



USE CASE:

Leveraging BetterCloud to fulfill and automate Self Service requests from Slack using Zapier

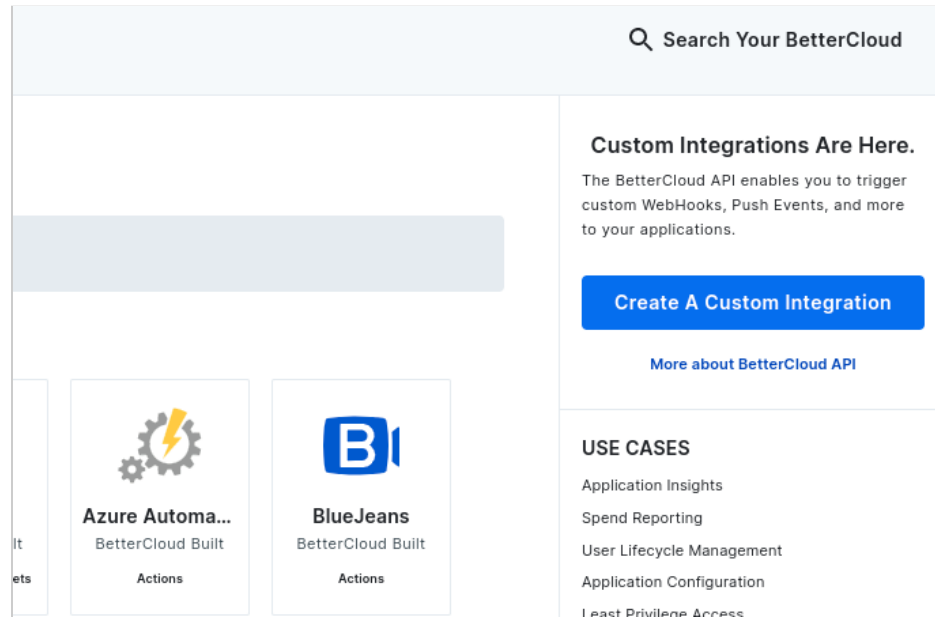
PRE REQS:

- BetterCloud Platform with the Platform API add-on
- Slack (all paid SKUs) and familiarity with Slack Workflows
- Zapier (all paid SKUs)

SETUP:

BetterCloud

- To get started, navigate to BetterCloud's Integration Center and select "Create a Custom Integration" to begin building a new Custom Integration.



- 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)

