

The logo for BetterCloud, featuring a stylized white cloud icon above the text "BetterCloud" in a white sans-serif font.

+

The logo for Zendesk, featuring a stylized white "Z" icon above the text "zendesk" in a white sans-serif font.

## Use Case:

We are going to show you how you can configure Zendesk and BetterCloud to automate application self service requests for end users. The goal is to help you reduce the number of tickets that need to be manually addressed when an end user requests access to a business application.

## Pre-Reqs:

- BetterCloud Platform with the Platform API add-on
- Zendesk for service (all SKUs)

## Setup:

### BetterCloud Configuration:

To get started, Navigate to BetterCloud's Integration Center and select "Create a Custom Integration" to start from scratch.

The screenshot shows the BetterCloud Integration Center interface. At the top right, there is a search bar with the text "Search Your BetterCloud". Below this, the main content area is divided into two columns. The left column features a large grey placeholder bar and two integration cards. The first card is for "Azure Automate" with a gear and lightning bolt icon, labeled "BetterCloud Built" and "Actions". The second card is for "BlueJeans" with a blue "B" icon, also labeled "BetterCloud Built" and "Actions". The right column contains a heading "Custom Integrations Are Here." followed by a paragraph: "The BetterCloud API enables you to trigger custom WebHooks, Push Events, and more to your applications." Below this is a prominent blue button labeled "Create A Custom Integration" and a link "More about BetterCloud API". At the bottom of the right column, there is a section titled "USE CASES" with a list of items: "Application Insights", "Spend Reporting", "User Lifecycle Management", "Application Configuration", and "Least Privilege Access".

Fill out the name of your Integration, and select a logo. You do not need to provide authentication for this integration, just select "None." Click "Save" to finish setting up your Custom Integration.

✕

Custom Integrations allow you to manage multiple Webhooks, Push Events, and Translation Scripts as a single Integration.

Below, provide a display name and provider used (you may select Other if you cannot find a specific provider), then provide the authentication details for this Integration.

### Integration Details

Name \* (Maximum 30 Characters)

Logo \*

✕ ▼

### Authentication

Use API Token     Use Basic HTTP Authentication     None

Limit Execution Rate To  Request(s) Per Second

### Additional Help Resources

[Rate Limiting Help](#)

