

Domain Configuration for Web Push Consents: Adding the Service Worker File via Cloudflare

Introduction

If you want Web Push consents to be assigned to your own domain, and not `salesmanagopush.com`, you need to place an `sw.js` file in the root directory of your website server. However, many SaaS platforms restrict direct access to the root directory, making this process challenging. While adding a `Service-Worker-Allowed` header could be a solution, this option is often not feasible on SaaS platforms. Fortunately, Cloudflare offers a workaround by allowing you to serve the `sw.js` file without obtaining direct access to the root directory.

This guide provides instructions on how to use Cloudflare to place the `sw.js` file on supported SaaS platforms.

IMPORTANT: This approach is **not compatible with Shopify**, as the platform actively prevents the use of Cloudflare for this purpose.

Instructions

The process described in this guide is relatively simple and can be divided into four stages: setting up your domain on Cloudflare; creating a worker file responsible for adding the sw.js file; assigning the worker file to your domain; and changing the response headers to make it appear as JavaScript.

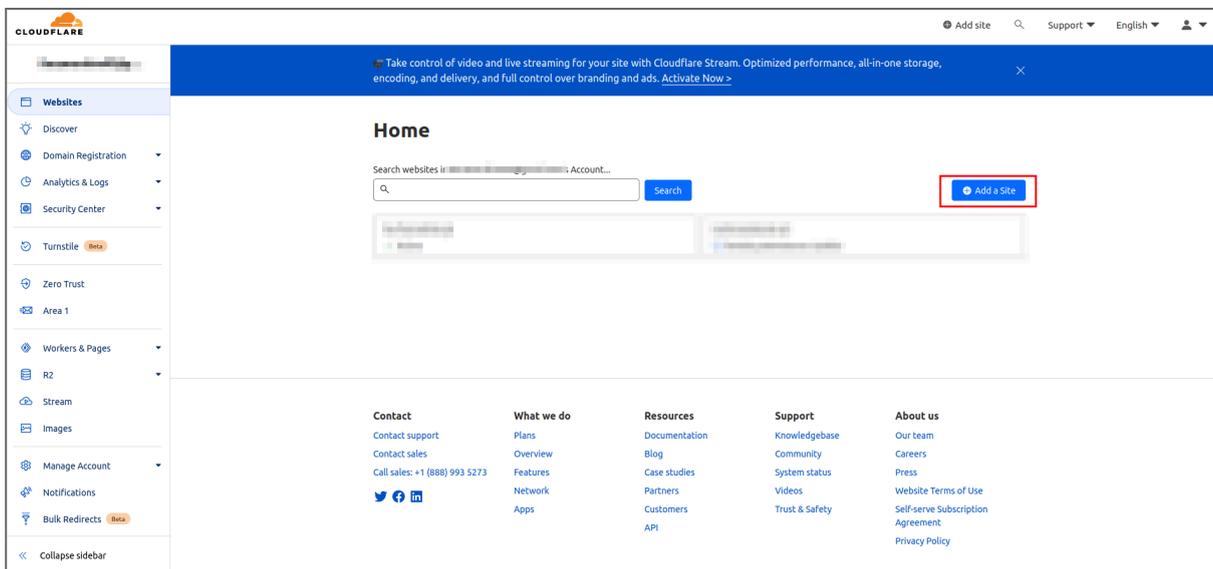
Step 1.

Create a Cloudflare account. You can use the free plan for the purpose described in this guide.

<https://www.cloudflare.com/plans/>

Step 2.

Go to the main Cloudflare panel and click *Add a Site* to add your domain to Cloudflare. Follow the instructions provided by Cloudflare.



Step 3.

