

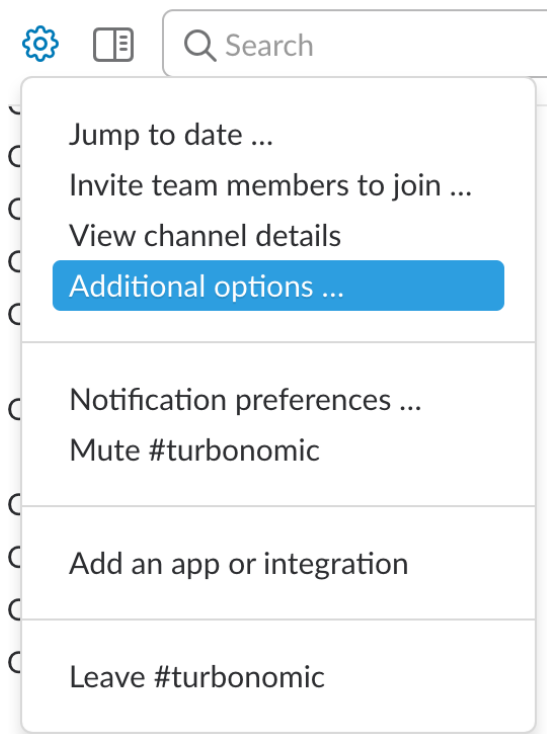
Setting up a Slack WebHook to Post Notifications to a Team Channel

If ChatOps is something you've been hearing a lot about, there is a reason. Slack is fast becoming the de facto standard in what we are calling ChatOps. Before we go full out into making chatbots and such, the first cool use-case I explored is enabling notifications for different systems.

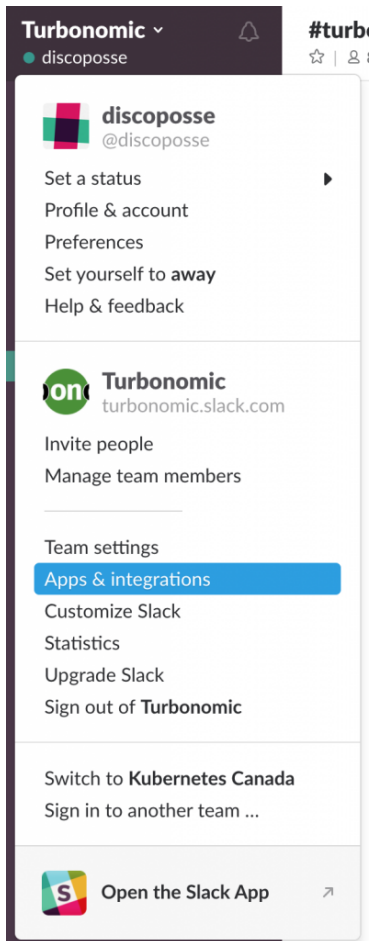
In order to do any notifications to Slack, you need to enable a WebHook. This is super easy but it made sense for me to give you the quick example so that you can see the flow yourself.

Setting up the Slack Webhook

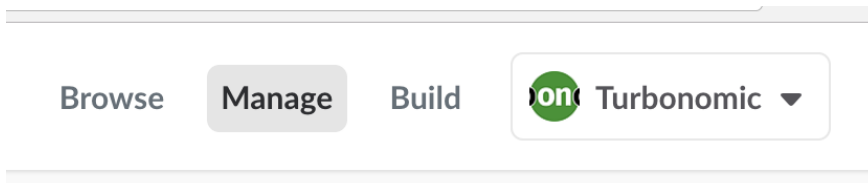
First, we login to your Slack team in the web interface. From there we can open up the management view of the team to be able to get to the apps and integrations. Choose Additional Options under the settings icon:



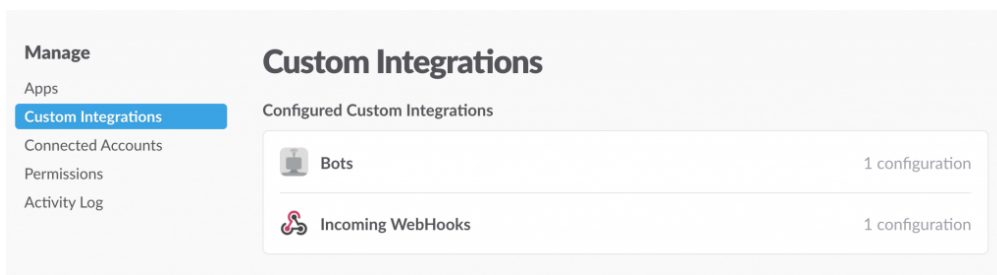
You can also get there by using the droplets in left-hand pane and selecting **Apps and Integrations** from the menu:



Next, click the **Manage** button in the upper right portion of the screen near the team name:



Select **Custom Integrations** and then from there click the **Incoming WebHooks** option:



Choose the channel you want to post to and then click the **Add Incoming WebHooks Integration** button:

Post to Channel

Start by choosing a channel where your Incoming Webhook will post messages to.

#turbonomic or [create a new channel](#)

Add Incoming WebHooks integration

By creating an incoming webhook, you agree to the [Slack API Terms of Service](#).

It's really just that easy! You will see a results page with a bunch of documentation such as showing your WebHook URL:

Setup Instructions close

We'll guide you through the steps necessary to configure an Incoming Webhook so you can start sending data to Slack.

Webhook URL `https://hooks.slack.com/services/T2CR5L/B5JS35JS3/4Rm7Wj4aeRK6`


Other parts of the documentation also show you how to configure some customizations and even an example cURL command to show how to do a post using the new WebHook integration:

Example

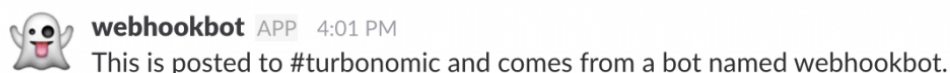
Putting this all together, here is a sample [curl](#) request for posting to a channel.

```
curl -X POST --data-urleencode 'payload={"channel": "#turbonomic", "u
```

This will be displayed in the channel as:

 **webhookbot** 11:43 AM
This is posted to #general and comes from a bot named webhookbot.

If you go out to a command line where you have the cURL command available, you can run the example command and you should see the results right in your Slack UI:



There are many other customization options such as which avatar to use, and the specifics of the command text and such. You can get at the WebHook any time under the **Incoming WebHooks** area within the Slack admin UI:

[< Browse Apps](#)



Incoming WebHooks

Incoming Webhooks are a simple way to post messages from external sources into Slack. They make use of normal HTTP requests with a JSON payload, which includes the message and a few other optional details described later.

[Message Attachments](#) can also be used in Incoming Webhooks to display richly-formatted messages that stand out from regular chat messages.

[Add Configuration](#)

[Help and support](#) >

[Privacy policy](#) >

Configurations



Posts to #turbonomic as disco-webhook
discofosse on Apr 24, 2017



Now all you have to do is configure whatever script or function you have that you want to send notifications to Slack with and you are off to the races.