synced & Computed fields](#)

---

[Learn more about Status & Workflow syncing](#)

- Click “Edit Recipe”
- Choose your routing recipe to

- a) **Trigger:** When a ticket is created with the form
- b) **AND:** Select the form you created in above steps
- c) **THEN:** Sync the ticket
- d) **CHOOSE:** Your Jira Project

Dialog box titled "Edit a routing recipe" with a close button (X) in the top right corner.

**Trigger**

- When a ticket is created with the form
- App Request 1

**Then...**

**Action**

- Sync the ticket (including comments, statuses, and fields) with an integration
- Jira (https://myotherurl.atlassian.net)

**Project**

- myotherurl (Marisa App Request)

Buttons: Cancel, Save

- Your Halp form should now be ready to go

## Jira Service Desk (Setting up automation/webhook to BetterCloud)

- Within your Jira console navigate to your project> project settings >automation
- Create Rule with below settings
  - When:** Issue is created
  - Creator:** Halp
  - Request type:** \*Name of your request type in Jira
  - Then:** Send Web request

# Automation

DRAFT

[Return to list](#)



## Marisa App Request

Rule details

Audit log

**When: Issue created**  
Rule is run when an issue is created.

**Creator equals**  
Halp

**Request Type equals**  
Marisa App Request

**Then: Send web request**

## Send web request

This action will send a HTTP request to the url specified below:

Web request URL \*

Request parameters must be url encoded, smart values should use: {{value.urlEncode}}.

Headers (optional)

Name (e.g. Content-Type)	Value (e.g. application/json)

Add

HTTP method \*

POST

Web request body \*

Issue data

Wait for response

Delay execution of subsequent rule actions until we've received a response for this web request

## Connecting Jira Service Desk to BetterCloud

- Paste the endpoint copied from bettercloud into the Web request URL in Jira

The screenshot shows the Jira Workflow Builder interface. On the left, a rule is configured with the following steps:

- When: Issue created** (Rule is run when an issue is created.)
- Creator equals Halp**
- Request Type equals Marisa App Request**
- Then: Send web request** (POST) with the following URL:  
`https://api.bettercloud.com/triggers/53a1adec-83e7-11ea-98a2-2172b1d2ee47/ca135f1f-3660-11ec-85f6-25025b622e4c/c71f8b20-7b07-4dc8-9e9d-cc6fc8e4f078?authorization=H50yNyl47qW1tcU7olef`

The main configuration area for the 'Send web request' action is shown on the right:

- Web request URL:** `https://api.bettercloud.com/triggers/53a1adec-83e7-11ea-98a2-2172b1d2ee47/ca135f1f-3660-11ec-85f6-`
- Request parameters:** Must be url encoded, smart values should use: `{{value.urlEncode}}`.
- Headers (optional):** Fields for Name (e.g. Content-Type) and Value (e.g. application/json).
- HTTP method:** POST
- Web request body:** Issue data
- Wait for response:**  Delay execution of subsequent rule actions until we've received a response for this web request
- Buttons:** Cancel, Save
- Link:** > Validate your web request configuration
- Footer:** > How do I access web request response values in subsequent rule actions?

- For the “Web request body”, you can either choose to send all of the “Issue data” or “Custom Data” which allows you to limit the amount of data sent to BetterCloud.
- You can add other customizations you need into your automation, or just save it as is. The rest of the setup will take place in BetterCloud

## Setting up Custom Trigger on the BetterCloud

- Return to the configuration page in BetterCloud for your Custom Trigger.
- If BetterCloud has not received an inbound trigger from Jira, you will see the page will state that it is, “Still waiting for your request”. If this is the case, you will need to ensure your automation is set up correctly so that it will send the webhook to BetterCloud when the conditions have been met.
- If you have successfully triggered the endpoint, you will be taken to a page that will allow you to determine which parameters that have been passed to BetterCloud should be available in the Workflow Builder.

MG Jira Halp ×

Basic Information

Environment Variables

**Extensions**

---

Share This Integration

Delete Integration

## Create a Trigger for Workflows

### Configure Workflow Options

Based on the request we received, you may use the following parameters for either Conditions or Dynamic Fields in your workflow.

**Endpoint URL:** `https://api.bettercloud.com/triggers/2e190efc-6dd0-11e9-a424-8d4b08091a95/8926d6cf-37f2-11ec-8204-11c717be3245/9f650dc3-ae1c-425a-ae2f-4fad947156ed?authorization=LsjxO2nllmhULGRSBZGVQOWT09UAWbVL`

View Raw Data Response ▶

Select the parameter that will identify your workflow trigger [Learn more](#)

fields.issue.type.name ▼

Select which parameters you want to use for either Conditions or Dynamic Fields in your workflow.

Parameter	If Condition	Dynamic Field
<a href="#">self</a>	<input type="checkbox"/>	<input type="checkbox"/>

Back
Save & Publish

- From here you can decide which parameters should be available as IF statements and which should be available as variables when performing actions within a BetterCloud workflow.
- You can now “Save & Publish” your new Custom Trigger.
- From here, you are ready to build a new workflow that will trigger from a Service Request being raised in a Halp chat bot in Teams, through Jira to BetterCloud.