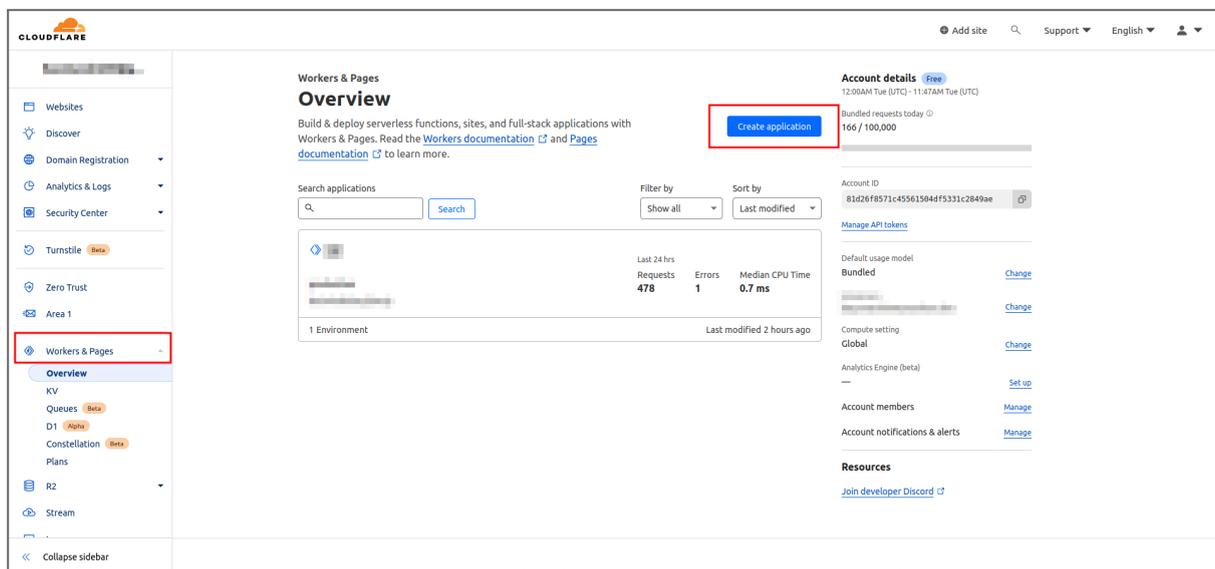
Change the DNS settings of your domain according to the instructions provided by Cloudflare.

<https://developers.cloudflare.com/dns/zone-setups/full-setup/setup/>

When this step is completed, your domain must be verified by Cloudflare.

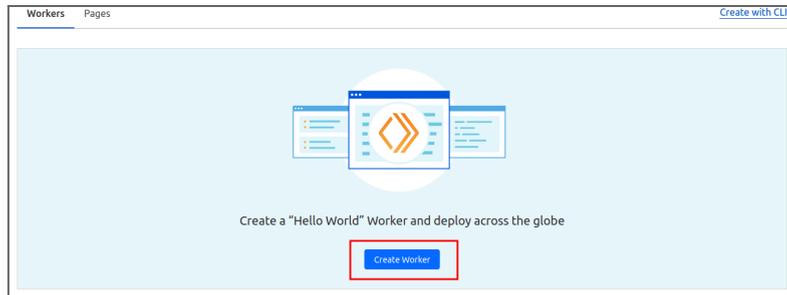
Step 4.

When your domain has been verified by Cloudflare, go to the *Workers & Pages* section and click *Create application*.

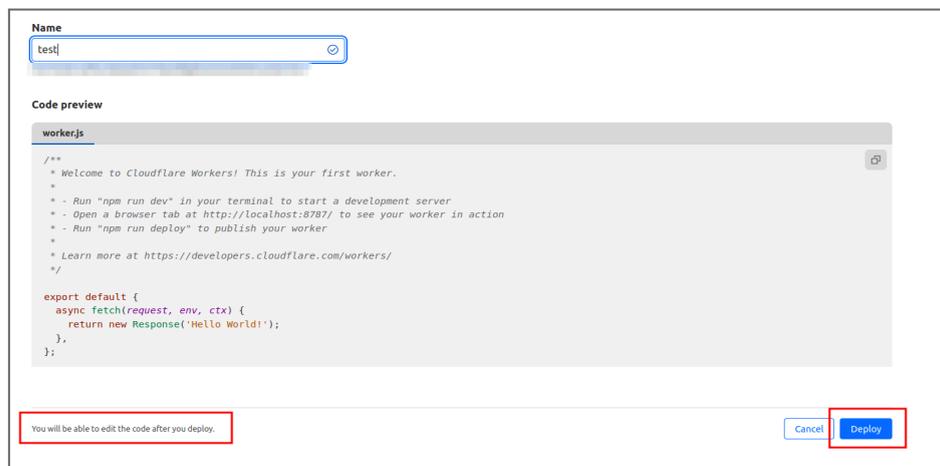


Step 5.