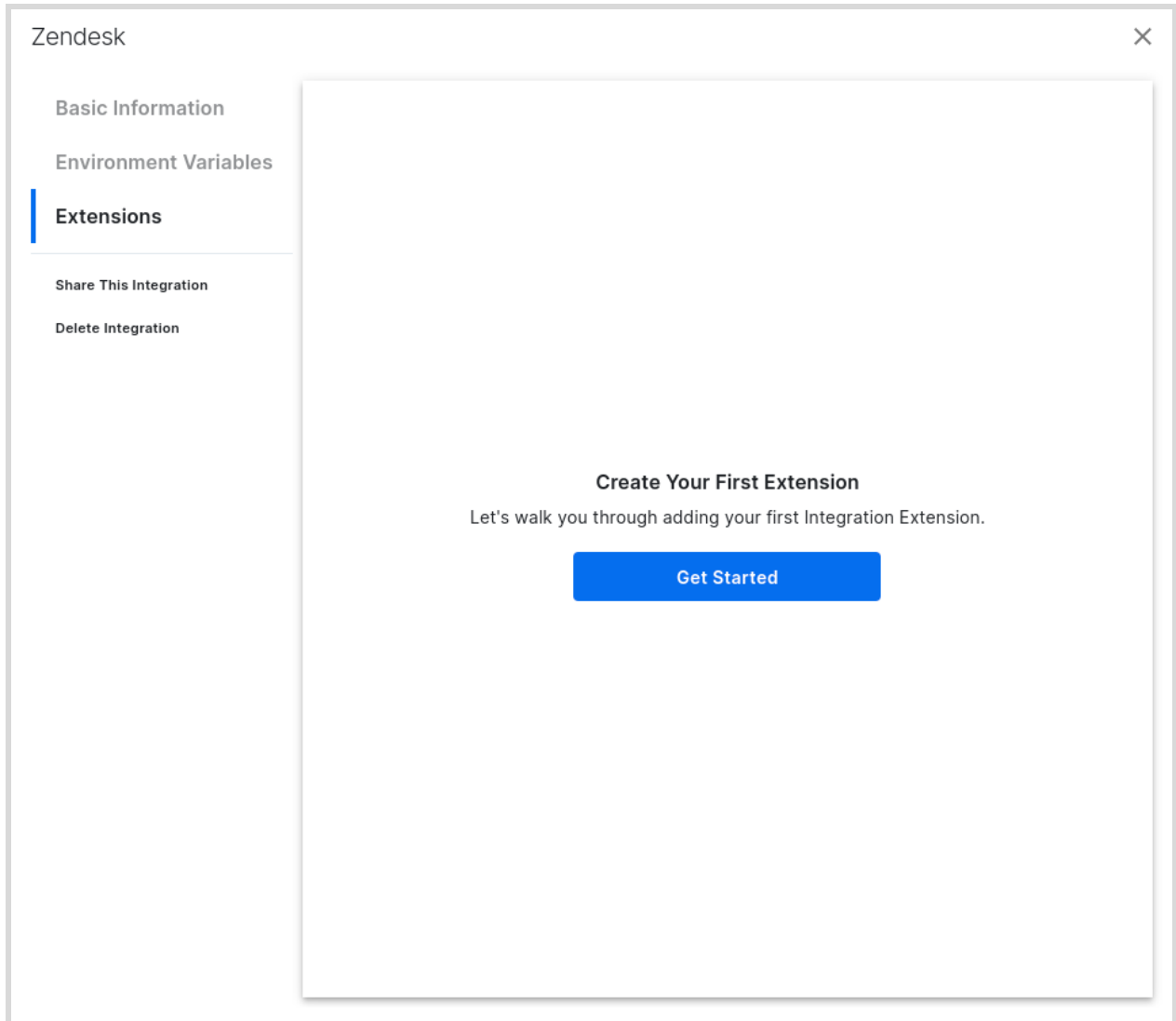
Cancel Save

**Note:**

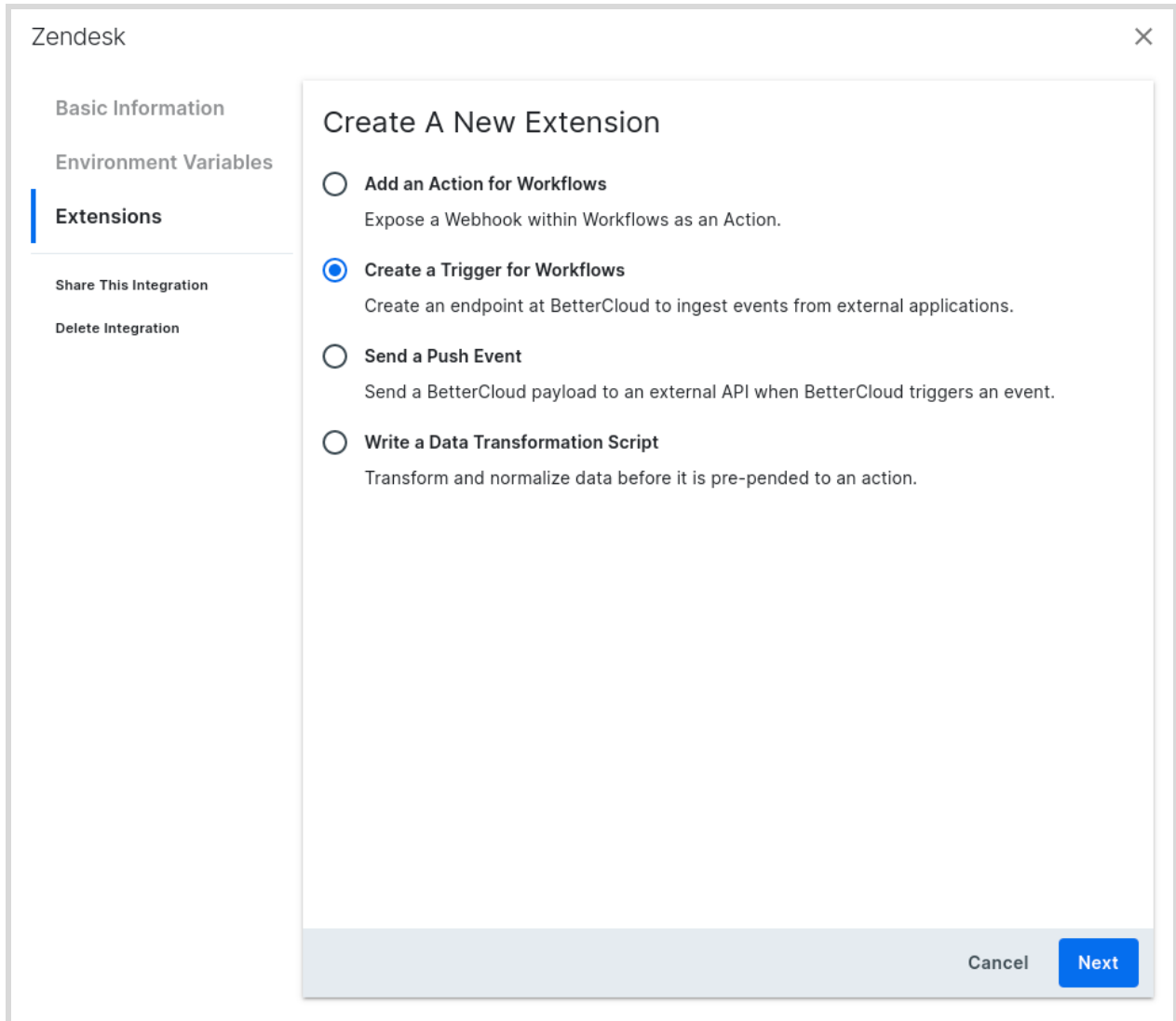
Choosing an Authentication method is optional at this step, as an authentication token is built into our endpoint URLs when they are generated. However, if you also want to use this Custom Integration for Custom Actions, authentication may be required depending on the platform you are working with and your use case.