Slack Custom Trigger

Logo *

Other

Authentication

Use API Token Use Basic HTTP Authentication None

Limit Execution Rate To Request(s) Per Second

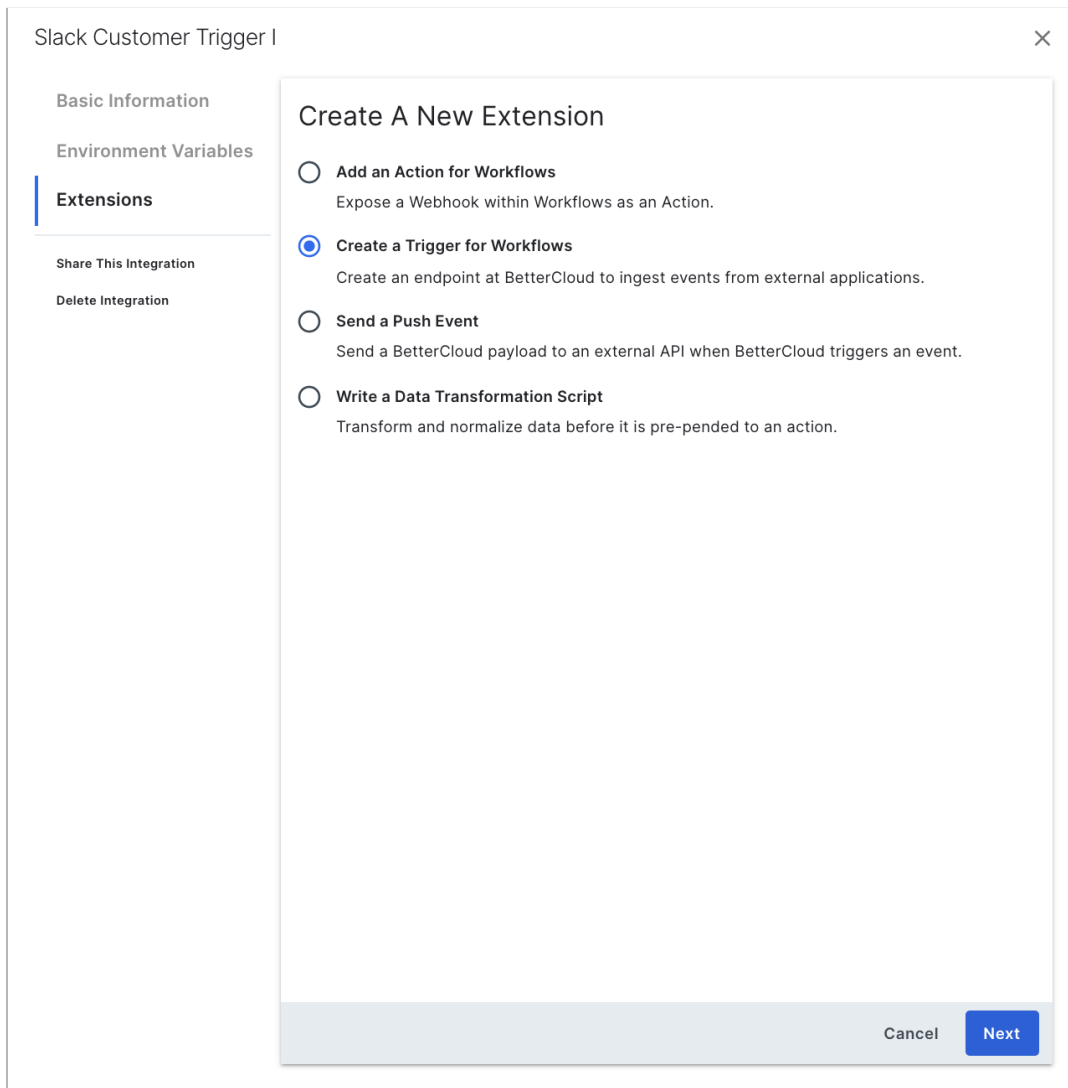
Additional Help Resources

[Rate Limiting Help](#)

Cancel Save

Note:

- 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".
- From here, navigate to the "Extensions" menu and click "Get Started."
- Select "Create a Trigger for Workflows" and click "Next"



- After naming your Trigger and providing a brief description, move to the next step and copy the endpoint URL to your clipboard directly from the BetterCloud UI.

Note:

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

Slack Customer Trigger I


- 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/8307fd8f-c55a-4b5c-a8e`  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.

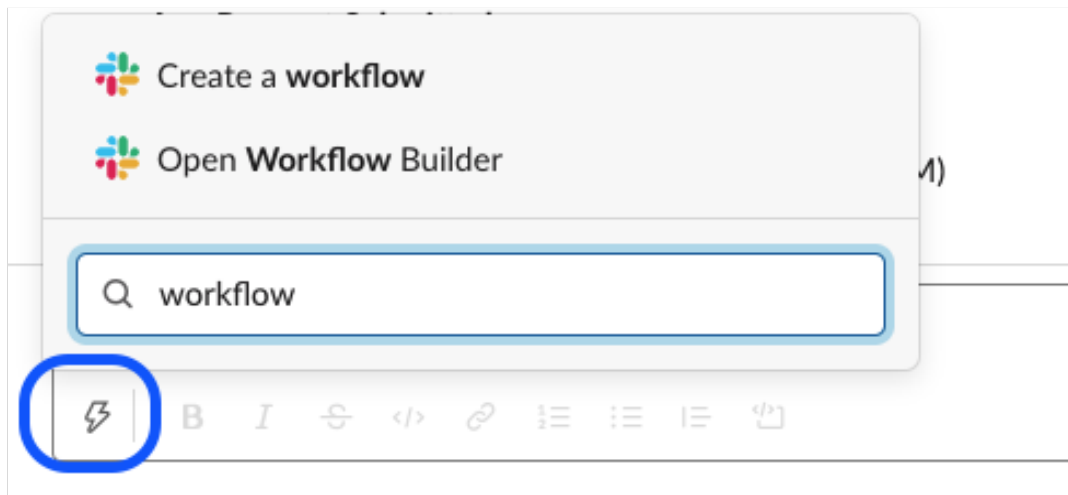
You may leave this page if you wish, and we will keep listening in the background. You may return to the Extensions page (by clicking "Modify" on the integration) at any time to check the status.