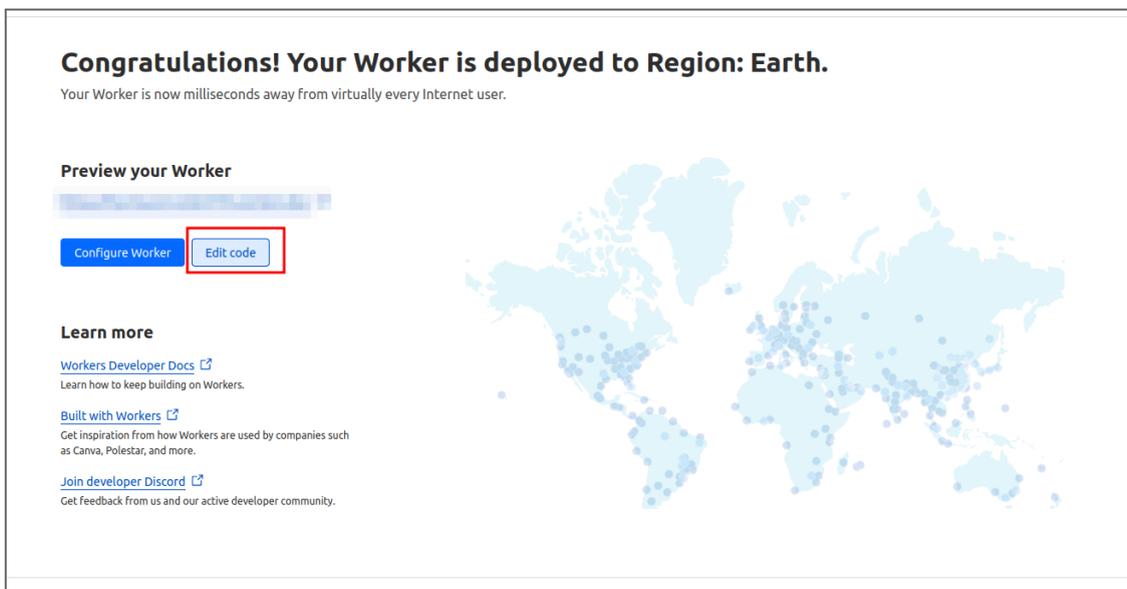
Click *Create worker* to create a script that will allow you to programmatically add sw.js file to your website.



Next, set a name for the worker, e.g. "Add sw.js file", and click *Deploy* to continue.

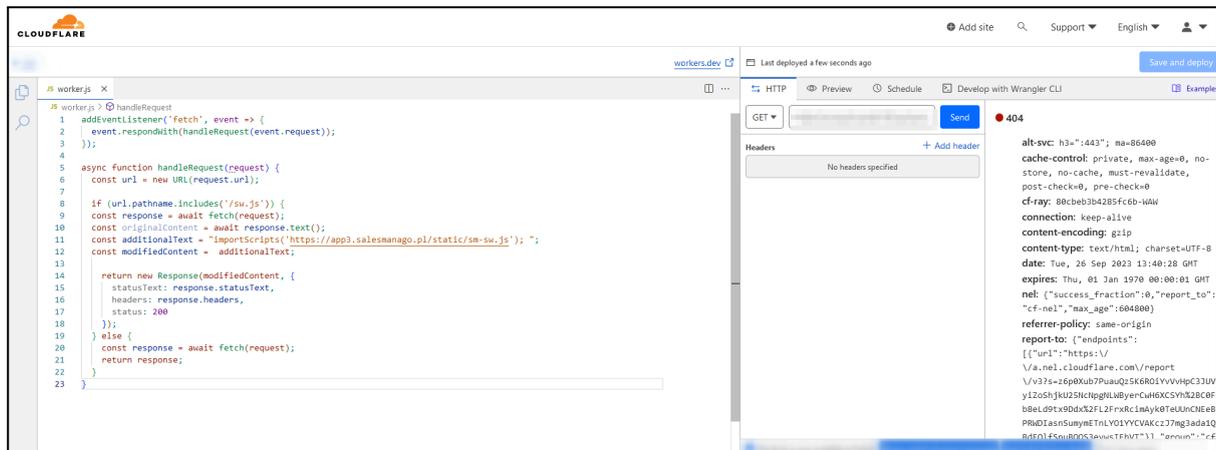


Click *Edit code* to enter the editing screen.



