

Setting up GitLab, GitHub, JIRA and Webhooks Integrations With the Installer

In Element Server Suite, our GitLab, GitHub, and JIRA extensions are provided by the hookshot package. This documentation explains how to configure hookshot.

Configuring Hookshot with the Installer

From the Installer's Integrations page, click "Install" under "Hookshot: Github, Gitlab, Jira, and Custom Webhooks."



Hookshot: Github, Gitlab, Jira, and Custom Webhooks.

Forward messages from external sources into rooms as they occur

[Cancel and return to Integrations](#)

Logging Level

Info

Default ▼

Verify TLS

Use Global Setting

Default ▼

TLS Verification

Secrets

/

Hookshot

/

Hookshot Pass Key ▼

**Hookshot
Password key**

Choose File No file chosen

Edited You can generate it using : `openssl genpkey -out passkey.pem -outform PEM -algorithm RSA -pkeyopt rsa_keygen`

Secrets

/

Hookshot

/

Provisioning Secret ▼

Hookshot provisioning secret

.....



Edited

This password should be generated randomly.

On the first screen here, we can set the logging level and a hookshot specific verify tls setting. Most users can leave these alone.

To use hookshot, you will need to generate a hookshot password key, when can be done by running the following command on a Linux command line:

```
openssl genpkey -out passkey.pem -outform PEM -algorithm RSA -pkeyopt rsa_keygen_bits:4096
```

which will generate output similar to this:

```
.....+++++
.....++++
```

Once this has finished, you will have a file called passkey.pem that can use to upload as the "Hookshot Password key".

If you wish to change the hookshot provisioning secret, you can, but you can also leave this alone as it is randomly generated by the installer.

Bot

The hookshot bot

Avatar

mx://

Default

The hookshot bot avatar mxc url

Display Name

Hookshot Bot

Default

The hookshot bot display name

Username

hookshot

Default

The hookshot bot username

Widgets

The hookshot widgets settings

SHOW

GitLab

Gitlab hooks



Jira



Webhooks

Configuration of hookshot generic webhooks



Next, we get to a set of settings that allow us to make changes to the Hookshot bot's appearance.

There is also a button to show widget settings, which brings up these options:

Widgets

The hookshot widgets settings

HIDE

☒ Add on Invite

Add widgets on invite

Default

☐ Add to Admin Rooms

Add widgets to admin rooms

Default

Title

Hookshot Configuration

Default

The hookshot widget title

Disallowed IP Ranges

Which IP ranges should be disallowed when resolving homeserver IP addresses (for security reasons). Unless you know what you are doing, it is recommended to not change this.

=

An IP range, ipv4 or ipv6 format

192.168.122.0/24

Default

⊖

=

An IP range, ipv4 or ipv6 format

127.0.0.0/8

Default

⊖

=

An IP range, ipv4 or ipv6 format

10.0.0.0/8

Default

⊖

=

An IP range, ipv4 or ipv6 format

172.16.0.0/12

Default

⊖

In this form, we have the ability to control how widgets are incorporated into rooms (the defaults are usually fine) and to set a list of Disallowed IP ranges wherein widgets will not load if the homeserver IP falls in the range. If your homeservers IP falls in any of these ranges, you will want to remove that range so that the widgets will load!

Next, we have the option to enable Gitlab, which shows us the following settings:

GitLab

Gitlab hooks

Secrets

 /

Hookshot

 /

GitLab Webhook Secret

Gitlab webhook secret

.....

Edited

This password should be generated and shared with gitlab on the webhook page.

Gitlab instances Rooms

Name *

Gitlab instance name

URL *

Gitlab instance URL

ADD MORE GITLAB INSTANCES ROOMS

The webhook secret is randomly generated and does not need to be changed. You can also add Gitlab instances by specifying an instance name and pasting the URL.

Next, we have the option to enable Jira, which shows us the following settings:

The screenshot shows a configuration page for Jira. At the top, the word "Jira" is displayed next to a green toggle switch. Below this is a section for "OAuth Client ID *". It includes a text input field, a label "Jira OAuth client id", and a breadcrumb trail: "Secrets / Hookshot / Jira OAuth Client Secret". Below the input field is a label "Jira OAuth client secret" and a text note: "This secret should be copied from the OAuth secret in Jira settings." The next section is for "Jira Webhook Secret". It has a breadcrumb trail: "Secrets / Hookshot / Jira Webhook Secret". Below this is a text input field labeled "Jira webhook secret" containing a series of dots, a label "Jira webhook secret", an eye icon, and a green "Edited" button. A text note below the field states: "This password should be generated and shared with Jira on the webhook page."