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

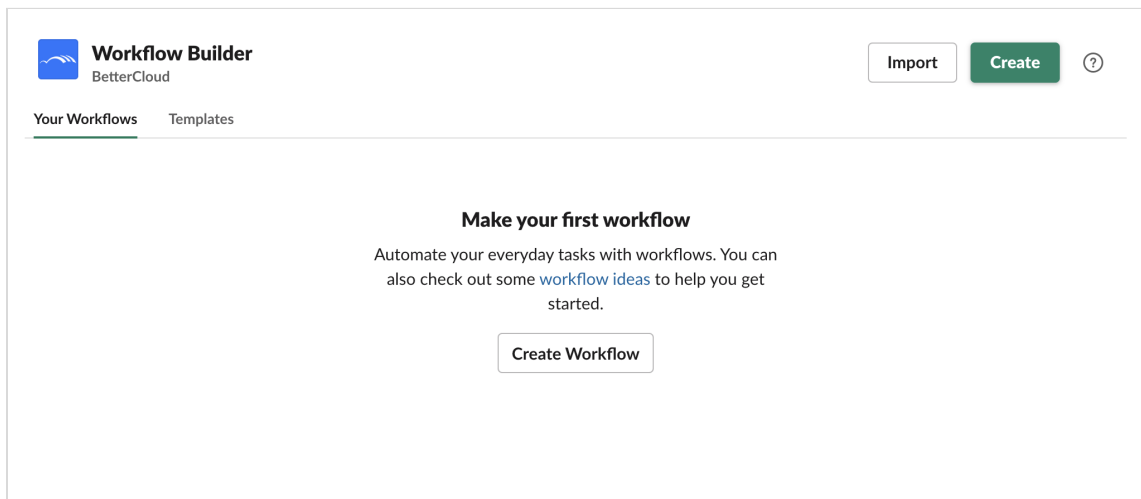
[Back](#) [Next: Configure Workflow Options](#)

Slack


- In your Slack workspace, create a new Public Channel named “app-request” and a new Private Channel named “internal-app-request.”
 - Public channel will allow users to group and submit the appropriate application requests
 - Private channel will consolidate and route all incoming app requests to Zapier and subsequently, BetterCloud
- Now, we want to build out the workflow within Slack to route the requests.
- From your newly created public channel, navigate to the lightning icon at the bottom of a message box, and type in Workflow. This will return a suggestion for “Open Workflow Builder.”



- Select on the option to be redirected to Slack’s Workflow Builder Page. Begin building a new Slack Workflow by clicking on the “Create” button.



- Name your workflow and choose a way to start this workflow
 - In our example, we want to trigger workflow from the “Shortcut,” meaning the workflow will kick off when someone selects it from the “Shortcut” tool.
- Next, determine which channel you want the “Shortcut” to be available from. This is where the workflow will begin
 - In this example, select your newly created Public Channel and provide a short name

 **Shortcut**
✕

This workflow starts when someone selects it from the available shortcuts in a channel.

Which channel?


🔒 apprequest
▾

Add a short name

5

This name appears in the menu so try briefly describing what someone's about to do.