Step 6.

Use the left-hand panel to add the code snippet provided below.



You can remove all the default content of the worker file and just paste the copied code.

IMPORTANT: Remember to substitute the xxx in `importScripts` instruction with your SALESmanago endpoint, e.g. “www”, “app2”, or “app3”.

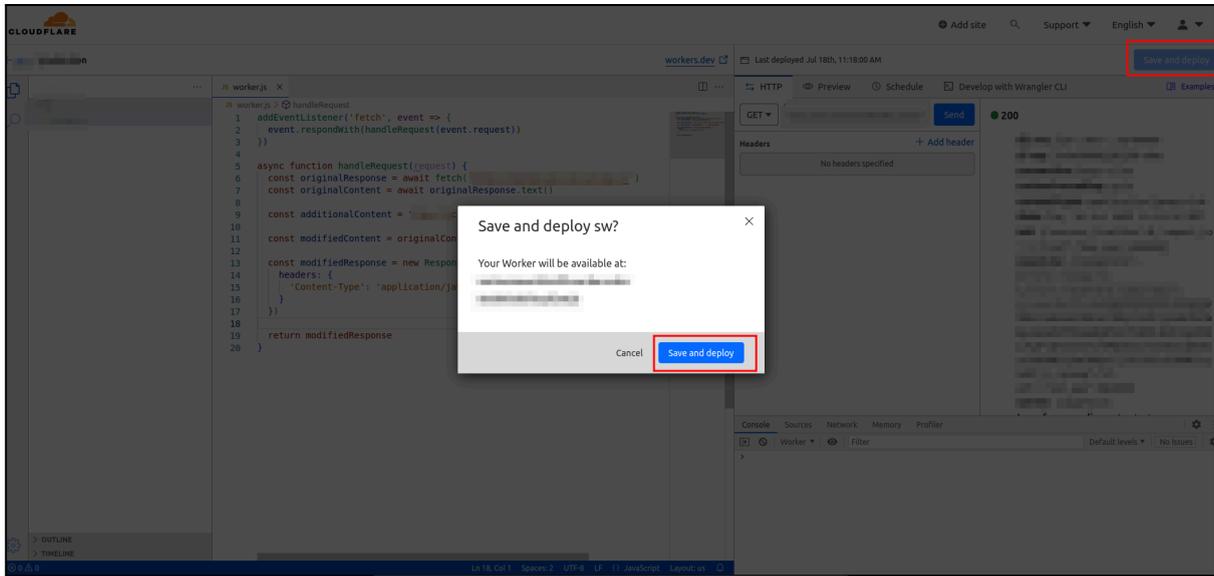
CODE SNIPPET TO BE INCLUDED IN THE WORKER FILE:

```
JavaScript
addEventListener("fetch", (event) => {
  event.respondWith(handleRequest(event.request));
});
async function handleRequest(request) {
  const url = new URL(request.url);
  if (url.pathname.includes('/sw.js')) {
    const response = await fetch(request);
    await response.text();
    const modifiedContent =
      "importScripts('https://xxx.salesmanago.com/static/sm-sw.js');";

    return new Response(modifiedContent, {
      statusText: response.statusText,
      headers: response.headers,
      status: 200
    });
  } else {
    return await fetch(request);
  }
}
```