In here, we can specify the OAuth Client ID and the OAuth client secret to connect to Jira. To obtain this information, please follow these steps:

The JIRA service currently only supports atlassian.com (JIRA SaaS) when handling user authentication. Support for on-prem deployments is hoping to land soon.

- You'll first need to head to <https://developer.atlassian.com/console/myapps/create-3lo-app/> to create a "OAuth 2.0 (3LO)" integration.
- Once named and created, you will need to:
- Enable the User REST, JIRA Platform REST and User Identity APIs under Permissions.
- Use rotating tokens under Authorisation.
- Set a callback url. This will be the public URL to hookshot with a path of /jira/oauth.
- Copy the client ID and Secret from Settings

Once you've set these, you'll notice that a webhook secret has been randomly generated for you. You can leave this alone or edit it if you desire.

Next, let's look at configuring Webhooks:

Webhooks



Configuration of hookshot generic webhooks

☒ Allow JS Transformation Functions

To allow JS Transformations functions

Default

☒ Enabled

Enable or disable generic webhooks

Default

User ID Prefix

webhooks user id prefixes

You can set whether or not webhooks are enabled and whether they allow JS Transformation functions. It is good to leave these enabled per the defaults. You can also specify the user id prefix for the creation of custom webhooks. If you set this to `webhook_` then each new webhook will appear in a room with a username starting with `webhook_`.

Next, let's look at configuring Github:

GitHub



Configuration of hookshot github integration

Auth ID *

Github application auth id

OAuth Client ID *

Github OAuth client id

Secrets

/

Hookshot

/

GitHub Key File ▾

**Github
application key
file**

Choose File

No file chosen

Edited

It can be generated by clicking "Generate a private key" under the Private keys section on the GitHub app page

Secrets

/

Hookshot

/

GitHub OAuth Client Secret ▾

Github OAuth client secret



The OAuth Client secret of the github app page.

Secrets

/

Hookshot

/

GitHub Webhook Secret ▾

Github webhook secret



It is the Webhook secret on the GitHub App page.

This bridge requires a [GitHub App](#). You will need to create one. Once you have created this, you'll be able to fill in the Auth ID and OAuth Client ID. You will also need to generate a "Github application key file" to upload this. Further, you will need to specify a "Github OAuth client secret" and a "Github webhook secret", both of which will appear on your newly created Github app page.

Default Options

The default options to apply to github hooks

Command Prefix

Choose the prefix to use when sending commands to the bot. Ideally starts with "!" lgh

Hotlink command prefix

Send a link to an issue/PR in the room when a user mentions a prefix followed by a number

☐ Share issue room link on new issues

When new issues are created, provide a Matrix alias link to the issue room

Default

Enable Hooks

Enable notifications for some event types

ADD ENABLE HOOKS

Excluding Labels

Never notify on issues matching these label names

ADD EXCLUDING LABELS

Ignore Hooks

Choose to exclude notifications for some event types

ADD IGNORE HOOKS

On this screen, we have the option to change how we call the bot and other minor settings. We also have the ability to select which hooks we provide notifications for, what labels we wish to exclude, and then which hooks we will ignore completely.

Including Labels

Only notify on issues matching these label name

ADD INCLUDING LABELS

New issues options

Configuration options for new issues

Labels

Automatically set these labels on issues created via commands

ADD LABELS

PR Diff

Show a diff in the room when a PR is created, subject to limits

☐ Enabled

Enable the PR diff

Default

Max Lines

0

Default

Max number of lines to display in the room

Now we have the ability to add a list of labels that we want to match. This has the impact of the integration only notifying you of issues with a specific set of labels.

We then have the ability to add a list of labels that all newly created issues through the bot should be labeled with.

Then we have the ability to enable showing diffs in the room when a PR is created.

Workflow Run

Configuration options for workflow run results

Matching Branch

Only report workflow runs if it matches this regex.

Excluding Workflows

Never report workflow runs with a matching workflow name.

[ADD EXCLUDING WORKFLOWS](#)

Including Workflows

Only report workflow runs with a matching workflow name.

[ADD INCLUDING WORKFLOWS](#)

Moving along, we can configure how workflow run results are configured in the bot, including matching specific workflows and including or excluding specific workflows.

Finishing Configuration

You further have the ability to click "Advanced" and set any kubernetes specific settings for how this pod is run. Once you have set everything up on this page, you can click "Continue" to go back to the Integrations page.

When you have finished running the installer and the hookshot pod is up and running, there are some configurations to handle in the Element client itself in the rooms that you wish the integration to be present.

As an admin, you will need to enable hookshot in the rooms using the "Add widgets, bridges, & bots" functionality to add the "Hookshot" widget to the room and finish the setup.