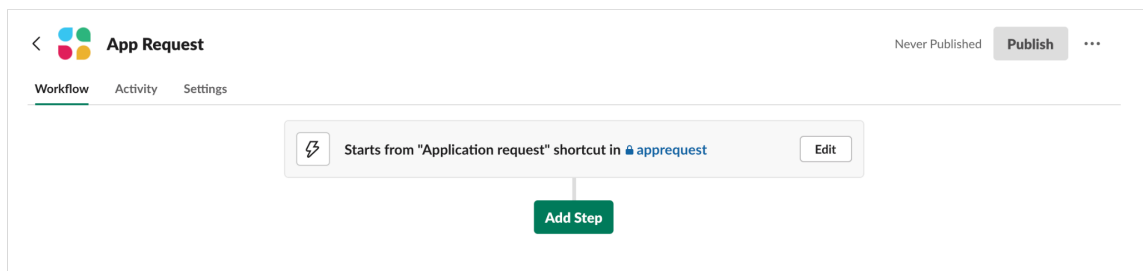
Example


B
I
☒
</>
☰
☰
☰
📄

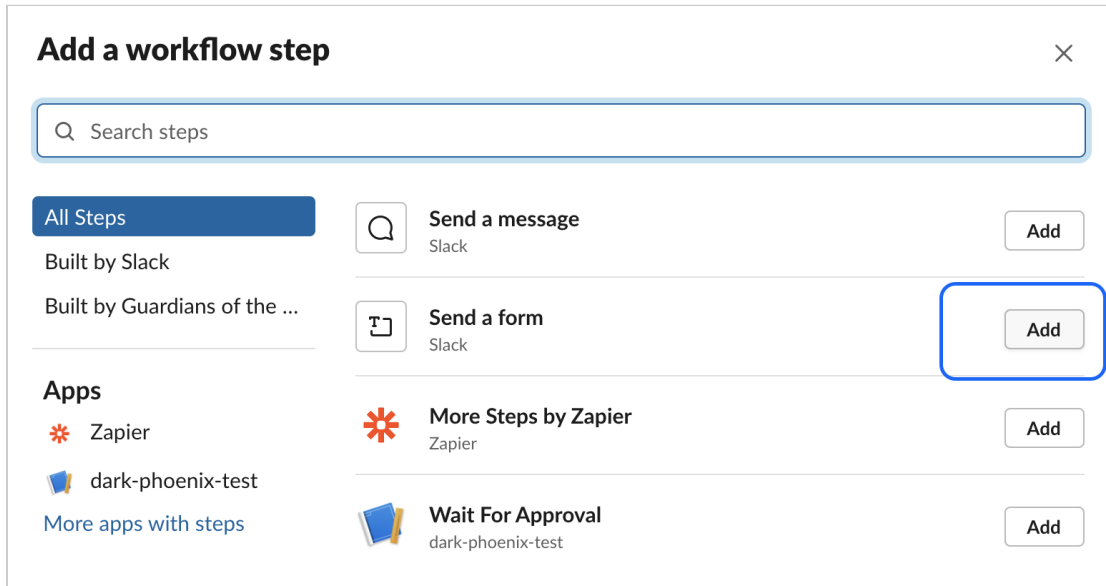
Cancel

Next

- Click “Next” to begin configuring the actual workflow.
- As you can see, the first step in the workflow is the “Starts from ...” trigger.
- To simplify the request process, we will build out a simple form. To do so, click “Add Step.”



- Step 2 - One of the workflow options available through Slack is “Send a form,” we will use this step to configure the Application Request Form.



- Give the form a “Title” and define the question(s) required.
 - For this example, we want to know what application the user is requesting access to
 - To further simplify the process, we can make available a dropdown of available application options by “Select from a list”
 - Leave all else as is and “Save”

- Step 3 - We want to route the submitted request to our newly created Private channel. Using Slack’s “Send a message” option, we will configure the message to send to the Private channel.
 - Select the Private channel to send the message to
 - In the Message text: there are some notes to be careful of
 - The fields defined here will actually be used as data within the payload that will be sent to BetterCloud
 - No spaces in any data point that you are planning to send, so Requestor Email should be formatted as RequestorEmail. (You can change the display name at a later time)
 - For any variable fields, such as the Requestor Email Address or the Application Requested, we can dynamically pull the data using the “Insert a variable” option.

- We can wrap these variable fields in () parentheses directly after the data field name, again no spaces
- In the Email field, click on the hyperlinked variable dropdown to select “Email” as the display option
- Preview the message to make sure everything is in good order
- Use the following image as reference

🗨️ **Send a message**
✕

Send this message to:

🔒 internal-app-request-approval
▼

Message text

App Request Submitted

RequestorEmail([Person who submitted form](#) | Email ▼)

AppRequested([Response to: "What application do you need access to?"](#) ▼)

DateRequested([Time when workflow started](#) ▼)

✎ Insert a variable

Include a button
 You can add a button to this message. Clicking it will move people to the next step of the workflow.

Preview

A
P
App Request WORKFLOW

A
P
App Request Submitted

RequestorEmail(Variable text)

AppRequested(Variable text)


DateRequested(Variable text)

Remove


Cancel

Save

- Following steps are optional to enhance auditability and end-user experience within Slack
- Step 4 - Send confirmation message to Public channel, using below as reference









 **Send a message** ✕

Send this message to:

 apprequest ▼

Message text



Person who submitted form ▼ submitted an App Request!

B *I*        

[🔗 Insert a variable](#)


Include a button
You can add a button to this message. Clicking it will move people to the next step of the workflow.

Preview


 **App Request** WORKFLOW
 @Variable User submitted an App Request!

Remove Cancel Save

- Step 5 - Send confirmation to person who submitted the form.


 **Send a message** ✕


Send this message to:

 Person who submitted form ▼

Message text

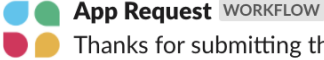
Thanks for submitting the request. We're on it!

 😊

 [Insert a variable](#)

Include a button
You can add a button to this message. Clicking it will move people to the next step of the workflow.