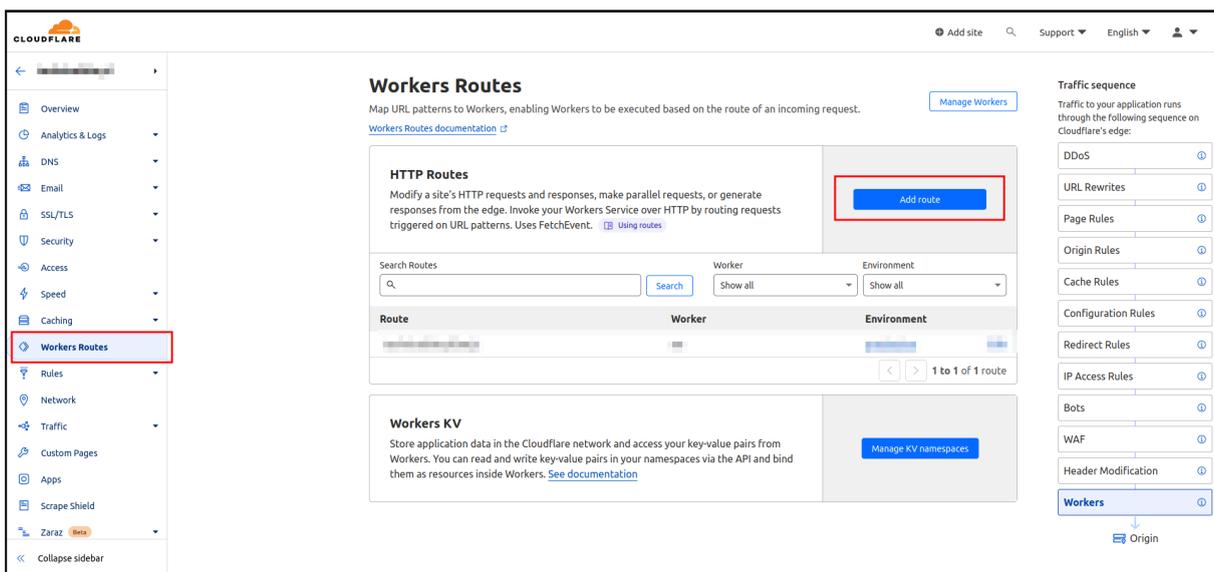
Step 7.

Click *Save and deploy* to save the changes. Next, confirm it by clicking *Save and deploy* again.



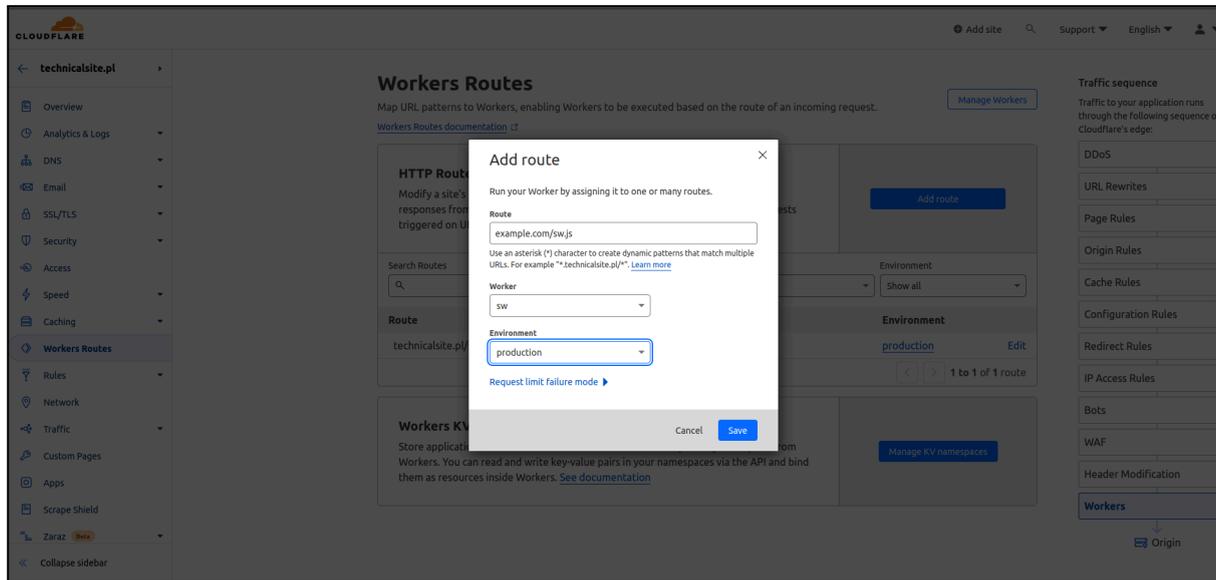
Step 8.