If you don't see a logo you want to use, select the "Other" logo. To add a custom logo, you may reach out to our Support team with a .svg file of the logo you want to use. You cannot change your Integration's logo once you click "Save".

In your integration, navigate to the "Extensions" menu and click "Get Started"



Select "Create a Trigger for Workflows" and click "Next"



Name your trigger and give it a description, then click "Next: Inbound Request" to generate your Inbound Request URL.

Basic Information

Environment Variables

**Extensions**

Share This Integration

Delete Integration

## Create a Trigger for Workflows

### About Custom Triggers

Custom triggers allow you to send information from another SaaS application to BetterCloud via an inbound request. Once received, you can use the request data as a When condition within a Workflow, and as a Dynamic Field within a Workflow Action.

To complete a custom trigger, you will be required to have proper access within the external SaaS application to set up an API request. Contact your external SaaS provider for more information.

Trigger Name 30 Characters Max

This name will display in the workflow library when selecting it as your When condition.

Description Optional

Provide a description for what your trigger will be.

[Back](#)

[Next: Inbound Request](#)

Zendesk ×

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.

[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](#)

**Note:**

The name you input here is what displays in the "When" section of Workflow builder

## Zendesk Configuration


### Create New Input Field/Form

1. Navigate to "Objects and rules" and then "fields"
2. Create a New Field and select either Text or Dropdown
3. Make the field editable to end users and name it something like "App Name"
4. Make the field required in order to solve for the ticket

5. If you selected the Dropdown option, ensure you populate the field values with the different request options (ie. application names) and save.