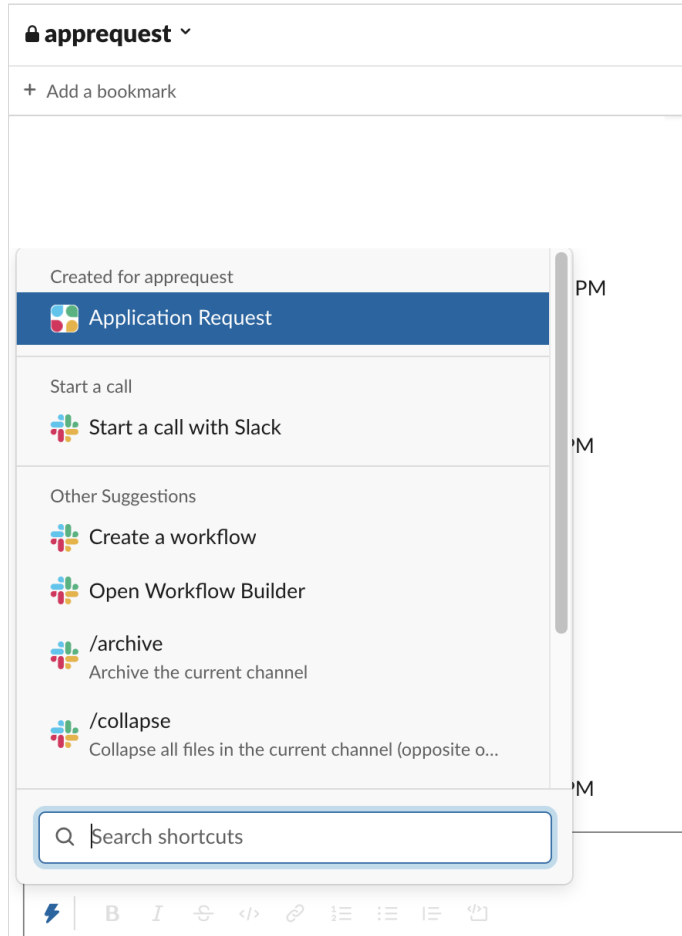
Preview

 **App Request** WORKFLOW
Thanks for submitting the request. We're on it!

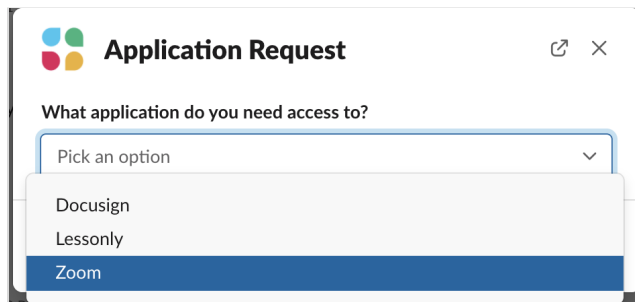
Remove Cancel Save

- Publish the Workflow
- Test the workflow
 - Navigate back to Slack instance
 - Enter the #apprequest Channel and initiate the Shortcut

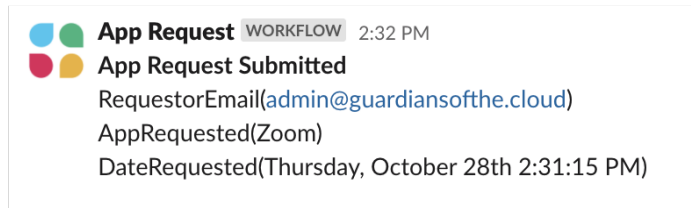
- First option available will be “Application Request”



- Selecting on this option will open the App Request Form



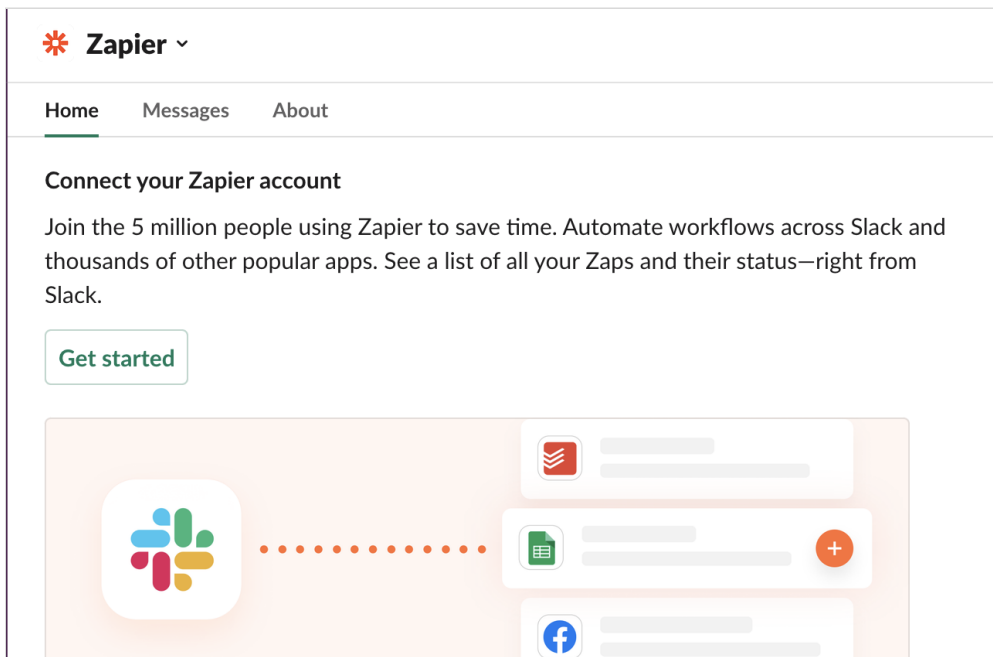
- Select an app from the dropdown and submit the form
- Confirm auto-response in the private #internalapprequest channel for the Ticket Submitted