At this stage, the worker is ready, however, it is not yet assigned to your domain. To do this, go to the *Website* tab and select your domain. Next, go to the *Workers Routes* section and click *Add route*.



In the *Route* field, specify your domain followed by */sw.js*, for example:

example.com/sw.js

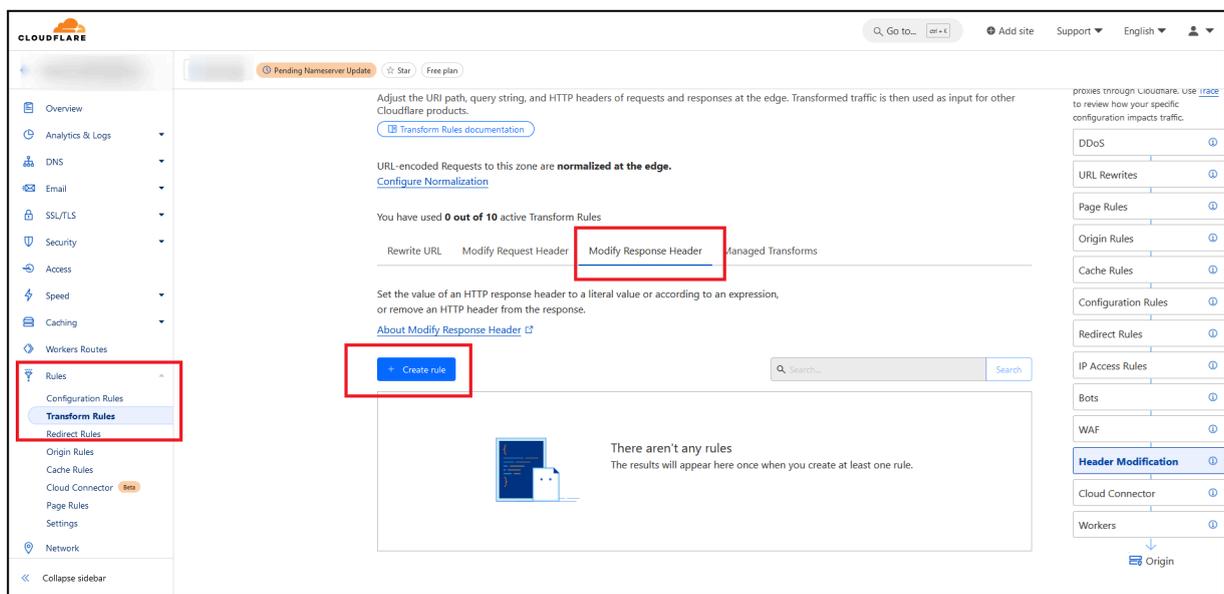


Save the settings and wait for approximately 15 minutes for the changes to be applied.

Step 9.

Select *Rules* from the left-hand panel. If you cannot see this option, make sure you are in the *Website* → *[your domain]* section.

Go to *Transform Rules* → *Modify Response Header* tab and click *Create rule*.



Apply the following configuration:

When incoming requests match...

All incoming requests
The rule will apply to all traffic

Custom filter expression
The rule will only apply to traffic matching the custom expression

When incoming requests match...

Field	Operator	Value		
URI Path	contains	sw.js	And	Or

e.g. /content

Expression Preview [Edit expression](#)

```
(http.request.uri.path contains "sw.js")
```

Then...