### < App Name

**Active** Custom field ID 1900001894107. Created by Michael Praino on 2021/09/30. Last updated on 2021/10/26.

**Text** [Learn more](#) 

**Description**

Enter an optional description

---

**Permissions**

**Agent only**  
Field can only be edited by an agent.

**Editable for end users**

**Read-only for end users**

**Title shown to agents**

App Name

**Required to solve a ticket**

**Title shown to end users**

App Name

**Required to submit a request**

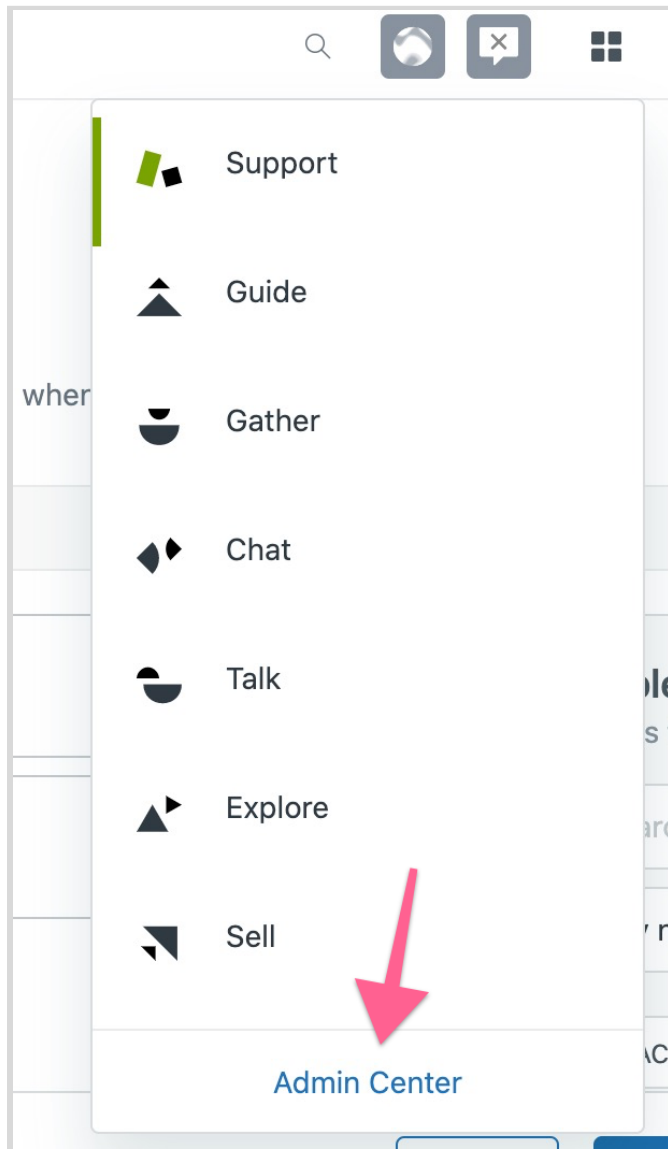
**Description shown to end users**

Enter an optional description



## Create Webhook and Custom Trigger

1. Navigate to the Admin Center in Zendesk



2. Click 'Integration settings'
3. Click Webhooks > Actions > Create Webhooks
4. Enter name and description
5. Navigate to BetterCloud and create a new Custom Integration
6. Name: Zendesk  
Logo: Other  
Authentication: None

7. Create an extension > Create Trigger for Workflows
8. Trigger Name: New Zendesk Ticket Submission
9. Click Next
10. Copy the URL and paste back in Zendesk under “Endpoint URL”
11. Request Method: POST  
Request Format: JSON  
Authentication: None
12. Click ‘Test Webhook’ in Zendesk
13. Test event source: Custom test
14. Request JSON Body  

```
{"email": "user@domain.com","appName": "app"}
```
15. Confirm BetterCloud successfully received the event

Test webhook

Test event source ⓘ

Custom test

Send test

Request JSON Body

JSON

```
1 {
2   "email": "user@domain.com",
3   "appName": "appName"
4 }
```

Response body      Response headers

**200 OK**

TEXT

```
1 Successfully received event|
```

16. BetterCloud should also display a message “Request Received!”. Once confirmed, click Next.


## 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/050e62bb-aa91-11ea-8d`

 Copy to Clipboard

[More about configuring inbound calls](#)

 Request received! Click Next to configure workflow options.

**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.

17. Select the following

## 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/8307fd8f-c55a-4b5c-a8ec-df63e66013b0/72e33992-2096-11ec-b31d-cfc68ec5de6b/8731162d-92ee-4ac3-af6e-34b717738db9?authorization=SkXSvWk14V0HsYIBuYF2CvMk9mTd0WZc`

[View Raw Data Response](#)

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

Email

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

Parameter	If Condition	Dynamic Field
App Name	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Email	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Raw JSON Response		<input type="checkbox"/>