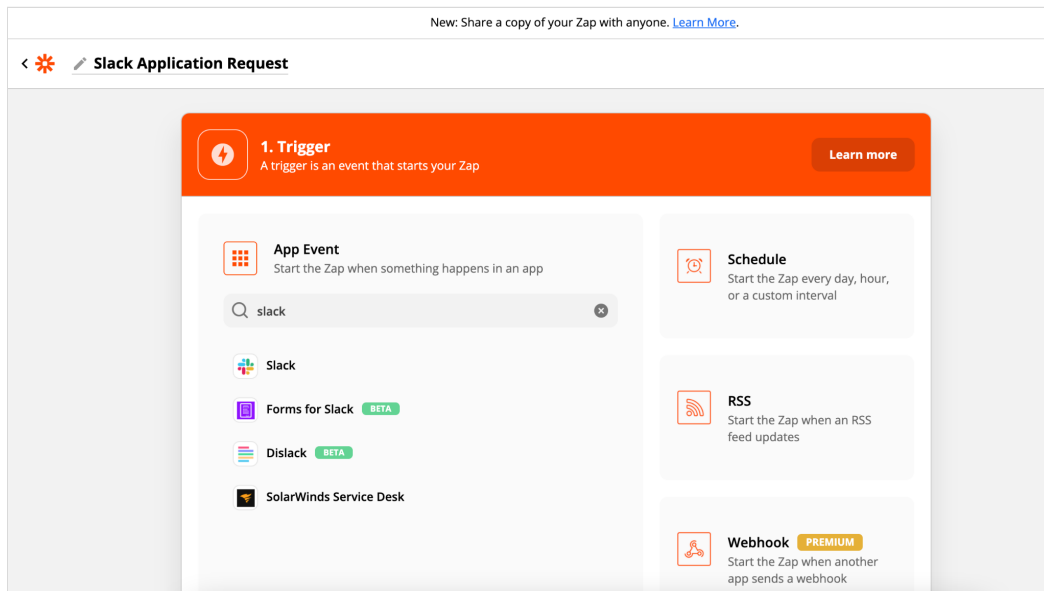
- Optional:
 - Confirm auto-response in the #apprequest channel for App Request Submitted
 - Confirm auto-response from Slackbot confirm App Request Submitted

Zapier

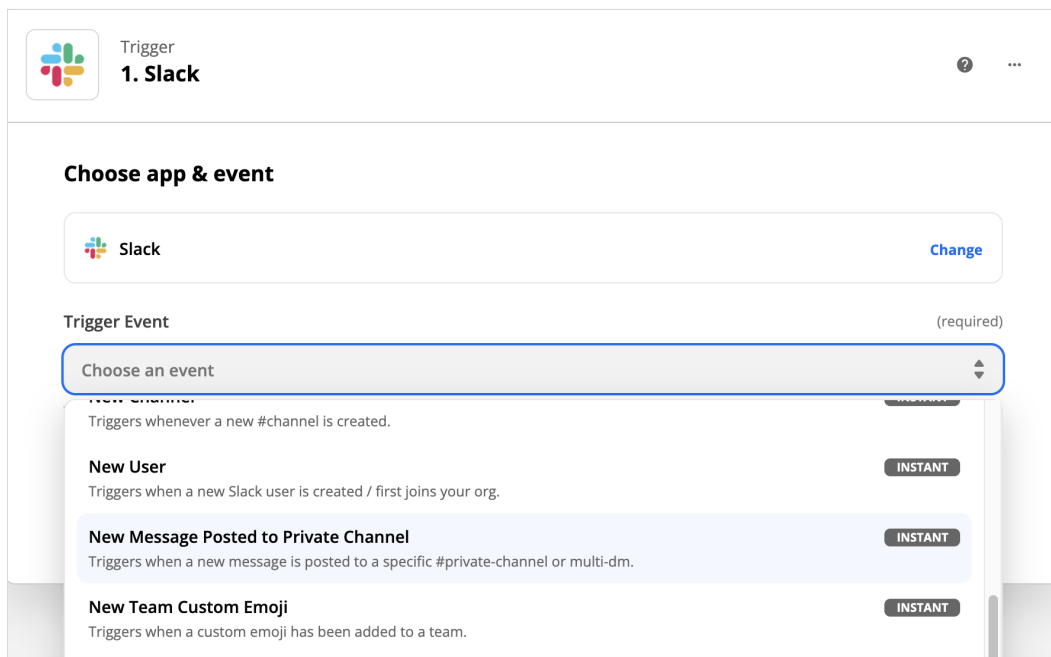
- Navigate to the Private Channel, click on the “Shortcut” option again and type in “Zapier”
- Connect the Zapier integration to Slack by clicking “Get Started” in Slack