Modify response header ⓘ
You can modify up to 30 response headers

	Header name		Value
Set static	Content-Type	=	text/javascript

[+ Set new header](#)

[Cancel](#) [Save](#)

Name—Enter any name, for example, “sw.js content type overwrite”

When incoming request match—Set *Custom filter expression*

Field—Set *URI Path*

Operator—Set *contains*

Value—Set *sw.js*

In the section “Then... Modify response header” set the following:

Dropdown—Set *static*

Header name—Set *Content-Type*

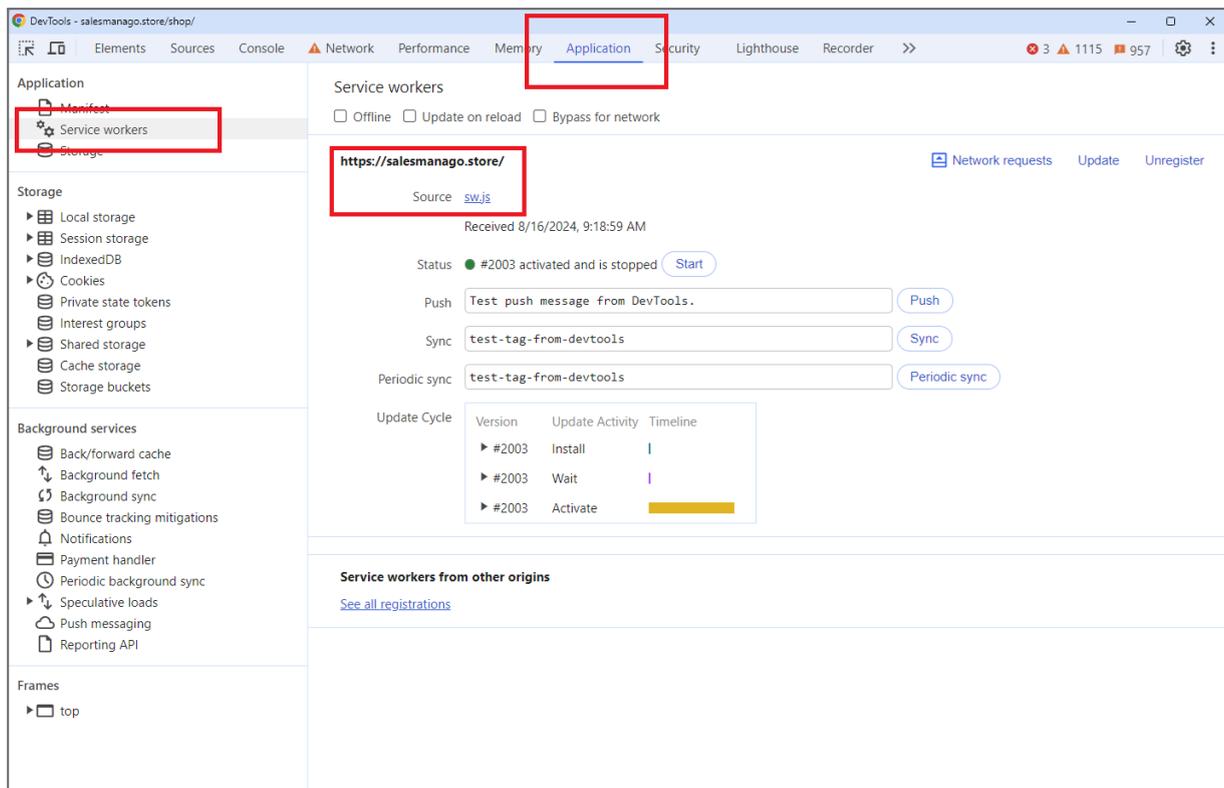
Value—Set *text/javascript*

Click *Save/Deploy* to apply the changes.

Testing the changes

In order to verify whether the service worker has been successfully added to your domain, wait for 15 minutes, then go to your website and open developer tools (F12).

Go to the *Application* tab and select *Service workers*. If the `sw.js` file is present next to the *Source* label, the setup is complete.



From now on, collected Web Push consents will be assigned to your own domain.

If you continue to use any existing consent prompts, Web Push notifications associated with these consent prompts will be sent to Contacts who consented both before and after the addition of the `sw.js` file, however, the former will still see “salesmanagopush.com” as the sender.

You can also create new consent prompts and start collecting consents anew.

If you still have any questions or need assistance, please contact us at:

support@salesmanago.com