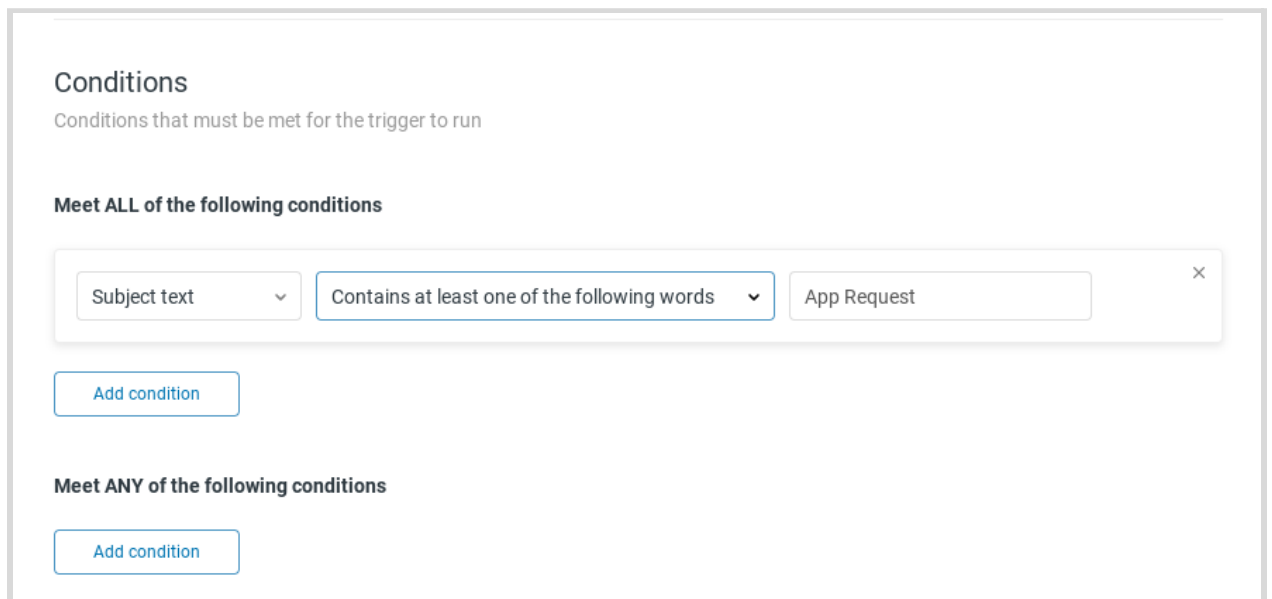
[Back](#) [Save & Publish](#)

18. Save and publish

## Create Zendesk Trigger

Here, you'll create a trigger so whenever a user submits an app request, it'll send the ticket field values to BetterCloud

1. Navigate to Settings > Triggers and click Add trigger
2. Under "Contains at least one of the following words", add 'Subject text' and then "App Request"
3. Add 'Status Is New' to trigger this when a new ticket is created



The screenshot shows the 'Conditions' section of a Zendesk trigger configuration. It is titled 'Conditions' and has a subtitle 'Conditions that must be met for the trigger to run'. Below this, there are two options for how conditions are met: 'Meet ALL of the following conditions' and 'Meet ANY of the following conditions'. The 'Meet ALL' option is currently selected and active. Under this option, a single condition is configured: 'Subject text' (selected from a dropdown) 'Contains at least one of the following words' (selected from a dropdown) 'App Request' (entered in a text field). There is a close button (X) on the right of this condition. Below the 'Meet ALL' section is an 'Add condition' button. The 'Meet ANY' section also has an 'Add condition' button.

4. Under Actions, choose 'Notify active webhook' and choose the webhook you created
5. You'll need to enter in a JSON body with the form values. In this example we are sending over the requester's email and the app requested. For native ticket fields, you can select 'View available placeholders'.

## Actions

Actions that will occur if global conditions are satisfied

Notify active webhook ▾ App Request ▾

**Endpoint**  
https://api.bettercloud.com/triggers/8307fd8f-c55a-4b5c-a8ec-df63e66013b0/72e33992-2096-11ec-b31d-cfc68ec5de6b/8731162d-92ee-4ac3-af6e-34b717738db9?authorization=SkXSvWk14V0HsYIBuYF2CvMk9mTd0WZc

**Description**  
-

**Method**  
POST

**JSON body**

```
1 {
2   "appName": "{{ticket.ticket_field_1900001894107}}",
3   "email": "{{ticket.requester.email}}"
4 }
```

[View available placeholders](#)

6. For custom fields, you will need to follow the format `{{ticket.ticket_field_xxxxxxx}}`. Replace 'xxxxxxx' with the field id found under Settings > Ticket Fields

## Ticket Fields

Ticket fields contain all ticket data, such as subject, description, and priority. Add and manage ticket fields to customize your ticket forms for agents and end users. [Learn more](#)

Active Inactive

Title	Field ID	Type	Date modified	
Assignee <small>System field</small>	47229547	Drop-down	2021-10-26	
Group <small>System field</small>	47229527	Drop-down	2021-10-26	
Priority <small>System field</small>	47229507	Drop-down	2021-10-26	
Type <small>System field</small>	47229487	Drop-down	2021-10-26	
Status <small>System field</small>	47229467	Drop-down	2016-11-21	
Description <small>System field</small>	47229447	Multi-line	2016-11-21	
Subject <small>System field</small>	47229427	Text	2016-11-21	
App Name	1900001894107	Text	2021-10-26	
Request Type	360048654532	Drop-down	2021-10-26	

### 7. Save