- Log-in as needed to Zapier and you will be redirected to the “Zap” page
- Name your Zap, “Slack Application Request”
- Type Slack in the search box and select the Slack Integration



- Once selected, Slack will populate as the first step. Define the Trigger Event. For this example, we will use the “New Message Posted to Private Channel” trigger.



- Continue
- Select the slack instance/account you want to connect with.

Trigger

1. New Message Posted to Private Channel in Slack

Choose app & event ✓

Choose account

Slack account: (required) [Manage connected accounts](#)

Slack @admin (Guardians of the Cloud)

Slack is a secure partner with Zapier. [Your credentials are encrypted & can be removed at any time.](#)

Continue

- Define the specific channel that you want Zapier to listen to.
 - In this example, we want to listen to the #internalapprequest channel. All accessible channels will populate in dropdown for you to select from
 - Enable Trigger for Bot Messages

Choose account ✓

Set up trigger

Channel (required)

internal-app-request-approval

Trigger for Bot Messages? Yes No

If no, only human messages will trigger the Zap. If yes, both bots and humans will trigger it.

Warning: Bot users don't have profiles. If yes, don't depend on user profile information being present. Also, avoid zap loops where zaps trigger and post bot messages to the same channel!

Refresh fields

Continue



- Test the trigger to confirm the last message received.

Trigger

1. New Message Posted to Private Channel in Slack


- Choose app & event ✓
- Choose account ✓
- Set up trigger ✓

Test trigger

 → 



Test your trigger


We'll find a recent message posted in your Slack account

 Slack @admin (Guardians of the Cloud) to confirm that the right account is connected and your trigger is set up correctly.

[Test trigger](#)

Test trigger

 **We found a message posted!**
 This message posted was found in your  Slack @admin (Guardians of the Cloud) account. [Learn more about test data.](#)

 Message Posted C

Search...

```

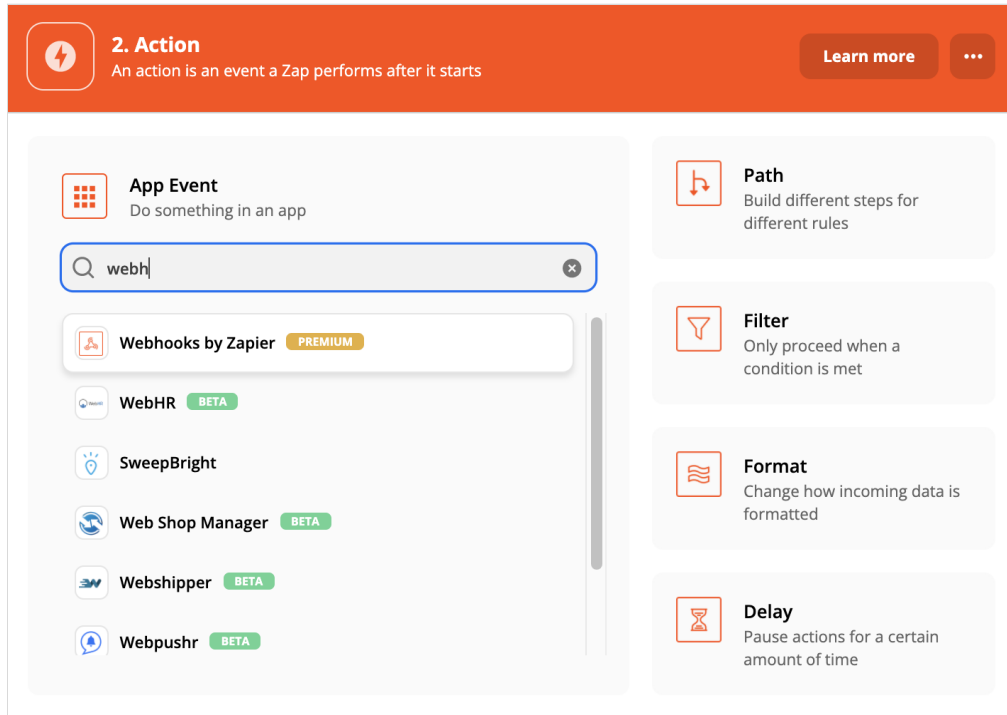
text: *App Request Submitted*
  RequestorEmail(admin@guardiansofthe.cloud)
  AppRequested(Zoom)
  DateRequested(&lt;date^1635445875^{date_long} {time_secs}&gt;)
text_RequestorEmail: admin@guardiansofthe.cloud
text_AppRequested: Zoom
text_DateRequested: &lt;date^1635445875^{date_long} {time_secs}&gt;
ts: 1635445920.000400
channel:
  id: C02K7A6JRPK
  name: internal-app-request-approval
thread_ts: 1635445920.000400
permalink: https://guardiansofthecloud.slack.com/archives/C02K7A6JRPK/p1635445920000400
user:
  name: App Request
  is_bot: true
team:

```

[Continue](#)

- This will conclude the Trigger Step of the Zap Integration. Continue.

- Step 2 - Now is time to configure the Webhook and payload to BetterCloud. Select “Webhooks by Zapier” from the App Event dropdown



- Select “POST” as the Action Event, we will be pushing the payload to BetterCloud. Continue
- Setting up the action
 - URL - We will now paste in the copied Endpoint URL from BetterCloud
 - Payload Type - Select JSON
 - Data - these are the data points we want to send to BetterCloud. Along with each data point, we can define the variable field to push, Zapier will dynamically pull back the fields from posted messages in the configured #internalapprequest channel
 - RequestorEmail
 - AppRequested
 - Leave all else as is and Continue.

Choose app & event

Set up action

URL (required)
 https://api.bettercloud.com/triggers/8307fd8f-c55a-4b5c-a8ec-df63e66013b0/106f8ebf-37fe-11ec-844e-df695fed0ec7/f40d5793-0325-4b57-9831-4fb266d540cc?authorization=401C07DeuvroojmxxvzoZseNgRvMuyXmj

Any URL with a querystring will be re-encoded properly.

Payload Type
 json

Pay special attention to the proper mapping of the data below.

Data

RequestorEmail 1. Text Requestor Email: admin@guardiansofthe.cloud

AppRequested

Insert Data ...

Search...

Go Back

1. Raw Text *App Request Submitted* RequestorEmail(<mailto:admin@guardiansofth

1. Raw Text App Requested Zoom

1. Raw Text Date Requested <date^1635445875^(date_long) {time_secs}>

1. Raw Text Requestor Email <mailto:admin@guardiansofthe.cloud|admin@guardiar

Wrap Request In Array

No

Wraps the request in an array if t

File

Enter text or insert data...

- Test the Action and confirm Success.

Action
2. POST

Choose app & event

Set up action

Test action

Test was successful!
 We'll use this as a sample for setting up the rest of your Zap.

A request was sent to Webhooks by Zapier just now.

Search request data...

text: Successfully received event

Retest action

Turn on Zap

Close

- Turn on the Zap once testing is successfully completed.

BetterCloud

- Return to the configuration page in BetterCloud for your Custom Trigger.
- If BetterCloud has not received an inbound trigger from Zapier, you will see the page will state that it is, “Still waiting for your request”. In this 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.
 - From here, define which parameters should be available as IF statements and which should be available as variables when performing actions within a BetterCloud workflow.
 - In this example, we will want the Application Requested to be available as an “IF Condition” and allow both parameters to be available as a dynamic field.

Note: If you need to rename any parameters at this point, selecting on the hyperlinked parameter name will direct you to a page to rename and provide a brief description for the field.

Slack Custom Webhook ×

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/8307fd8f-c55a-4b5c-a8ec-df63e66013b0/a70e25b3-35ce-11ec-844e-273924be38fb/7ad9cd4d-9ae2-47a1-a452-636b460a0789?authorization=RxSvIs046nPRI1chAyOU6rgpMyu8vPET>

[View Raw Data Response](#)

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

Requestor Email

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

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

Please do not share sensitive information. For best practices, please review [our support article](#).

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

- You can now “Save & Publish” your new Custom Trigger.
 - You cannot Save & Publish this trigger until you have selected the target parameter, this is the parameter that will be associated with the triggered workflow.
- From here, you are ready to build a new workflow that will trigger from an Application Request being submitted from Slack.